

Dizionario di ActionScript

Marchi

Afterburner, AppletAce, Attain, Attain Enterprise Learning System, Attain Essentials, Attain Objects for Dreamweaver, Authorware, Authorware Attain, Authorware Interactive Studio, Authorware Star, Authorware Synergy, Backstage, Backstage Designer, Backstage Desktop Studio, Backstage Enterprise Studio, Backstage Internet Studio, Design in Motion, Director, Director Multimedia Studio, Doc Around the Clock, Dreamweaver, Dreamweaver Attain, Drumbeat, Drumbeat 2000, Extreme 3D, Fireworks, Flash, Fontographer, FreeHand, FreeHand Graphics Studio, Generator, Generator Developer's Studio, Generator Dynamic Graphics Server, Knowledge Objects, Knowledge Stream, Knowledge Track, Lingo, Live Effects, Macromedia, Macromedia M Logo & Design, Macromedia Flash, Macromedia Xres, Macromind, Macromind Action, MAGIC, Mediamaker, Object Authoring, Power Applets, Priority Access, Roundtrip HTML, Scriptlets, SoundEdit, ShockRave, Shockmachine, Shockwave, Shockwave Remote, Shockwave Internet Studio, Showcase, Tools to Power Your Ideas, Universal Media, Virtuoso, Web Design 101, Whirlwind e Xtra sono marchi di Macromedia, Inc. e possono essere registrati negli Stati Uniti o in altre giurisdizioni incluse le giurisdizioni internazionali. Altri nomi di prodotti, logo, progetti, titoli, termini o frasi che appaiono nella presente pubblicazione possono essere marchi, marchi di servizio o denominazioni commerciali di Macromedia, Inc. o di altre società e potrebbero essere registrati in alcune giurisdizioni, incluse le giurisdizioni internazionali.

Informazioni su prodotti di terzi

Tecnologia per la compressione e la decompressione vocale concessa in licenza da Nellymoser, Inc. (www.nellymoser.com).



Tecnologia per la compressione e la decompressione video Sorenson™ Spark™, concessa in licenza da Sorenson Media, Inc.

La presente Guida contiene collegamenti a siti Web di terzi non gestiti da Macromedia; pertanto Macromedia non è responsabile del contenuto di tali siti. Qualora si acceda a un sito Web di terzi menzionato nella presente Guida, lo si farà a proprio rischio. Tali collegamenti sono inclusi nella presente Guida solo per la loro utilità e ciò non implica che Macromedia approvi il loro contenuto o se ne assuma la responsabilità.

Dichiarazione di non responsabilità Apple

APPLE COMPUTER, INC. NON FORNISCE ALCUNA GARANZIA, ESPLICITA O IMPLICITA, PER IL PACCHETTO SOFTWARE ACCLUSO, IN RELAZIONE ALLA SUA COMMERCIALIZZAZIONE O IDONEITÀ PER SCOPI SPECIFICI. POICHÉ L'ESCLUSIONE DELLE GARANZIE IMPLICITE NON È CONSENTITA DA ALCUNI PAESI, POTREBBE NON ESSERE APPLICABILE. LA PRESENTE GARANZIA CONFERISCE ALL'UTENTE PARTICOLARI DIRITTI. ALTRI DIRITTI POSSONO ESSERE RITENUTI VALIDI, A SECONDA DEL PAESE.

Copyright © 2002 Macromedia, Inc. Tutti i diritti riservati. Nessuna parte della presente guida può essere copiata, fotocopiata, riprodotta, tradotta o convertita in qualsiasi formato elettronico o meccanico senza la previa autorizzazione scritta di Macromedia, Inc.

Ringraziamenti

Direttore: Erick Vera

Produttore: Wayne Wieseler

Redazione: Jody Bleyle, JuLee Burdekin, Mary Burger, Dale Crawford, Marcelle Taylor

Progetto didattico: Stephanie Gowin, Barbara Nelson

Revisione: Rosana Francescato, Lisa Stanziano, Anne Szabla

Produzione e progetto multimediale: Aaron Begley, Benjamin Salles, Noah Zilberberg

Produzione e progetto di stampa: Chris Basmajian, Caroline Branch

Localizzazione: Michael Dominguez, Cristina Guembe, Yoshika Hedberg, Tim Hussey, Masayo "Noppe" Noda, Simone Pux, Yoko Shindo, Yuko Yagi

Prima edizione: marzo 2002

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

SOMMARIO

Voci di esempio per la maggior parte degli elementi di ActionScript	21
Voci di esempio per oggetti e componenti	22
Contenuto del dizionario	23
— (decremento)	41
++ (incremento)	42
! (NOT logico)	43
!= (diseguaglianza)	43
!= (diseguaglianza rigorosa)	44
% (modulo)	45
%= (assegnazione modulo)	46
& (AND bit a bit)	46
&& (cortocircuito AND)	47
&= (assegnazione AND bit a bit)	48
() (parentesi tonde)	48
- (meno)	49
* (moltiplicazione)	51
*= (assegnazione moltiplicazione)	52
, (virgola)	52
. (punto)	53
?: (condizionale)	54
/ (divisione)	54
// (delimitatore di commento)	55
/* (delimitatore di commento)	56
/= (assegnazione divisione)	56
[] (accesso matrice)	57
^(XOR bit a bit)	59
^= (assegnazione XOR bit a bit)	60
{ (operatore di inizializzazione degli oggetti)	60
(OR bit a bit)	61
(OR logico)	62
= (assegnazione OR bit a bit)	63
~ (NOT bit a bit)	64
+ (addizione)	64
+= (assegnazione addizione)	65
< (minore di)	66
<< (spostamento a sinistra bit a bit)	67
<<= (spostamento a sinistra bit a bit e assegnazione)	68
<= (minore o uguale a)	69

<> (diseguaglianza)	70
= (assegnazione)	71
-= (assegnazione sottrazione)	71
== (uguaglianza)	72
=== (uguaglianza rigorosa)	73
> (maggiore di)	74
>= (maggiore di o uguale a)	75
>> (spostamento a destra bit a bit)	75
>>= (spostamento a destra bit a bit e assegnazione)	76
>>> (spostamento a destra senza segno bit a bit)	77
>>>= (spostamento a destra senza segno bit a bit e assegnazione)	78
Accessibility (oggetto)	78
Accessibility.isActive	79
add	79
and	80
arguments (oggetto)	80
arguments.callee	80
arguments.caller	81
arguments.length	81
Array (oggetto)	81
Array.concat	83
Array.join	84
Array.length	85
Array.pop	85
Array.push	86
Array.reverse	86
Array.shift	87
Array.splice	87
Array.sort	88
Array.sortOn	89
Array.splice	90
Array.toString	91
Array.unshift	91
asfunction	92
Boolean (funzione)	92
Boolean (oggetto)	93
Boolean.toString	94
Boolean.valueOf	94
break	94
Button (oggetto)	95
Riepilogo eventi dell'oggetto Button	96
Button._alpha	97
Button.enabled	97
Button._focusrect	97
Button.getDepth	98
Button._height	98
Button._highquality	98
Button._name	99
Button.onDragOut	99
Button.onDragOver	99

Button.onKeyDown	100
Button.onKeyUp	100
Button.onKillFocus	101
Button.onPress	101
Button.onRelease	102
Button.onReleaseOutside	102
Button.onRollOut	103
Button.onRollOver	103
Button.onSetFocus	104
Button._parent	104
Button._quality	105
Button._rotation	105
Button._soundbuftime	105
Button.tabEnabled	106
Button.tabIndex	106
Button._target	107
Button.trackAsMenu	107
Button._url	107
Button.useHandCursor	108
Button._visible	108
Button._width	108
Button._x	109
Button._xmouse	109
Button._xscale	109
Button._y	110
Button._ymouse	110
Button._yscale	110
call	111
call (funzione)	111
case	111
chr	112
clearInterval	112
Color (oggetto)	113
Color.getRGB	114
Color.getTransform	114
Color.setRGB	115
Color.setTransform	115
continue	117
CustomActions (oggetto)	117
CustomActions.get	118
CustomActions.install	118
CustomActions.list	119
CustomActions.uninstall	119
Date (oggetto)	120
Date.getDate	123
Date.getDay	123
Date.getFullYear	123
Date.getHours	124
Date.getMilliseconds	124
Date.getMinutes	125

Date.getMonth	125
Date.getSeconds	125
Date.getTime	126
Date.getTimezoneOffset	126
Date.getUTCDate	127
Date.getUTCDay	127
Date.getUTCFullYear	127
Date.getUTCHours	128
Date.getUTCMilliseconds	128
Date.getUTCMinutes	128
Date.getUTCMonth	129
Date.getUTCSeconds	129
Date.getYear	129
Date.setDate	130
Date.setFullYear	130
Date.setHours	131
Date.setMilliseconds	131
Date.setMinutes	131
Date.setMonth	132
Date.setSeconds	132
Date.setTime	132
Date.setUTCDate	133
Date.setUTCFullYear	133
Date.setUTCHours	134
Date.setUTCMilliseconds	134
Date.setUTCMinutes	134
Date.setUTCMonth	135
Date.setUTCSeconds	135
Date.setYear	136
Date.toString	136
Date.UTC	137
default	137
delete	138
do while	140
duplicateMovieClip	140
else	141
else if	141
#endinitclip	142
eq (uguale; specifico per stringhe)	143
escape	143
eval	144
evaluate	145
false	145
FCheckBox (componente)	145
FCheckBox.setEnabled	146
FCheckBox.getLabel	147
FCheckBox.getValue	147
FCheckBox.registerSkinElement	148
FCheckBox.setChangeHandler	148
FCheckBox.setEnabled	149

FCheckBox.setLabel	150
FCheckBox.setLabelPlacement	150
FCheckBox.setSize	151
FCheckBox.setStyleProperty	151
FCheckBox.setValue	152
FComboBox (componente)	153
FComboBox.addItem	154
FComboBox.addItemAt	155
FComboBox.setEnabled	156
FComboBox.getItemAt	157
FComboBox.getLength	157
FComboBox.getRowCount	158
FComboBox.getScrollPosition	158
FComboBox.getSelectedIndex	159
FComboBox.getSelectedItem	159
FComboBox.getValue	160
FComboBox.registerSkinElement	160
FComboBox.removeAll	161
FComboBox.removeItemAt	163
FComboBox.replaceItemAt	163
FComboBox.setChangeHandler	164
FComboBox.setDataProvider	165
FComboBox.setEditable	166
FComboBox.setEnabled	167
FComboBox.setItemSymbol	167
FComboBox.setRowCount	168
FComboBox.setSelectedIndex	168
FComboBox.setSize	169
FComboBox.setStyleProperty	169
FComboBox.setValue	170
FComboBox.sortItemsBy	171
FListBox (componente)	171
FListBox.addItem	173
FListBox.addItemAt	174
FListBox.setEnabled	174
FListBox.getItemAt	175
FListBox.getLength	176
FListBox.getRowCount	176
FListBox.getScrollPosition	177
FListBox.getSelectedIndex	177
FListBox.getSelectedIndices	178
FListBox.getSelectedItem	178
FListBox.getSelectedItems	179
FListBox.selectMultiple	179
FListBox.getValue	180
FListBox.registerSkinElement	180
FListBox.removeAll	181
FListBox.removeItemAt	182
FListBox.replaceItemAt	182
FListBox.setAutoHideScrollbar	183

FListBox.setChangeHandler	184
FListBox.setDataProvider	185
FListBox.setEnabled	186
FListBox.setItemSymbol	187
FListBox.setRowCount	187
FListBox.setScrollPosition	188
FListBox.setSelectedIndex	188
FListBox.setSelectedIndices	189
FListBox.setSelectMultiple	189
FListBox.setSize	190
FListBox.setStyleProperty	191
FListBox.setWidth	191
FListBox.sortItemsBy	192
_focusrect	192
for	193
for..in	194
FPushButton (componente)	195
FPushButton.setEnabled	196
FPushButton.getLabel	196
FPushButton.registerSkinElement	197
FPushButton.setClickHandler	198
FPushButton.setEnabled	199
FPushButton.setLabel	199
FPushButton.setSize	200
FPushButton.setStyleProperty	200
FRadioButton (componente)	201
FRadioButton.getData	202
FRadioButton.setEnabled	203
FRadioButton.getLabel	203
FRadioButton.getState	204
FRadioButton.getValue	204
FRadioButton.registerSkinElement	205
FRadioButton.setChangeHandler	206
FRadioButton.setData	207
FRadioButton.setEnabled	208
FRadioButton.setGroupName	208
FRadioButton.setLabel	209
FRadioButton.setLabelPlacement	210
FRadioButton.setSize	210
FRadioButton.setState	211
FRadioButton.setStyleProperty	212
FRadioButton.setValue	212
FScrollBar (componente)	213
FScrollBar.setEnabled	214
FScrollBar.getScrollPosition	215
FScrollBar.registerSkinElement	215
FScrollBar.setChangeHandler	216
FScrollBar.setEnabled	218
FScrollBar.setHorizontal	218
FScrollBar.setLargeScroll	219

FScrollBar.setScrollContent	220
FScrollBar.setScrollPosition	220
FScrollBar.setScrollProperties	221
FScrollBar.setScrollTarget	221
FScrollBar.setSize	222
FScrollBar.setSmallScroll	222
FScrollBar.setStyleProperty	223
FScrollPane (componente)	224
FScrollPane.getPaneHeight	225
FScrollPane.getPaneWidth	225
FScrollPane.getScrollContent	226
FScrollPane.getScrollPosition	226
FScrollPane.loadScrollContent	227
FScrollPane.refreshPane	228
FScrollPane.registerSkinElement	229
FScrollPane.setDragContent	230
FScrollPane.setHScroll	230
FScrollPane.setScrollContent	231
FScrollPane.setScrollPosition	231
FScrollPane.setSize	232
FScrollPane.setStyleProperty	232
FScrollPane.setVScroll	233
FStyleFormat (oggetto)	234
Riepilogo delle proprietà valide per l'oggetto FStyleFormat	235
FStyleFormat.addListener	236
FStyleFormat.applyChanges	237
FStyleFormat.arrow	238
FStyleFormat.background	238
FStyleFormat.backgroundDisabled	239
FStyleFormat.check	239
FStyleFormat.darkshadow	240
FStyleFormat.face	240
FStyleFormat.foregroundDisabled	241
FStyleFormat.highlight	241
FStyleFormat.highlight3D	242
FStyleFormat.radioDot	242
FStyleFormat.removeListener	242
FStyleFormat.scrollTrack	243
FStyleFormat.selection	244
FStyleFormat.selectionDisabled	244
FStyleFormat.selectionUnfocused	245
FStyleFormat.shadow	245
FStyleFormat.textAlign	246
FStyleFormat.textBold	246
FStyleFormat.textColor	246
FStyleFormat.textDisabled	247
FStyleFormat.textFont	247
FStyleFormat.textIndent	248
FStyleFormat.textItalic	248
FStyleFormat.textLeftMargin	249

FStyleFormat.textRightMargin	249
FStyleFormat.textSelected	250
FStyleFormat.textSize	250
FStyleFormat.textUnderline	251
Function (oggetto)	251
Function.apply	251
Function.call	252
Function.prototype	253
fscommand	254
function	256
ge (maggiore di o uguale a; specifico per stringhe)	257
getProperty	257
getTimer	258
getURL	258
getVersion	259
_global	260
globalStyleFormat	260
gotoAndPlay	261
gotoAndStop	262
gt (maggiore di; specifico per stringhe)	262
_highquality	263
if	263
ifFrameLoaded	264
#include	265
#initclip	265
instanceof	266
int	267
isFinite	268
isNaN	268
Key (oggetto)	269
Key.addListener	270
Key.BACKSPACE	271
Key.CAPSLOCK	271
Key.CONTROL	271
Key.DELETEKEY	272
Key.DOWN	272
Key.END	272
Key.ENTER	272
Key.ESCAPE	273
Key.getAscii	273
Key.getCode	273
Key.HOME	274
Key.INSERT	274
Key.isDown	274
Key.isToggled	274
Key.LEFT	275
Key.onKeyDown	275
Key.onKeyUp	276
Key.PGDN	276
Key.PGUP	276

Key.removeListener	276
Key.RIGHT	277
Key.SHIFT	277
Key.SPACE	277
Key.TAB	277
Key.UP	278
le (minore di o uguale a; specifico per stringhe)	278
length	278
_level.	279
loadMovie	280
loadMovieNum	281
loadVariables	282
loadVariablesNum	283
LoadVars (oggetto)	284
LoadVars.contentType	286
LoadVars.getBytesLoaded	286
LoadVars.getBytesTotal	286
LoadVars.load	287
LoadVars.loaded	287
LoadVars.onLoad	287
LoadVars.send	288
LoadVars.sendAndLoad	289
LoadVars.toString	289
lt (minore di; specifico per stringhe)	290
Math (oggetto)	290
Math.abs	292
Math.acos	292
Math.asin	292
Math.atan	293
Math.atan2	293
Math.ceil	294
Math.cos	294
Math.E	294
Math.exp	295
Math.floor	295
Math.log	296
Math.LOG2E	296
Math.LOG10E	296
Math.LN2	297
Math.LN10	297
Math.max	298
Math.min	298
Math.PI	298
Math.pow	299
Math.random	299
Math.round	300
Math.sin	300
Math.sqrt	300
Math.SQRT1_2	301
Math.SQRT2	301

Math.tan	301
maxscroll	302
mbchr	302
mblength	303
mbord	303
mbsubstring	303
method	304
Mouse (oggetto)	304
Mouse.addListener	305
Mouse.hide	306
Mouse.onMouseDown	306
Mouse.onMouseMove	307
Mouse.onMouseUp	307
Mouse.removeListener	308
Mouse.show	308
MovieClip (oggetto)	308
MovieClip._alpha	312
MovieClip.attachMovie	312
MovieClip.beginFill	313
MovieClip.beginGradientFill	314
MovieClip.clear	318
MovieClip.createEmptyMovieClip	318
MovieClip.createTextField	319
MovieClip._currentframe	320
MovieClip.curveTo	321
MovieClip._droptarget	322
MovieClip.duplicateMovieClip	322
MovieClip.enabled	323
MovieClip.endFill	324
MovieClip.focusEnabled	324
MovieClip._focusrect	324
MovieClip._framesloaded	325
MovieClip.getBounds	325
MovieClip.getBytesLoaded	326
MovieClip.getBytesTotal	326
MovieClip.getDepth	327
MovieClip.getURL	327
MovieClip.globalToLocal	328
MovieClip.gotoAndPlay	328
MovieClip.gotoAndStop	329
MovieClip._height	330
MovieClip._highquality	330
MovieClip.hitArea	331
MovieClip.hitTest	331
MovieClip.lineStyle	332
MovieClip.lineTo	333
MovieClip.loadMovie	334
MovieClip.loadVariables	335
MovieClip.localToGlobal	336
MovieClip.moveTo	336

MovieClip._name	337
MovieClip.nextFrame	337
MovieClip.onData	338
MovieClip.onDragOut	338
MovieClip.onDragOver	339
MovieClip.onEnterFrame	339
MovieClip.onKeyDown	340
MovieClip.onKeyUp	340
MovieClip.onKillFocus	341
MovieClip.onLoad	341
MovieClip.onMouseDown	342
MovieClip.onMouseMove	342
MovieClip.onMouseUp	343
MovieClip.onPress	343
MovieClip.onRelease	344
MovieClip.onReleaseOutside	344
MovieClip.onRollOut	345
MovieClip.onRollOver	345
MovieClip.onSetFocus	346
MovieClip.onUnload	346
MovieClip._parent	347
MovieClip.play	347
MovieClip.prevFrame	348
MovieClip.removeMovieClip	348
MovieClip._rotation	348
MovieClip.setMask	349
MovieClip._soundbuftime	349
MovieClip.startDrag	350
MovieClip.stop	350
MovieClip.stopDrag	351
MovieClip.swapDepths	351
MovieClip.tabChildren	352
MovieClip.tabEnabled	352
MovieClip.tabIndex	353
MovieClip._target	353
MovieClip._totalframes	353
MovieClip.trackAsMenu	354
MovieClip.unloadMovie	354
MovieClip._url	354
MovieClip.useHandCursor	355
MovieClip._visible	355
MovieClip._width	355
MovieClip._x	356
MovieClip._xmouse	356
MovieClip._xscale	356
MovieClip._y	357
MovieClip._ymouse	357
MovieClip._yscale	357
NaN	358
ne (non uguale; specifico per stringhe)	358

new	358
newline	359
nextFrame	360
nextScene	360
not	361
null	361
Number (funzione)	362
Number (oggetto)	362
Number.MAX_VALUE	364
Number.MIN_VALUE	364
Number.NaN	364
Number.NEGATIVE_INFINITY	364
Number.POSITIVE_INFINITY	365
Number.toString	365
Number.valueOf	365
Object (oggetto)	366
Object.addProperty	367
Object.__proto__	368
Object.registerClass	369
Object.toString	371
Object.unwatch	371
Object.valueOf	372
Object.watch	372
onClipEvent	374
on	375
or	376
ord	377
_parent	377
parseFloat	378
parseInt	378
play	379
prevFrame	380
prevScene	380
print	381
printAsBitmap	382
printAsBitmapNum	383
printNum	384
_quality	385
random	386
removeMovieClip	387
return	387
_root	388
scroll	389
Selection (oggetto)	389
Selection.addListener	390
Selection.getBeginIndex	390
Selection.getCaretIndex	391
Selection.getEndIndex	391
Selection.getFocus	391
Selection.onSetFocus	392

Selection.removeListener	392
Selection.setFocus	393
Selection.setSelection	393
set variable	394
setInterval	395
setProperty	397
Sound (oggetto)	397
Sound.attachSound	399
Sound.duration	399
Sound.getBytesLoaded	400
Sound.getBytesTotal	400
Sound.getPan	400
Sound.getTransform	401
Sound.getVolume	401
Sound.loadSound	402
Sound.onLoad	402
Sound.onSoundComplete	403
Sound.position	403
Sound.setPan	404
Sound.setTransform	405
Sound.setVolume	407
Sound.start	407
Sound.stop	408
_soundbuftime	408
Stage (oggetto)	409
Stage.addListener	409
Stage.align	410
Stage.height	411
Stage.onResize	411
Stage.removeListener	411
Stage.scaleMode	412
Stage.width	412
startDrag	412
stop	413
stopAllSounds	413
stopDrag	414
String (funzione)	415
" " (delimitatore di stringa)	415
String (oggetto)	416
String.charAt	418
String.charCodeAt	418
String.concat	419
String.fromCharCode	419
String.indexOf	419
String.lastIndexOf	420
String.length	420
String.slice	421
String.split	421
String.substr	422
String.substring	423

String.toLowerCase	423
String.toUpperCase	424
substring	424
super	424
switch	426
System (oggetto)	427
System.capabilities (oggetto)	427
System.capabilities.hasAudioEncoder	428
System.capabilities.hasAccessibility	428
System.capabilities.hasAudio	429
System.capabilities.hasMP3	429
System.capabilities.language	429
System.capabilities.manufacturer	430
System.capabilities.os	430
System.capabilities.pixelAspectRatio	431
System.capabilities.screenColor	431
System.capabilities.screenDPI	431
System.capabilities.screenResolution.x	431
System.capabilities.screenResolution.y	432
System.capabilities.version	432
System.capabilities.hasVideoEncoder	432
targetPath	432
tellTarget	433
TextField (oggetto)	434
TextField._alpha	437
TextField.addListener	437
TextField.autoSize	438
TextField.background	438
TextField.backgroundColor	438
TextField.border	439
TextField.borderColor	439
TextField.bottomScroll	439
TextField.embedFonts	440
TextField._focusrect	440
TextField.getDepth	440
TextField.getFontList	440
TextField.getNewTextFormat	441
TextField.getTextFormat	441
TextField._height	442
TextField._highquality	442
TextField.hscroll	443
TextField.html	443
TextField.htmlText	443
TextField.length	445
TextField.maxChars	445
TextField.maxhscroll	445
TextField.maxscroll	445
TextField.multiline	446
TextField._name	446
TextField.onChanged	446

TextField.onKillFocus	446
TextField.onScroller.	447
TextField.onSetFocus.	447
TextField._parent	448
TextField.password	448
TextField._quality	448
TextField.removeListener.	449
TextField.removeTextField.	449
TextField.replaceSel.	450
TextField.restrict	450
TextField._rotation	451
TextField.scroll	451
TextField.selectable	452
TextField.setNewTextFormat.	452
TextField.setTextFormat	452
TextField._soundbuftime.	453
TextField.tabEnabled.	454
TextField.tabIndex.	454
TextField._target	455
TextField.text	455
TextField.textColor	455
TextField.textHeight	456
TextField.textWidth	456
TextField.type	456
TextField._url	456
TextField.variable	457
TextField._visible.	457
TextField._width	457
TextField.wordWrap	458
TextField._x.	458
TextField._xmouse.	458
TextField._xscale	459
TextField._y.	459
TextField._ymouse	459
TextField._yscale	460
TextFormat (oggetto).	460
TextFormat.align	462
TextFormat.blockIndent	462
TextFormat.bold	463
TextFormat.bullet	463
TextFormat.color.	463
TextFormat.font	463
TextFormat.getTextExtent	464
TextFormat.indent.	464
TextFormat.italic	464
TextFormat.leading	465
TextFormat.leftMargin	465
TextFormat.rightMargin	465
TextFormat.size	465
TextFormat.tabStops	466

TextFormat.target	466
TextFormat.underline	466
TextFormat.url	466
this	467
toggleHighQuality	467
trace	468
true	469
typeof	469
undefined	470
unescape	471
unloadMovie	472
unloadMovieNum	473
updateAfterEvent	473
var	474
void	474
while	475
with	476
XML (oggetto)	478
XML.appendChild	481
XML.attributes	481
XML.childNodes	482
XML.cloneNode	482
XML.contentType	483
XML.createElement	483
XML.createTextNode	484
XML.docTypeDecl	484
XML.firstChild	485
XML.getBytesLoaded	485
XML.getBytesTotal	485
XML.hasChildNodes	486
XML.ignoreWhite	486
XML.insertBefore	487
XML.lastChild	487
XML.load	487
XML.loaded	488
XML.nextSibling	488
XML.nodeName	489
XML.nodeType	489
XML.nodeValue	489
XML.onData	490
XML.onLoad	491
XML.parentNode	492
XML.parseXML	492
XML.previousSibling	492
XML.removeNode	493
XML.send	493
XML.sendAndLoad	493
XML.status	494
XML.toString	495
XML.xmlDecl	495

XMLSocket (oggetto)	496
XMLSocket.close.	498
XMLSocket.connect	498
XMLSocket.onClose	499
XMLSocket.onConnect.	500
XMLSocket.onData	501
XMLSocket.onXML	502
XMLSocket.close.	503

Dizionario di ActionScript

Questo dizionario contiene la descrizione della sintassi e dell'uso degli elementi di ActionScript in Macromedia Flash MX. Per usare gli esempi in uno script, è sufficiente copiare il codice di esempio dal Dizionario di ActionScript e incollarlo nel pannello Azioni in Modalità esperto.

Nel dizionario sono elencati tutti gli elementi di ActionScript: operatori, parole chiave, istruzioni, azioni, proprietà, funzioni, oggetti, componenti e metodi. Per una panoramica generale delle voci del dizionario, consultare "Contenuto del dizionario" a pagina 23. Le tabelle contenute in questa sezione offrono un buon punto di partenza per la ricerca di metodi od operatori simbolici, quando non si conosce la classe dell'oggetto o del componente.

ActionScript è conforme allo standard ECMA-262, la specifica elaborata dall'Associazione europea dei produttori di computer, se non indicato diversamente. Alcuni elementi di ActionScript di Flash 5 e versioni precedenti sono diventati obsoleti e sono stati sostituiti con elementi conformi a tale standard. Sebbene gli elementi obsoleti siano ancora supportati da Flash Player 5, si consiglia di usare i nuovi elementi di Flash MX.

Il dizionario contiene due tipi di voci:

- Singole voci relative a operatori, parole chiave, funzioni, variabili, proprietà, metodi e istruzioni
- Voci relative agli oggetti e ai componenti che forniscono informazioni generali sugli oggetti incorporati e sui componenti Flash

Le informazioni fornite nelle voci di esempio consentono di interpretare la struttura e le convenzioni usate nei due tipi di voci.

Voci di esempio per la maggior parte degli elementi di ActionScript

Di seguito vengono descritte le convenzioni adottate per gli elementi di ActionScript che non siano oggetti o componenti.

Titolo della voce

Tutte le voci sono elencate in ordine alfabetico senza tenere conto dell'uso delle maiuscole, dei caratteri di sottolineatura iniziali e così via.

Disponibilità

In questa sezione viene indicata la versione di Flash Player che supporta l'elemento in esame. Non si tratta della stessa versione di Flash usata per creare contenuti. Se, ad esempio, si creano contenuti per Flash Player 5 mediante Flash MX, è possibile usare solo gli elementi di ActionScript disponibili in Flash Player 5.

Uso

In questa sezione viene fornita la sintassi corretta per l'uso degli elementi di ActionScript nel codice. La parte di sintassi richiesta è indicata da un *carattere codice*, mentre il codice inserito dall'utente è in *carattere codice corsivo*. I parametri opzionali sono racchiusi tra parentesi quadre ([]).

Parametri

In questa sezione vengono descritti i parametri elencati nella sintassi.

Valori restituiti

In questa sezione vengono indicati gli eventuali valori restituiti dall'elemento.

Descrizione

In questa sezione viene identificato il tipo di elemento (ad esempio un operatore, un metodo, una funzione e così via) con la descrizione della relativa modalità d'uso.

Esempio

Questa sezione fornisce un esempio di codice illustrativo sull'uso dell'elemento.

Vedere anche

In questa sezione sono elencate le voci del Dizionario di ActionScript correlate.

Voci di esempio per oggetti e componenti

Di seguito vengono descritte le convenzioni adottate per i componenti e gli oggetti incorporati di ActionScript. Gli oggetti e i componenti sono elencati in ordine alfabetico insieme a tutti gli altri elementi del dizionario. I componenti Flash sono elencati come FCheckBox, FComboBox e così via.

Titolo della voce

Il titolo della voce indica il nome dell'oggetto o del componente, seguito da un paragrafo con informazioni di carattere generale sull'oggetto o sul componente corrispondente.

Tabelle di riepilogo di metodi e proprietà

Tutte le voci degli oggetti o dei componenti contengono una tabella che elenca tutti i metodi associati. Le eventuali proprietà (spesso costanti) di cui dispone l'oggetto o il componente sono elencate in una tabella aggiuntiva. Tutti i metodi e le proprietà elencati in queste tabelle hanno voci di dizionario specifiche, correlate alla voce principale dell'oggetto o del componente.

Funzione di costruzione

Se un oggetto o un componente richiede una funzione di costruzione per accedere ai relativi metodi e proprietà, questa funzione è descritta per ciascuna voce. Nella descrizione sono inclusi tutti gli elementi standard (sintassi, descrizione e così via) presenti nelle altre voci del dizionario.

Elenco dei metodi e delle proprietà

I metodi e le proprietà di un oggetto o di un componente sono elencati in ordine alfabetico sotto la voce principale dell'oggetto o del componente.

Contenuto del dizionario

Tutte le voci del dizionario sono elencate in ordine alfabetico. Gli operatori rappresentati da simboli vengono elencati nell'ordine ASCII. Inoltre, i metodi associati a un oggetto (o a un componente) sono elencati accanto al nome corrispondente, ad esempio il metodo `abs` di un oggetto matematico viene elencato come `Math.abs`, il metodo `getValue` del componente `FRadioButton` viene elencato come `FRadioButton.getValue`.

Le due tabelle riportate di seguito agevolano l'individuazione di questi elementi. La prima tabella elenca gli operatori simbolici nell'ordine in cui vengono presentati nel dizionario. La seconda tabella elenca tutti gli altri elementi di `ActionScript`.

Nota: per l'ordine di precedenza e associatività degli operatori, consultare l'Appendice B "Precedenza e associatività degli operatori" nella "Guida all'uso di Flash".

Operatori simbolici

--	-- (decremento)
++	++ (incremento)
!	! (NOT logico)
!=	!=(diseguaglianza)
!==	!==(diseguaglianza rigorosa)
%	% (modulo)
%=	%= (assegnazione modulo)
&	& (AND bit a bit)
&&	&& (cortocircuito AND)
&=	&= (assegnazione AND bit a bit)
()	() (parentesi tonde)
-	- (meno)
*	* (moltiplicazione)
*=	*= (assegnazione moltiplicazione)
,	, (virgola)
.	. (punto)
?:	?: (condizionale)
/	/ (divisione)
//	// (delimitatore di commento)
/*	/* (delimitatore di commento)
/=	/= (assegnazione divisione)
[]	[] (accesso matrice)
^	^(XOR bit a bit)

Operatori simbolici

<code>^=</code>	<code>^=</code> (assegnazione XOR bit a bit)
<code>{}</code>	<code>{}</code> (operatore di inizializzazione degli oggetti)
<code> </code>	<code> </code> (OR bit a bit)
<code> </code>	<code> </code> (OR logico)
<code> =</code>	<code> =</code> (assegnazione OR bit a bit)
<code>~</code>	<code>~</code> (NOT bit a bit)
<code>+</code>	<code>+</code> (addizione)
<code>+=</code>	<code>+=</code> (assegnazione addizione)
<code><</code>	<code><</code> (minore di)
<code><<</code>	<code><<</code> (spostamento a sinistra bit a bit)
<code><<=</code>	<code><<=</code> (spostamento a sinistra bit a bit e assegnazione)
<code><=</code>	<code><=</code> (minore o uguale a)
<code><></code>	<code><></code> (disuguaglianza)
<code>=</code>	<code>=</code> (assegnazione)
<code>-=</code>	<code>-=</code> (assegnazione sottrazione)
<code>==</code>	<code>==</code> (uguaglianza)
<code>===</code>	<code>===</code> (uguaglianza rigorosa)
<code>></code>	<code>></code> (maggiore di)
<code>>=</code>	<code>>=</code> (maggiore di o uguale a)
<code>>></code>	<code>>></code> (spostamento a destra bit a bit)
<code>>>=</code>	<code>>>=</code> (spostamento a destra bit a bit e assegnazione)
<code>>>></code>	<code>>>></code> (spostamento a destra senza segno bit a bit)
<code>>>>=</code>	<code>>>>=</code> (spostamento a destra senza segno bit a bit e assegnazione)

La tabella seguente elenca tutti gli elementi di `ActionScript` che non sono operatori simbolici.

Elemento di <code>ActionScript</code>	Vedere la voce
<code>abs</code>	<code>Math.abs</code>
<code>acos</code>	<code>Math.acos</code>
<code>add</code>	<code>add</code>
<code>addItem</code>	<code>FComboBox.addItem</code> , <code>FListBox.addItem</code>
<code>addItemAt</code>	<code>FComboBox.addItemAt</code> , <code>FListBox.addItem</code>
<code>addListener</code>	<code>FStyleFormat.addListener</code> , <code>Key.addListener</code> , <code>Mouse.addListener</code> , <code>Selection.addListener</code> , <code>Stage.addListener</code> , <code>TextField.addListener</code>
<code>addProperty</code>	<code>Object.addProperty</code>
<code>and</code>	<code>and</code>

Elemento di ActionScript	Vedere la voce
align	Stage.align, TextFormat.align
_alpha	MovieClip._alpha, Button._alpha, TextField._alpha
appendChild	XML.appendChild
apply	Function.apply
applyChanges	FStyleFormat.applyChanges
Arguments	arguments (oggetto)
Array	Array (oggetto)
arrow	FStyleFormat.arrow
asfunction	asfunction
asin	Math.asin
atan	Math.atan
atan2	Math.atan2
attachMovie	MovieClip.attachMovie
attachSound	Sound.attachSound
attributes	XML.attributes
autosize	TextField.autoSize
background	FStyleFormat.background, TextField.background
backgroundColor	TextField.backgroundColor
backgroundDisabled	FStyleFormat.backgroundDisabled
BACKSPACE	Key.BACKSPACE
beginFill	MovieClip.beginFill
beginGradientFill	MovieClip.beginGradientFill
blockIndent	TextFormat.blockIndent
bold	TextFormat.bold
Valore booleano	Boolean (funzione), Boolean (oggetto)
border	TextField.border
borderColor	TextField.borderColor
bottomScroll	TextField.bottomScroll
break	break
bullet	TextFormat.bullet
Button	Button (oggetto)
call	call, Function.call
call function	call (funzione)
callee	arguments.callee
caller	arguments.caller
capabilities	System.capabilities (oggetto)

Elemento di ActionScript	Vedere la voce
CAPSLOCK	Key.CAPSLOCK
case	case
ceil	Math.ceil
charAt	String.charAt
charCodeAt	String.charCodeAt
check	FStyleFormat.check
childNodes	XML.childNodes
chr	chr
clear	MovieClip.clear
clearInterval	clearInterval
cloneNode	XML.cloneNode
close	XMLSocket.close
Color	Color (oggetto),TextFormat.color
concat	Array.concat,String.concat
connect	XMLSocket.connect
funzione di costruzione	Array (oggetto),Boolean (oggetto),Color (oggetto),Date (oggetto),Number (funzione),Object (oggetto),Sound (oggetto),String (oggetto),XML (oggetto),XMLSocket (oggetto)
contentType	LoadVars.contentType, XML.contentType
continue	continue
CONTROL	Key.CONTROL
cos	Math.cos
createElement	XML.createElement
createEmptyMovieClip	MovieClip.createEmptyMovieClip
createTextField	MovieClip.createTextField
createTextNode	XML.createTextNode
_currentframe	MovieClip._currentframe
curveTo	MovieClip.curveTo
Date	Date (oggetto)
darkshadow	FStyleFormat.darkshadow
default	default
delete	delete
DELETEKEY	Key.DELETEKEY
docTypeDecl	XML.docTypeDecl
do while	do while
DOWN	Key.DOWN
_droptarget	MovieClip._droptarget

Elemento di ActionScript	Vedere la voce
duplicateMovieClip	duplicateMovieClip, MovieClip.duplicateMovieClip
duration	Sound.duration
E	Math.E
#endinitclip	#endinitclip
else	else
else if	else if
embedFonts	TextField.embedFonts
enabled	Button.enabled, MovieClip.enabled
END	Key.END
endFill	MovieClip.endFill
ENTER	Key.ENTER
eq	eq (uguale; specifico per stringhe)
escape (funzione)	escape
ESCAPE (costante)	Key.ESCAPE
eval	eval
evaluate	evaluate
exp	Math.exp
face	FStyleFormat.face
false	false
FCheckBox	FCheckBox (componente)
FComboBox	FListBox (componente)
firstChild	XML.firstChild
FListBox	FListBox (componente)
floor	Math.floor
focusEnabled	MovieClip.focusEnabled
_focusrect	_focusrect, Button._focusrect, TextField._focusrect, MovieClip._focusrect
carattere	TextFormat.font
for	for
for..in	for..in
foregroundDisabled	FStyleFormat.foregroundDisabled
FPushButton	FPushButton (componente)
FRadioButton	FPushButton (componente)
_framesloaded	MovieClip._framesloaded
fromCharCode	String.fromCharCode
fscommand	fscommand

Elemento di ActionScript	Vedere la voce
FScrollBar	FScrollBar (componente)
FScrollPane	FScrollPane (componente)
FStyleFormat	FStyleFormat (oggetto)
function	function, Function (oggetto)
ge	ge (maggiore di o uguale a; specifico per stringhe)
get	CustomActions.get
getAscii	Key.getAscii
getBeginIndex	Selection.getBeginIndex
getBounds	MovieClip.getBounds
getBytesLoaded	LoadVars.getBytesLoaded, MovieClip.getBytesLoaded, Sound.getBytesLoaded, XML.getBytesLoaded
getBytesTotal	LoadVars.getBytesTotal, MovieClip.getBytesTotal, Sound.getBytesTotal, XML.getBytesTotal
getCaretIndex	Selection.getCaretIndex
getCode	Key.getCode
getData	FRadioButton.getData
getDate	Date.getDate
getDay	Date.getDay
getDepth	Button.getDepth, MovieClip.getDepth, TextField.getDepth
getEnabled	FCheckBox.getEnabled, FComboBox.getEnabled, FListBox.getEnabled, FPushButton.getEnabled, FRadioButton.getEnabled, FScrollBar.getEnabled
getEndIndex	Selection.getEndIndex
getFocus	Selection.getFocus
getFontList	TextField.getFontList
getFullYear	Date.getFullYear
getHours	Date.getHours
getItemAt	FComboBox.getItemAt, FListBox.addItemAt
getLabel	FCheckBox.getLabel, FPushButton.getLabel, FRadioButton.getLabel
getLength	FComboBox.getLength, FListBox.getLength
getMilliseconds	Date.getMilliseconds
getMinutes	Date.getMinutes
getMonth	Date.getMonth
getNewTextFormat	TextField.getNewTextFormat
getPan	Sound.getPan
getPaneHeight	FScrollPane.getPaneHeight
getPaneWidth	FScrollPane.getPaneWidth

Elemento di ActionScript	Vedere la voce
getProperty	getProperty
getRowCount	FComboBox.getRowCount, FListBox.getRowCount
getRGB	Color.getRGB
getScrollContent	FScrollPane.getScrollContent
getScrollPosition	FComboBox.getScrollPosition, FListBox.getScrollPosition, FScrollBar.getScrollPosition, FScrollPane.getScrollPosition
getSeconds	Date.getSeconds
getSelectedIndex	FComboBox.getSelectedIndex, FListBox.getSelectedIndex
getSelectedIndices	FListBox.getSelectedIndices
getSelectedItem	FComboBox.getSelectedItem, FListBox.getSelectedItem
getSelectedItems	FListBox.getSelectedItem
getSelectMultiple	FListBox.getSelectMultiple
getState	FRadioButton.getState
getTextExtent	TextFormat.getTextExtent
getTextFormat	TextField.getTextFormat
getTime	Date.getTime
getTimer	getTimer
getTimezoneOffset	Date.getTimezoneOffset
getTransform	Color.getTransform, Sound.getTransform
getURL	getURL, MovieClip.getURL
getUTCDate	Date.getUTCDate
getUTCDay	Date.getUTCDay
getUTCFullYear	Date.getUTCFullYear
getUTCHours	Date.getUTCHours
getUTCMilliseconds	Date.getUTCMilliseconds
getUTCMinutes	Date.getUTCMinutes
getUTCMonth	Date.getUTCMonth
getUTCSeconds	Date.getUTCSeconds
getValue	FCheckBox.getValue, FComboBox.getValue, FListBox.getValue, FRadioButton.getValue
getVersion	getVersion
getVolume	Sound.getVolume
getYear	Date.getYear
_global	_global
globalStyleFormat	globalStyleFormat
globalToLocal	MovieClip.globalToLocal
goto	gotoAndPlay, gotoAndStop

Elemento di ActionScript	Vedere la voce
gotoAndPlay	gotoAndPlay, MovieClip.gotoAndPlay
gotoAndStop	gotoAndStop, MovieClip.gotoAndStop
gt	gt (maggiore di; specifico per stringhe)
hasAccessibility	System.capabilities.hasAccessibility
hasAudio	System.capabilities.hasAudio
hasAudioEncoder	System.capabilities.hasAudioEncoder
hasMP3	System.capabilities.hasMP3
hasVideoEncoder	System.capabilities.hasVideoEncoder
hasChildNodes	XML.hasChildNodes
height	Stage.height
_height	MovieClip._height, TextField._height, Button._height
hide	Mouse.hide
highlight	FStyleFormat.highlight
highlight3D	FStyleFormat.highlight3D
_highquality	_highquality, Button._highquality, MovieClip._highquality, TextField._highquality
hitArea	MovieClip.hitArea
hitTest	MovieClip.hitTest
HOME	Key.HOME
hscroll	TextField.hscroll
html	TextField.html
htmlText	TextField.htmlText
if	if
ifFrameLoaded	ifFrameLoaded
ignoreWhite	XML.ignoreWhite
#include	#include
indent	TextFormat.indent
indexOf	String.indexOf
#initclip	#initclip
INSERT	Key.INSERT
insertBefore	XML.insertBefore
install	CustomActions.install
instanceof	instanceof
int	int
isActive	Accessibility.isActive
isDown	Key.isDown

Elemento di ActionScript	Vedere la voce
isFinite	isFinite
isNaN	isNaN
isToggled	Key.isToggled
italic	TextFormat.italic
join	Array.join
Tasto	Key (oggetto)
language	System.capabilities.language
lastChild	XML.lastChild
lastIndexOf	String.lastIndexOf
le	le (minore di o uguale a; specifico per stringhe)
leading	TextFormat.leading
LEFT	Key.LEFT
leftMargin	TextFormat.leftMargin
length	arguments.length, Array.length, String.length, Sound.loadSound, TextField.length
level	_level
lineStyle	MovieClip.lineStyle
lineTo	MovieClip.lineTo
list	CustomActions.uninstall
LN2	Math.LN2
LN10	Math.LN10
load	XML.load, LoadVars.load
loaded	XML.loaded, LoadVars.loaded
loadMovie	loadMovie, MovieClip.loadMovie
loadMovieNum	loadMovieNum
loadScrollContent	FScrollPane.loadScrollContent
loadSound	Sound.loadSound
loadVariables	loadVariables, MovieClip.loadVariables
loadVariablesNum	loadVariablesNum
LoadVars	LoadVars (oggetto)
localToGlobal	MovieClip.localToGlobal
log	Math.log
LOG2E	Math.LOG2E
LOG10E	Math.LOG10E
lt	lt (minore di; specifico per stringhe)
manufacturer	System.capabilities.manufacturer

Elemento di ActionScript	Vedere la voce
Math	Math (oggetto)
max	Math.max
maxChars	TextField.maxChars
maxhscroll	TextField.maxhscroll
maxscroll	maxscroll, TextField.maxscroll
MAX_VALUE	Number.MAX_VALUE
mbchr	mbchr
mblength	mblength
mbord	mbord
mbsubstring	mbsubstring
metodo	method
min	Math.min
MIN_VALUE	Number.MIN_VALUE
Mouse	Mouse (oggetto)
moveTo	MovieClip.moveTo
MovieClip	MovieClip (oggetto)
multiline	TextField.multiline
_name	MovieClip._name, TextField._name, Button._name
NaN	NaN, Number.NaN
ne	ne (non uguale; specifico per stringhe)
NEGATIVE_INFINITY	Number.NEGATIVE_INFINITY
new (operatore)	new
newline	newline
nextFrame	nextFrame, MovieClip.nextFrame
nextScene	nextScene
nextSibling	XML.nextSibling
nodeName	XML.nodeName
nodeType	XML.nodeType
nodeValue	XML.nodeValue
not	not
null	null
Numero	Number (funzione), Number (oggetto)
Object	Object (oggetto)
on	on
onClipEvent	onClipEvent
onClose	XMLSocket.onClose

Elemento di ActionScript	Vedere la voce
onChanged	TextField.onChanged
onConnect	XMLSocket.onConnect
onData	XML.onData, XMLSocket.onData
onDragOut	Button.onDragOut, MovieClip.onDragOut
onDragOver	Button.onDragOver, MovieClip.onDragOver
onEnterFrame	MovieClip.onEnterFrame
onKeyDown	Button.onKeyDown, Key.onKeyDown, MovieClip.onKeyDown
onKeyUp	Button.onKeyUp, Key.onKeyUp, MovieClip.onKeyUp
onKillFocus	Button.onKillFocus, MovieClip.onKillFocus, TextField.onKillFocus
onLoad	LoadVars.onLoad, MovieClip.onLoad, Sound.onLoad, XML.onLoad
onMouseDown	Mouse.onMouseDown, MovieClip.onMouseDown
onMouseMove	Mouse.onMouseMove, MovieClip.onMouseMove
onMouseUp	Mouse.onMouseUp, MovieClip.onMouseUp
onPress	Button.onPress, MovieClip.onPress
onRelease	Button.onRelease, MovieClip.onRelease
onReleaseOutside	Button.onReleaseOutside, MovieClip.onReleaseOutside
onResize	Stage.onResize
onRollOut	Button.onRollOut
onRollOver	Button.onRollOver
onScroller	TextField.onScroller
onSetFocus	Button.onSetFocus, MovieClip.onSetFocus, Selection.onSetFocus, TextField.onSetFocus
onSort	Array.pop
onSoundComplete	Sound.onSoundComplete
onUnload	MovieClip.onUnload
onXML	XMLSocket.onXML
or (OR logico)	or
ord	ord
os	System.capabilities.os
_parent	_parent, Button._parent, MovieClip._parent, TextField._parent
parentNode	XML.parentNode
parseFloat	parseFloat
parseInt	parseInt
parseXML	XML.parseXML
password	TextField.password
PGDN	Key.PGDN

Elemento di ActionScript	Vedere la voce
PGUP	Key.PGUP
PI	Math.PI
pixelAspectRatio	System.capabilities.pixelAspectRatio
play	play, MovieClip.play
pop	Array.pop
position	Sound.position
POSITIVE_INFINITY	Number.POSITIVE_INFINITY
pow	Math.pow
prevFrame	prevFrame, MovieClip.prevFrame
previousSibling	XML.previousSibling
prevScene	prevScene
print	print
printAsBitmap	printAsBitmap
printAsBitmapNum	printAsBitmapNum
printNum	printNum
__proto__	Object.__proto__
push	Array.push
_quality	_quality, TextField._quality, Button._quality
radioDot	FStyleFormat.radioDot
random	random, Math.random
refreshPane	FScrollPane.refreshPane
registerClass	Object.registerClass
registerSkinElement	FCheckBox.registerSkinElement, FComboBox.registerSkinElement, FListBox.registerSkinElement, FPushButton.registerSkinElement, FRadioButton.registerSkinElement, FScrollPane.registerSkinElement, FScrollPane.registerSkinElement
removeAll	FComboBox.removeAll, FListBox.removeAll
removeItemAt	FComboBox.removeItemAt, FListBox.removeItemAt
removeListener	FStyleFormat.removeListener, Key.removeListener, Mouse.removeListener, Selection.removeListener, Stage.removeListener, TextField.removeListener
removeMovieClip	removeMovieClip, MovieClip.removeMovieClip
removeNode	XML.removeNode
removeTextField	TextField.removeTextField
replaceItemAt	FComboBox.replaceItemAt, FListBox.replaceItemAt
replaceSel	TextField.replaceSel
resolutionX	Capabilities.screenResolutionX

Elemento di ActionScript	Vedere la voce
resolutionY	Capabilities.screenResolutionY
restrict	TextField.restrict
return	return
reverse	Array.reverse
RIGHT	Key.RIGHT
rightMargin	TextFormat.rightMargin
_root	_root
_rotation	MovieClip._rotation, Button._rotation, TextField._rotation
round	Math.round
scaleMode	Stage.scaleMode
screenColor	System.capabilities.screenColor
screenDPI	System.capabilities.screenDPI
screenResolution.x	System.capabilities.screenResolution.x
screenResolution.y	System.capabilities.screenResolution.y
scroll	scroll, TextField.scroll
scrollTrack	FStyleFormat.scrollTrack
selectable	TextField.selectable
selection	FStyleFormat.selection
Selection	Selection (oggetto)
selectionDisabled	FStyleFormat.selectionDisabled
selectionUnfocused	FStyleFormat.selectionUnfocused
send	LoadVars.send, XML.send, XMLSocket.close
sendAndLoad	LoadVars.sendAndLoad, XML.sendAndLoad
set variable	set variable
setAutoHideScrollBar	FListBox.setAutoHideScrollbar
setChangeHandler	FCheckBox.setChangeHandler, FComboBox.setChangeHandler, FListBox.setChangeHandler, FRadioButton.setChangeHandler, FScrollBar.setChangeHandler
setClickHandler	FPushButton.setClickHandler
setData	FRadioButton.setData
setDataProvider	FComboBox.setDataProvider, FListBox.setDataProvider
setDate	Date.setDate
setDragContent	FScrollPane.setDragContent
setEditable	FComboBox.setEditable
setEnabled	FCheckBox.setEnabled, FComboBox.setEnabled, FListBox.setEnabled, FPushButton.setEnabled, FRadioButton.setEnabled, FScrollBar.setEnabled

Elemento di ActionScript	Vedere la voce
setFocus	Selection.setFocus
setFullYear	Date.setFullYear
setGroupName	FRadioButton.setGroupName
setHorizontal	FScrollBar.setHorizontal
setHours	Date.setHours
setHScroll	FScrollPane.setHScroll
setInterval	setInterval
setItemSymbol	FComboBox.setItemSymbol, FListBox.setItemSymbol
setLabel	FCheckBox.setLabel, FPushButton.setLabel, FRadioButton.setLabel
setLabelPlacement	FCheckBox.setLabelPlacement, FRadioButton.setLabelPlacement
setLargeScroll	FScrollBar.setLargeScroll
setMask	MovieClip.setMask
setMilliseconds	Date.setMilliseconds
setMinutes	Date.setMinutes
setMonth	Date.setMonth
setNewTextFormat	TextField.setNewTextFormat
setPan	Sound.setPan
setProperty	setProperty
setRGB	Color.setRGB
setRowCount	FComboBox.setRowCount, FListBox.setRowCount
setScrollContent	FScrollBar.setSize, FScrollPane.setScrollContent
setScrollPosition	FListBox.setScrollPosition, FScrollBar.setScrollPosition, FScrollPane.setScrollPosition
setScrollProperties	FScrollBar.setScrollProperties
setScrollTarget	FScrollBar.setScrollTarget
setSeconds	Date.setSeconds
setSelectedIndex	FComboBox.setSelectedIndex, FListBox.setSelectedIndex
setSelectedIndices	FListBox.setSelectedIndices
setSelection	Selection.setSelection
setSelectMultiple	FListBox.setSelectMultiple
setSize	FCheckBox.setSize, FComboBox.setSize, FListBox.setSize, FPushButton.setSize, FScrollBar.setSize, FScrollPane.setSize
setSmallScroll	FScrollBar.setSmallScroll
setState	FRadioButton.setState

Elemento di ActionScript	Vedere la voce
setStyleProperty	FCheckBox.setStyleProperty, FComboBox.setStyleProperty, FListBox.setStyleProperty, FPushButton.setStyleProperty, FRadioButton.setStyleProperty, FScrollBar.setStyleProperty, FScrollPane.setStyleProperty
setTextFormat	TextField.setTextFormat
setTime	Date.setTime
setTransform	Color.setTransform, Sound.setTransform
setUTCDate	Date.setUTCDate
setUTCFullYear	Date.setUTCFullYear
setUTCHours	Date.setUTCHours
setUTCMilliseconds	Date.setUTCMilliseconds
setUTCMinutes	Date.setUTCMinutes
setUTCMonth	Date.setUTCMonth
setUTCSeconds	Date.setUTCSeconds
setValue	FCheckBox.setValue, FComboBox.setValue, FRadioButton.setValue
setVolume	Sound.setVolume
setVScroll	FScrollPane.setVScroll
setWidth	FListBox.setWidth
setYear	Date.setYear
shadow	FStyleFormat.shadow
shift (metodo)	Array.shift
SHIFT (costante)	Key.SHIFT
show	Mouse.show
sin	Math.sin
size	TextFormat.size
slice	Array.splice, String.slice
sort	Array.sort
sortItemsBy	FComboBox.sortItemsBy, FListBox.sortItemsBy
Audio	Sound (oggetto)
_soundbuftime	_soundbuftime, TextField._soundbuftime, MovieClip._soundbuftime, Button._soundbuftime
SPACE	Key.SPACE
splice	Array.splice
split	String.split
sqrt	Math.sqrt
SQRT1_2	Math.SQRT1_2
SQRT2	Math.SQRT2

Elemento di ActionScript	Vedere la voce
start	Sound.start
startDrag	startDrag, MovieClip.startDrag
status	XML.status
stop	stop, MovieClip.stop, Sound.stop
stopAllSounds	stopAllSounds
stopDrag	stopDrag, MovieClip.stopDrag
Stringa	String (funzione), String (oggetto)
substr	String.substring
substring	substring, String.substring
super	super
swapDepths	MovieClip.swapDepths
switch	switch
System	System (oggetto)
TAB	Key.TAB
tabChildren	MovieClip.tabChildren
tabEnabled	Button.tabEnabled, TextField.tabEnabled, MovieClip.tabEnabled
tabIndex	Button.tabIndex, MovieClip.tabIndex, TextField.tabIndex
tabStops	TextFormat.tabStops
tan	Math.tan
target	TextFormat.target
_target	Button._target, MovieClip._target, TextField._target
targetPath	targetPath
tellTarget	tellTarget
testo	TextField.text
textAlign	FStyleFormat.textAlign
textBold	FStyleFormat.textBold
textColor	FStyleFormat.textColor, TextField.textColor
textDisabled	FStyleFormat.textDisabled
TextField	TextField (oggetto)
textFont	FStyleFormat.textFont
TextFormat	TextFormat (oggetto)
textHeight	TextField.textHeight
textIndent	FStyleFormat.textIndent
textItalic	FStyleFormat.textItalic
textLeftMargin	FStyleFormat.textLeftMargin
textRightMargin	FStyleFormat.textRightMargin

Elemento di ActionScript	Vedere la voce
textSelected	FStyleFormat.textSelected
textSize	FStyleFormat.textSize
textUnderline	FStyleFormat.textUnderline
textWidth	TextField.textWidth
this	this
toggleHighQuality	toggleHighQuality
toLowerCase	String.toLowerCase
toString	Array.toString, Boolean.toString, Date.toString, Number.toString, Object.toString, XML.toString
_totalframes	MovieClip._totalframes
toUpperCase	String.toUpperCase
trace	trace
trackAsMenu	Button.trackAsMenu, MovieClip.trackAsMenu
true	true
testo	TextField.type
typeof	typeof
undefined	undefined
underline	TextFormat.underline
unescape	unescape
uninstall	CustomActions.uninstall
unloadMovie	unloadMovie, MovieClip.unloadMovie
unloadMovieNum	unloadMovieNum
unshift	Array.unshift
unwatch	Object.unwatch
UP	Key.UP
updateAfterEvent	updateAfterEvent
url	TextFormat.url
_url	MovieClip._url, TextField._url, Button._url
useHandCursor	Button.useHandCursor, MovieClip.useHandCursor
UTC	Date.UTC
valueOf	Boolean.valueOf, Number.valueOf, Object.valueOf
var	var
variabile	TextField.variable
versione	System.capabilities.version
_visible	MovieClip._visible, Button._visible, TextField._visible
void	void

Elemento di ActionScript	Vedere la voce
watch	Object.watch
while	while
width	Stage.width
_width	MovieClip._width, TextField._width, Button._width
with	with
wordwrap	TextField.wordWrap
_x	Button._x, MovieClip._x, TextField._x
XML	XML (oggetto)
xmlDecl	XML.xmlDecl
XMLSocket	XMLSocket (oggetto)
_xmouse	Button._xmouse, MovieClip._xmouse, TextField._xmouse
_xscale	Button._xscale, MovieClip._xscale, TextField._xscale
_y	Button._y, MovieClip._y, TextField._y
_ymouse	Button._ymouse, MovieClip._ymouse, TextField._ymouse
_yscale	Button._yscale, MovieClip._yscale, TextField._yscale

— (decremento)

Disponibilità

Flash Player 4.

Uso

`--espressione`

`espressione--`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); operatore unario di decremento prima e dopo l'operazione che sottrae 1 dal valore di *espressione*. L'operatore di decremento prima dell'operazione (`--espressione`) sottrae 1 dal valore di *espressione* e restituisce il risultato. L'operatore di decremento dopo l'operazione (`espressione--`) sottrae 1 dal valore di *espressione* e restituisce il valore iniziale di *espressione* (il valore prima della sottrazione).

Esempio

L'operatore di decremento prima dell'operazione cambia il valore di *x* in 2 ($x - 1 = 2$) e restituisce il risultato come *y*:

```
x = 3;  
y = --x;  
//y è uguale a 2
```

L'operatore di decremento dopo l'operazione cambia il valore di *x* in 2 ($x - 1 = 2$) e restituisce il valore originale di *x* come risultato *y*:

```
x = 3;  
y = x--  
//y è uguale a 3
```

++ (incremento)

Disponibilità

Flash Player 4.

Uso

`++espressione`

`espressione++`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); operatore unario di incremento prima e dopo l'operazione che aggiunge 1 al valore di *espressione*. L'*espressione* può essere una variabile, un elemento di una matrice o una proprietà di un oggetto. L'operatore di incremento prima dell'operazione (`++espressione`) aggiunge 1 al valore di *espressione* e restituisce il risultato. L'operatore di incremento dopo l'operazione (`espressione ++`) aggiunge 1 al valore di *espressione* e restituisce il valore iniziale di *espressione* (il valore prima dell'addizione).

L'operatore di incremento prima dell'operazione incrementa il valore di x a 2 ($x + 1 = 2$) e restituisce il risultato come y :

```
x = 1;
y = ++x
//y è uguale a 2
```

L'operatore di decremento dopo l'operazione incrementa il valore di x a 2 ($x + 1 = 2$) e restituisce il valore originale di x come risultato y :

```
x = 1;
y = x++;
//y è uguale a 1
```

Esempio

Nell'esempio seguente, ++ viene usato come operatore di incremento dopo l'operazione per eseguire cinque volte il ciclo `while`.

```
i = 0;
while(i++ < 5){
  trace("questa è l'esecuzione" + i);
}
```

Nell'esempio seguente, ++ viene usato come operatore di incremento prima dell'operazione:

```
var a = [];
var i = 0;
while (i < 10) {
  a.push(++i);
}
trace(a.join());
```

Questo script produce i seguenti risultati nella finestra Output:

1, 2, 3, 4, 5, 6, 7, 8, 9,10

Nell'esempio seguente, ++ viene usato come operatore dopo l'operazione:

```
var a = [];  
var i = 0;  
while (i < 10) {  
  a.push(i++);  
}  
trace(a.join());
```

Questo script produce i seguenti risultati nella finestra Output:

1, 2, 3, 4, 5, 6, 7, 8, 9

! (NOT logico)

Disponibilità

Flash Player 4.

Uso

!espressione

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (logico); inverte il valore booleano di una variabile o di un'espressione. Se *espressione* è una variabile con un valore assoluto o convertito *true*, il valore di *! espressione* è *false*. Se l'espressione *x && y* restituisce *false*, l'espressione *!(x && y)* restituisce *true*.

Le seguenti espressioni mostrano il risultato quando viene utilizzato l'operatore !:

! true restituisce *false*

! false restituisce *true*

Esempio

Nell'esempio seguente, la variabile *happy* è impostata su *false*. La condizione *if* valuta la condizione *!happy*. Se la condizione è *true*, l'azione *trace* invia una stringa alla finestra Output.

```
happy = false;  
if (!happy) {  
  trace("Non preoccuparti, sii felice");  
}
```

!=(diseguaglianza)

Disponibilità

Flash Player 5.

Uso

espressione1 != espressione2

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (diseguaglianza); verifica l'esatto opposto dell'operatore `==`. Se *espressione1* è uguale a *espressione2*, il risultato è `false`. Come nel caso dell'operatore `==`, l'*uguaglianza* dipende dal tipo di dati che vengono confrontati.

- Numeri, stringhe e valori booleani vengono confrontati come valore.
- Variabili, oggetti, matrici e funzioni vengono confrontati come riferimento.

Esempio

Nell'esempio seguente, viene mostrato il risultato dell'operatore `!=`:

```
5 != 8 restituisce true
```

```
5 != 5 restituisce false
```

Nell'esempio seguente, viene mostrato l'uso dell'operatore `!=` in un'istruzione `if`:

```
a = "Davide";  
b = "Sciocco"  
if (a != b){  
    trace("Davide non è uno sciocco");  
}
```

Vedere anche

`!=` (diseguaglianza rigorosa), `==` (uguaglianza), `===` (uguaglianza rigorosa)

!= (diseguaglianza rigorosa)

Disponibilità

Flash Player 6.

Uso

espressione1 `!=` *espressione2*

Descrizione

Operatore; verifica l'esatto opposto dell'operatore `===`. L'operatore di diseguaglianza rigorosa si comporta come l'operatore di diseguaglianza. Tuttavia, i tipi di dati non vengono convertiti. Se *espressione1* equivale a *espressione2* e i tipi di dati sono identici, il risultato è `false`. Come nel caso dell'operatore `===`, l'*uguaglianza* dipende dal tipo di dati confrontati.

- Numeri, stringhe e valori booleani vengono confrontati come valore.
- Variabili, oggetti, matrici e funzioni vengono confrontati come riferimento.

Esempio

Il codice riportato di seguito mostra il valore restituito dalle operazioni che usano operatori di uguaglianza, uguaglianza rigorosa e disuguaglianza rigorosa.

```
s1 = new String("4");
s2 = new String("4");
s3 = new String("Ciao");
n = new Number(4);
b = new Boolean(true);
```

```
s1 == s2; // true
s1 == s3; // false
s1 == n; // true
s1 == b; // false
```

```
s1 === s2; // true
s1 === s3; // false
s1 === n; // false
s1 === b; // false
```

```
s1 !== s2; // false
s1 !== s3; // true
s1 !== n; // true
s1 !== b; // true
```

Vedere anche

!=(disuguaglianza), == (uguaglianza), === (uguaglianza rigorosa)

% (modulo)

Disponibilità

Flash Player 4. Nei file Flash 4, l'operatore % viene espanso nel file SWF come $x - \text{int}(x/y) * y$ e il risultato potrebbe non essere ottenuto con la stessa velocità o precisione delle versioni più recenti di Flash Player.

Uso

espressione1 % *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); calcola il resto di *espressione1* diviso per *espressione2*. Se entrambi i parametri dell'*espressione* non sono numerici, l'operatore modulo tenta di convertirli in numeri. L'*espressione* può essere un numero o una stringa di conversione in un valore numerico.

Esempio

Di seguito è riportato un esempio numerico dell'uso dell'operatore del modulo (%):

```
trace (12 % 5);
// Restituisce 2
trace (4,3 % 2,1);
// Restituisce approssimativamente 0,1
```

%= (assegnazione modulo)

Disponibilità

Flash Player 4.

Uso

espressione1 %= *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione aritmetica composta); assegna a *espressione1* il valore di *espressione1* % *espressione2*. Ad esempio, le due espressioni riportate di seguito sono identiche:

```
x %= y  
x = x % y
```

Esempio

Nell'esempio seguente, alla variabile *x* viene assegnato il valore 4.

```
x = 14;  
y = 5;  
trace(x %= y);  
// Restituisce 4
```

Vedere anche

% (modulo)

& (AND bit a bit)

Disponibilità

Flash Player 5. In Flash 4, l'operatore & veniva usato per la concatenazione delle stringhe. In Flash 5, & è l'operatore AND bit a bit ed è necessario usare gli operatori `add` e `+` per concatenare le stringhe. I file Flash 4 che usano l'operatore & vengono aggiornati automaticamente per l'uso di `add` al momento dell'importazione nell'ambiente di creazione di Flash 5.

Uso

espressione1 & *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione1* e *espressione2* in numeri interi a 32 bit senza segno ed esegue un'operazione AND booleana su ciascun bit dei parametri dei numeri interi. Il risultato è un numero intero a 32 bit senza segno.

&& (cortocircuito AND)

Disponibilità

Flash Player 4.

Uso

espressione1 && *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (logico); esegue un'operazione booleana sui valori di una o entrambe le espressioni. Valuta *espressione1* (l'espressione a sinistra dell'operatore) e restituisce *false*, se l'espressione ha restituito *false*. Se *espressione1* restituisce *true*, viene valutata *espressione2* (ovvero l'espressione a sinistra dell'operatore). Se *espressione2* restituisce *true*, il risultato finale è *true*; in caso contrario, sarà *false*.

Esempio

Nell'esempio seguente, l'operatore && viene usato per eseguire una verifica e stabilire se un giocatore ha vinto una partita. Le variabili *turns* e *score* vengono aggiornate quando il giocatore inizia un turno oppure guadagna punti nel corso di una partita. Il seguente script consente di visualizzare il messaggio "Hai vinto!" nella finestra Output quando il giocatore totalizza 75 punti o un punteggio superiore in tre turni o meno.

```
turns=2;
score=77;
winner = (turns <= 3) && (score >= 75);
if (winner) {
    trace("Hai vinto!");
} else {
    trace("Riprova!");
}
```

&= (assegnazione AND bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 &= *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); assegna a *espressione1* il valore di *espressione1* & *espressione2*. Ad esempio, le due espressioni riportate di seguito sono identiche:

```
x &= y  
x = x & y
```

Esempio

Nell'esempio seguente, a *x* è assegnato il valore 9.

```
x = 15;  
y = 9;  
trace(x &= y);  
// Restituisce 9
```

Vedere anche

& (AND bit a bit)

() (parentesi tonde)

Disponibilità

Flash Player 4.

Uso

```
(espressione1, espressione2);  
funzione(parametro1, ..., parametroN);
```

Parametri

espressione1, *espressione2* Numeri, stringhe, variabili o testo.

funzione La funzione da eseguire sul contenuto racchiuso tra parentesi tonde.

parametro1...*parametroN* Una serie di parametri da eseguire prima che i risultati vengano inviati come parametri della funzione posta all'esterno delle parentesi tonde.

Valori restituiti

Nessuno.

Descrizione

Operatore; esegue un'operazione di raggruppamento su uno o più parametri oppure ne delimita uno (o più di uno) e li invia come parametri a una funzione posta all'esterno delle parentesi tonde.

Uso 1: controlla l'ordine di esecuzione degli operatori nell'espressione. Le parentesi consentono di ignorare il normale ordine di precedenza; in questo modo le espressioni racchiuse tra parentesi vengono valutate per prime. Nel caso di parentesi nidificate, viene valutato prima il contenuto delle parentesi più interne, quindi quello delle parentesi più esterne.

Uso 2: delimita uno o più parametri e li passa alla funzione posta all'esterno delle parentesi.

Esempio

Uso 1: le seguenti istruzioni illustrano l'uso delle parentesi tonde per gestire l'ordine in cui le espressioni vengono eseguite. Il valore di ciascuna espressione viene visualizzato al di sotto di ciascuna linea, come mostrato di seguito:

```
trace((2 + 3) * (4 + 5));  
// Visualizza 45  
  
trace(2 + (3 * (4 + 5)));  
// Visualizza 29  
  
trace(2 + (3 * 4) + 5);  
// Visualizza 19
```

Uso 2: l'esempio seguente illustra l'uso delle parentesi tonde con le funzioni:

```
getDate();  
  
invoice(item, amount);  
  
function traceParameter(param){  
    trace  
}  
traceParameter(2*2);
```

Vedere anche

with

- (meno)

Disponibilità

Flash Player 4.

Uso

(Negazione) *–*espressione

(Sottrazione) *espressione1 - espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); usato per la negazione o la sottrazione.

Uso 1: se usato per la negazione, inverte il segno dell'*espressione* numerica.

Uso 2: se usato per la sottrazione, esegue una sottrazione su due espressioni numeriche, sottraendo *espressione2* da *espressione1*. Se le due espressioni sono costituite da numeri interi, la differenza sarà un numero intero. Se una o entrambe le espressioni sono costituite da numeri in virgola mobile, la differenza sarà un numero in virgola mobile.

Esempio

Uso 1: l'istruzione seguente inverte il segno dell'espressione $2 + 3$.

$-(2 + 3)$

Il risultato è -5.

Uso 2: l'istruzione seguente sottrae il numero intero 2 dal numero intero 5.

$5 - 2$

Il risultato è 3 (ovvero un numero intero).

Uso 2: l'istruzione seguente sottrae il numero in virgola mobile 1,5 dal numero in virgola mobile 3,25.

$3,25 - 1,5$

Il risultato è 1,75 (ovvero un numero in virgola mobile).

*** (moltiplicazione)**

Disponibilità

Flash Player 4.

Uso

espressione1 * *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); moltiplica due espressioni numeriche. Se le due espressioni sono costituite da numeri interi, il prodotto sarà un numero intero. Se una o entrambe le espressioni sono costituite da numeri in virgola mobile, il prodotto sarà un numero in virgola mobile.

Esempio

L'istruzione seguente moltiplica i numeri interi 2 e 3:

$2 * 3$

Il risultato è 6 (ovvero un numero intero).

Esempio

Questa istruzione moltiplica tra loro i numeri in virgola mobile 2 e 3,1416:

$2 * 3,1416$

Il risultato è 6,2832 (ovvero un numero in virgola mobile).

***= (assegnazione moltiplicazione)**

Disponibilità

Flash Player 4.

Uso

espressione1 *= *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta aritmetica); assegna a *espressione1* il valore di *espressione1** *espressione2*. Ad esempio, le due espressioni riportate di seguito sono identiche:

```
x *= y
x = x * y
```

Esempio

Nell'esempio seguente alla variabile *x* è assegnato il valore 50.

```
x = 5;
y = 10;
trace(x *= y);
// Restituisce 50
```

Esempio

Nella seconda e terza riga dell'esempio seguente, vengono calcolate le espressioni a destra del segno di uguale e vengono assegnati i risultati a *x* e *y*.

```
i = 5;
x = 4 - 6;
y = i + 2;
trace(x *= y);
// Restituisce -14
```

Vedere anche

* (moltiplicazione)

, (virgola)

Disponibilità

Flash Player 4.

Uso

espressione1, *espressione2*

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Operatore; valuta *espressione1*, poi *espressione2* e restituisce il valore di *espressione2*. Questo operatore si usa principalmente con l'istruzione del ciclo `for`.

Esempio

Nel codice seguente viene usato l'operatore virgola:

```
var a=1, b=2, c=3;
```

Ciò equivale al codice riportato di seguito:

```
var a=1;  
var a=2;  
var c=3;
```

. (punto)

Disponibilità

Flash Player 4.

Uso

oggetto.proprietà_o_metodo

nomeistanza.variabile

nomeistanza.istanzasecondaria.variabile

Parametri

oggetto L'istanza di un oggetto. Può trattarsi di un oggetto incorporato di ActionScript o di un oggetto personalizzato. Questo parametro si trova sempre a sinistra dell'operatore punto (.).

proprietà_o_metodo Il nome di una proprietà o di un metodo associato a un oggetto. I metodi e le proprietà validi per gli oggetti incorporati sono elencati nelle tabelle di riepilogo dei metodi e delle proprietà relative all'oggetto. Questo parametro si trova sempre a destra dell'operatore punto (.).

nomeistanza Il nome dell'istanza di un clip filmato.

istanzasecondaria L'istanza di un clip filmato secondario o nidificato in un altro clip filmato.

variabile Una variabile sulla linea temporale del nome dell'istanza del clip filmato a sinistra dell'operatore punto (.).

Valori restituiti

Nessuno.

Descrizione

Operatore; usato per accedere ai clip filmato nidificati (secondari), alle variabili o alle proprietà all'interno delle gerarchie dei clip filmato. L'operatore punto consente anche di verificare o impostare le proprietà di un oggetto, eseguirne un metodo oppure creare una struttura di dati.

Esempio

L'istruzione riportata di seguito identifica il valore corrente della variabile `hairColor` nel clip filmato `person`.

```
person.hairColor
```

Ciò equivale alla seguente sintassi di Flash 4:

```
/person:hairColor
```

Esempio

Il codice riportato di seguito illustra come l'operatore punto possa essere usato per creare la struttura di una matrice.

```
account.name = "Sergio Doni";  
account.address = "Via Poli, 123";  
account.city = "Qualsiasi città";  
account.state = "I";  
account.zip = "12345";
```

Vedere anche

[] (accesso matrice)

?: (condizionale)

Disponibilità

Flash Player 4.

Uso

espressione1 ? *espressione2* : *espressione3*

Parametri

espressione1 Un'espressione che restituisce un valore booleano, solitamente un'espressione di confronto, ad esempio $x < 5$.

espressione2, *espressione3* Valori di qualsiasi tipo.

Valori restituiti

Nessuno.

Descrizione

Operatore; Flash riceve l'istruzione di valutare *espressione1*. Se il valore di *espressione1* è true, viene restituito il valore di *espressione2* o, in caso contrario, il valore di *espressione3*.

Esempio

L'istruzione riportata di seguito assegna il valore della variabile x a z, poiché *espressione1* corrisponde a true:

```
x = 5;  
y = 10;  
z = (x < 6) ? x : y;  
trace(z);  
// Restituisce 5
```

/ (divisione)

Disponibilità

Flash Player 4.

Uso

espressione1 / *espressione2*

Parametri

espressione Un numero o una variabile che restituisce un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (aritmetico); divide *espressione1* per *espressione2*. Il risultato della divisione è un numero in virgola mobile a doppia precisione.

Esempio

La seguente istruzione divide il numero in virgola mobile 22,0 per 7,0, quindi visualizza il risultato nella finestra Output:

```
trace(22,0 / 7,0);
```

Il risultato è 3,1429 (ovvero un numero in virgola mobile).

// (delimitatore di commento)

Disponibilità

Flash 1.

Uso

```
// commento
```

Parametri

commento Qualsiasi carattere.

Valori restituiti

Nessuno.

Descrizione

Commento; identifica l'inizio di un commento nello script. Qualsiasi carattere visualizzato tra il delimitatore di commento // e il carattere di fine riga viene interpretato come commento e ignorato dall'interprete di ActionScript.

Esempio

In questo script, le barre inclinate dei delimitatori di commento consentono di identificare la prima, la terza, la quinta e la settima come righe di commento.

```
// Registra la posizione X del clip filmato ball  
ballX = ball._x;  
// Registra la posizione Y del clip filmato ball  
ballY = ball._y;  
// Registra la posizione X del clip filmato bat  
batX = bat._x;  
// Registra la posizione Y del clip filmato bat  
batY = bat._y;
```

Vedere anche

```
/* (delimitatore di commento)
```

/* (delimitatore di commento)

Disponibilità

Flash Player 5.

Uso

```
/* commento */  
  
/*  
commento  
commento  
*/
```

Parametri

commento Qualsiasi carattere.

Valori restituiti

Nessuno.

Descrizione

Commento; indica una o più righe di commento nello script. Il testo racchiuso tra il tag di apertura `/*` e il tag di chiusura del commento `*/` viene interpretato come commento e ignorato dall'interprete di ActionScript. Per identificare i commenti costituiti da una sola riga, usare il primo tipo di sintassi. Per identificare i commenti su più righe consecutive, usare il secondo tipo di sintassi. L'esclusione del tag di chiusura `*/` con il delimitatore di commento restituisce un messaggio di errore.

Esempio

In questo script i delimitatori di commento sono all'inizio dello script.

```
/* Registra le posizioni X e Y del  
clip filmato ball e bat  
*/
```

```
ballX = ball._x;  
ballY = ball._y;  
batX = bat._x;  
batY = bat._y;
```

Vedere anche

`//` (delimitatore di commento)

/= (assegnazione divisione)

Disponibilità

Flash Player 4.

Uso

```
espressione1 /= espressione2
```

Parametri

espressione1, *espressione2* Un numero o una variabile che restituisce un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta aritmetica); assegna a *espressione1* il valore di *espressione1 / espressione2*. Ad esempio, le due istruzioni riportate di seguito sono identiche:

```
x /= y  
x = x / y
```

Esempio

L'esempio seguente illustra l'uso dell'operatore `/=` con variabili e numeri.

```
x = 10;  
y = 2;  
x /= y  
// x adesso contiene il valore 5
```

[] (accesso matrice)

Disponibilità

Flash Player 4.

Uso

```
myArray = ["a0", a1, ...aN];  
myMultiDimensionalArray = [[ "a0", ...aN], ...[ "a0", ...aN]]  
myArray[E] = value  
myMultiDimensionalArray[E][E] = value  
oggetto["valore"];
```

Parametri

myArray Il nome di una matrice.

a0, a1, ...aN Elementi in una matrice.

myMultiDimensionalArray Il nome di una matrice multidimensionale simulata.

E Il numero (o indice) di un elemento in una matrice.

oggetto Il nome di un oggetto.

valore Una stringa o un'espressione che restituisce una stringa che denomina una proprietà dell'oggetto.

Valori restituiti

Nessuno.

Descrizione

Operatore; inizializza una matrice nuova o multidimensionale con gli elementi specificati (*a0* e così via) o accede agli elementi in una matrice. L'operatore di accesso alla matrice consente di impostare e recuperare dinamicamente nomi di istanze, variabili e oggetti. Consente inoltre di accedere alle proprietà degli oggetti.

Uso 1: una matrice è un oggetto le cui proprietà sono definite *elementi*, identificate da un numero chiamato *indice*. Quando si crea una matrice, si delimitano gli elementi con l'operatore di accesso alla matrice (o *parentesi*). Una matrice può contenere elementi di diverso tipo. Ad esempio, la matrice riportata di seguito denominata `employee`, è composta da tre elementi: il primo è un numero, gli altri sono stringhe (tra virgolette).

```
employee = [15, "Barbara", "Enrico"];
```

Uso 2: per simulare matrici multidimensionali, è possibile nidificare le parentesi. Il codice riportato di seguito crea una matrice chiamata `ticTacToe` con tre elementi; ciascun elemento è una matrice con tre elementi.

```
ticTacToe = [[1,2,3],[4,5,6],[7,8,9]];
```

```
// Seleziona Debug > Elenco variabili nella modalità di prova filmato  
// per visualizzare un elenco degli elementi matrice
```

Uso 3: pone l'indice di ciascun elemento tra parentesi per consentire l'accesso diretto; è ora possibile aggiungere un nuovo elemento a una matrice, modificare o richiamare il valore di un elemento esistente. Al primo elemento della matrice corrisponde sempre 0:

```
myArray[0] = 15;  
myArray[1] = "Ciao!";  
myArray[2] = true;
```

È inoltre possibile usare le parentesi per aggiungere un quarto elemento, come nell'esempio seguente:

```
myArray[3] = "Giorgio";
```

Uso 4: per accedere a un elemento di una matrice multidimensionale, è possibile usare le parentesi. Il primo gruppo di parentesi identifica l'elemento della matrice originale, mentre il secondo identifica l'elemento della matrice nidificata. La riga di codice successiva invia il numero 6 nella finestra Output.

```
ticTacToe = [[1,2,3],[4,5,6],[7,8,9]];  
trace(ticTacToe[1][2]);
```

```
// Restituisce 6
```

Uso 5: è possibile usare l'operatore di accesso alla matrice invece della funzione `eval` per impostare e richiamare in modo dinamico i valori relativi ai nomi dei clip filmato o a una delle proprietà di un oggetto:

```
name["mc" + i] = "left_corner";
```

Esempio

Uso 1: i due esempi di codice riportati di seguito mostrano due sistemi di creazione di un nuovo oggetto Array vuoto. Nella prima riga si usano le parentesi.

```
myArray =[];  
myArray =[];
```

Uso 1 e 2: nell'esempio riportato di seguito, viene creata una matrice chiamata `employee` e si usa un'azione `trace` per inviare gli elementi alla finestra Output. Nella quarta riga, viene modificato un elemento della matrice e la quinta riga invia la nuova matrice modificata alla finestra Output:

```
employee=["Barbara", "Giorgio", "Maria"];  
trace(employee);  
// Barbara, Giorgio, Maria  
employee[2]="Ada"  
trace(employee);  
// Barbara, Giorgio, Ada
```

Uso 3: nell'esempio riportato di seguito, viene valutata l'espressione contenuta all'interno delle parentesi ("piece" + i) e si usa il risultato come nome della variabile da richiamare dal clip filmato mc. In questo esempio, la variabile i deve esistere nella stessa linea temporale del pulsante. Se, ad esempio, la variabile i corrisponde a 5, il valore della variabile piece5 nel clip filmato mc verrà visualizzato nella finestra Output:

```
on(release){
    x = mc["piece"+i];
    trace(x);
}
```

Uso 3: nel codice riportato di seguito, viene valutata l'espressione posta tra parentesi e si usa il risultato come nome della variabile da richiamare dal nome del clip filmato.

```
group["A" + i];
```

Se si ha una buona conoscenza della sintassi della barra inclinata di ActionScript per Flash 4, la funzione eval consente di ottenere lo stesso risultato:

```
eval("A" & i);
```

Uso 3: è inoltre possibile usare l'operatore di accesso alla matrice sul lato sinistro di un'istruzione di assegnazione per impostare in modo dinamico nomi di istanze, variabili e oggetti.

```
name[index] = "Sergio";
```

Vedere anche

Array (oggetto), Object (oggetto), eval

^(XOR bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 ^ *espressione2*

Parametri

espressione1, *espressione2* Un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione1* ed *espressione2* in numeri interi a 32 bit senza segno, quindi restituisce 1 nella posizione di ciascun bit, dove il bit corrispondente in *espressione1* o *espressione2* (non entrambi) equivale a 1.

Esempio

Nell'esempio seguente, l'operatore XOR bit a bit viene usato con i decimali 15 e 9 e il risultato è assegnato alla variabile x.

```
// 15 notazione decimale = 1111 notazione binaria
// 9 notazione decimale = 1001 notazione binaria
x = 15 ^ 9
trace(x)
// 1111 ^ 1001 = 0110
// Restituisce 6 notazione decimale (= 0110 notazione binaria)
```

^= (assegnazione XOR bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 ^= *espressione2*

Parametri

espressione1, *espressione2* Numeri interi e variabili.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); assegna a *espressione1* il valore di *espressione1* ^ *espressione2*. Ad esempio, le due istruzioni riportate di seguito sono identiche:

```
x ^= y  
x = x ^ y
```

Esempio

Il seguente è un esempio di operazione ^=:

```
// 15 notazione decimale = 1111 notazione binaria  
x = 15;  
// 9 notazione decimale = 1001 notazione binaria  
y = 9;  
trace(x ^= y);  
// Restituisce 6 notazione decimale (= 0110 notazione binaria)
```

Vedere anche

^(XOR bit a bit)

{ } (operatore di inizializzazione degli oggetti)

Disponibilità

Flash Player 5.

Uso

oggetto = {*nome1*: *valore1*, *nome2*: *valore2*, ... *nomeN*: *valoreN*};

Parametri

oggetto L'oggetto da creare.

nome1, 2, ... N I nomi delle proprietà.

valore1, 2, ... N I valori corrispondenti per ciascuna proprietà *nome*.

Valori restituiti

Nessuno.

Descrizione

Operatore; crea un nuovo oggetto e lo inizializza con la coppia di proprietà *nome* e *valore* specificata. L'uso di questo operatore equivale all'adozione della sintassi `new Object` e alla creazione delle coppie di proprietà tramite l'operatore di assegnazione. Il prototipo del nuovo oggetto creato è l'oggetto denominato genericamente *Object*.

Esempio

La prima riga del codice riportato di seguito crea un oggetto vuoto tramite l'operatore di inizializzazione degli oggetti. La seconda riga crea un nuovo oggetto tramite una funzione di creazione.

```
object = {};  
object = new Object();
```

Nell'esempio seguente, viene creato un oggetto `account` e vengono inizializzate le proprietà `name`, `address`, `city`, `state`, `zip` e `balance` con i relativi valori.

```
account = { name: "Elisa Grani",  
  address: "Via Roma, 123",  
  city: "Soncino",  
  state: "Italia",  
  zip: "12345",  
  balance: "1000" };
```

Nell'esempio seguente viene indicato come nidificare gli operatori di inizializzazione di matrici e oggetti l'uno nell'altro.

```
person = { name: "Gina Vecchi",  
  children: [ "Rudy", "Aurelia", "Carla" ] };
```

Nell'esempio seguente vengono usate le informazioni dell'esempio precedente per produrre lo stesso risultato mediante le funzioni di costruzione.

```
person = new Person();  
person.name = 'Gina Vecchi';  
person.children = new Array();  
person.children[0] = 'Rudy';  
person.children[1] = 'Aurelia';  
person.children[2] = 'Carla';
```

Vedere anche

[] (accesso matrice), new, Object (oggetto)

| (OR bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 | *espressione2*

Parametri

espressione1, *espressione2* Un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione1* ed *espressione2* in numeri interi a 32 bit senza segno e restituisce 1 nella posizione di ciascun bit dove il bit corrispondente di *espressione1* o *espressione2* equivale a 1.

Esempio

Il seguente è un esempio di operazione OR bit a bit.

```
// 15 notazione decimale = 1111 notazione binaria
x = 15;
// 9 notazione decimale = 1001 notazione binaria
y = 9;
trace(x | y);
// 1111 | 0011 = 1111
// Restituisce 15 notazione decimale (= 1111 notazione binaria)
```

|| (OR logico)

Disponibilità

Flash Player 4.

Uso

```
espressione1 || espressione2
```

Parametri

espressione1, *espressione2* Un valore booleano o un'espressione che restituisce un valore booleano

Valori restituiti

Nessuno.

Descrizione

Operatore (logico); valuta *espressione1* e *espressione2*. Il risultato è *true* se una o entrambe le espressioni restituiscono *true*; mentre il risultato è *false* solo se entrambe le espressioni restituiscono *false*. È possibile usare l'operatore OR logico con qualsiasi numero di operandi; se qualsiasi operando restituisce *true*, il risultato è *true*.

Con espressioni non booleane, l'operatore OR logico indica a Flash di valutare l'espressione posta a sinistra. Se questa può essere convertita in *true* il risultato è *true*. In caso contrario, verrà valutata l'espressione posta a destra e il risultato sarà il valore di tale espressione.

Esempio

L'esempio seguente usa l'operatore `||` in un'istruzione `if`. Poiché la seconda espressione restituisce *true*, il risultato finale è *true*:

```
x = 10
y = 250
start = false
if(x > 25 || y > 200 || start){
    trace('test OR logico superato');
}
```

Esempio

Questo esempio mostra come un'espressione non booleana possa produrre un risultato inatteso. Se l'espressione a sinistra diventa `true`, il risultato viene restituito senza conversione dell'espressione a destra.

```
function fx1(){
    trace ("richiamato fx1");
    returns true;
}
function fx2(){
    trace ("richiamato fx2");
    return true;
}
if (fx1() || fx2()){
    trace ("istruzione IF immessa");
}
// I seguenti valori vengono visualizzati nella finestra Output:
// Richiamato fx1
// Istruzione IF immessa
```

|= (assegnazione OR bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 |= *espressione2*

Parametri

espressione1, *espressione2* Un numero o una variabile.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); assegna a *espressione1* il valore di *espressione1* | *espressione2*. Ad esempio, le due istruzioni riportate di seguito sono identiche:

```
x |= y;
x = x | y;
```

Esempio

Nell'esempio seguente, si usa l'operatore |=:

```
// 15 notazione decimale = 1111 notazione binaria
x = 15;
// 9 notazione decimale = 1001 notazione binaria
y = 9;
trace(x | y);
// 1111 |= 1001
// Restituisce 15 notazione decimale (= 1111 notazione binaria)
```

Vedere anche

| (OR bit a bit)

~ (NOT bit a bit)

Disponibilità

Flash Player 5.

Uso

~ espressione

Parametri

espressione Un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione* in un intero a 32 bit senza segno, quindi inverte i bit. Un'operazione NOT bit a bit cambia il segno di un numero e sottrae 1.

Esempio

Nell'esempio seguente, viene mostrata un'operazione NOT bit a bit eseguita su una variabile:

```
a = 0;
trace ("quando a = 0, ~a = "+~a);
// Quando a = 0, ~a = -1
a = 1;
trace ("quando a = 1, ~a = "+~a);
// Quando a = 0, ~a = -2
// allora, ~0=-1 e ~1=-2
```

+ (addizione)

Disponibilità

Flash Player 4; Flash Player 5. In Flash 5, + è un operatore numerico o un operatore di concatenazione delle stringhe, a seconda del tipo di dati del parametro. In Flash 4 + è solo un operatore numerico. I file di Flash 4 importati in ambiente di creazione di Flash 5 vengono sottoposti a un processo di conversione per mantenere l'integrità del tipo di dati. L'esempio seguente mostra la conversione di un file Flash 4 contenente un confronto qualitativo numerico.

File Flash 4:

```
x + y
```

File Flash 5 convertito:

```
Numero(x) + Numero(y)
```

Uso

espressione1 + espressione2

Parametri

espressione1, espressione2 Un numero o una stringa.

Valori restituiti

Nessuno.

Descrizione

Operatore; aggiunge espressioni numeriche o concatena (combina) le stringhe. Se un'espressione è una stringa, tutte le altre espressioni vengono convertite in stringhe e concatenate.

Se entrambe le espressioni sono numeri interi, la loro somma sarà un numero intero; se una o entrambe le espressioni sono numeri in virgola mobile, la loro somma sarà un numero in virgola mobile.

Esempio

Nell'esempio seguente, due stringhe vengono concatenate e il risultato viene visualizzato nella finestra Output:

```
name = "Cola";
instrument = "Percussioni";
trace (name + " suona " + instrument);
```

Esempio

Le variabili associate ai campi dinamici e testo di input hanno il tipo di dati stringa. Nell'esempio seguente, la variabile `deposit` è un campo di testo di input nello stage. Dopo che è stato immesso l'importo depositato, lo script aggiunge `deposit` a `oldBalance`. Tuttavia, poiché `deposit` è un tipo di dati stringa, lo script concatena (ovvero combina in un'unica stringa) i valori della variabile invece di sommarli.

```
oldBalance = 1345.23;
currentBalance = deposit + oldBalance;
trace (currentBalance);
```

Se, ad esempio, si immette 475 nel campo di testo del deposito, l'azione `trace` invia il valore 4751345,23 alla finestra Output.

Per apportare una correzione, convertire la stringa in un numero tramite la funzione `Number`, come indicato di seguito:

```
currentBalance = Number(deposit) + oldBalance;
```

Esempio

La seguente istruzione somma i numeri interi 2 e 3, quindi visualizza il numero intero 5 nella finestra Output:

```
trace (2 + 3);
```

La seguente istruzione aggiunge i numeri in virgola mobile 2,5 e 3,25, quindi visualizza il numero in virgola mobile risultante (5,75) nella finestra Output:

```
trace (2,5 + 3,25);
```

Vedere anche

`add`

+= (assegnazione addizione)

Disponibilità

Flash Player 4.

Uso

espressione1 += *espressione2*

Parametri

espressione1, *espressione2* Un numero o una stringa.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione aritmetica composta); assegna a *espressione1* il valore di *espressione1* + *espressione2*. Ad esempio, le due istruzioni riportate di seguito hanno un risultato identico:

```
x += y;  
x = x + y;
```

Questo operatore consente anche di concatenare le stringhe. Tutte le regole valide per l'operatore di addizione (+) sono applicabili all'operatore di assegnazione addizione (+=).

Esempio

Di seguito, viene riportato un esempio di uso numerico dell'operatore +=.

```
x = 5;  
y = 10;  
x += y;  
trace(x);  
//x restituisce 15
```

Nell'esempio seguente, l'operatore += viene usato con un'espressione stringa e consente di visualizzare il messaggio "Il mio nome è Alberto" nella finestra Output.

```
x = "Il mio nome è"  
x += "Alberto"  
trace (x)
```

Vedere anche

+ (addizione)

< (minore di)

Disponibilità

Flash Player 4; Flash Player 5. In Flash 5, < (minore di) è un operatore di confronto in grado di gestire diversi tipi di dati. In Flash 4, < è un operatore numerico. I file di Flash 4 importati in ambiente di creazione di Flash 5 vengono sottoposti a un processo di conversione per mantenere l'integrità del tipo di dati. L'esempio seguente illustra la conversione di un file Flash 4 contenente un'operazione numerica che richiede un confronto del tipo.

File Flash 4:

```
x < y
```

File Flash 5 convertito:

```
Numero(x) < Numero(y)
```

Uso

```
espressione1 < espressione2
```

Parametri

espressione1, *espressione2* Un numero o una stringa.

Descrizione

Operatore (confronto); consente di confrontare due espressioni e di determinare se il valore di *espressione1* è inferiore al valore di *espressione2*; in questo caso, l'operatore restituisce `true`. Se *espressione1* è maggiore o uguale a *espressione2*, l'operatore restituisce `false`. Le espressioni stringa vengono valutate in ordine alfabetico, tutte le lettere maiuscole precedono le lettere minuscole.

Esempio

Nell'esempio seguente, vengono mostrati due casi in cui i valori restituiti sono `true` e `false` per confronti numerici e di stringhe:

```
3 < 10;  
// true  
  
10 < 3;  
// false  
  
"Paolo" < "Dino";  
// true  
  
"Paolo" < "Dino";  
// false  
  
"11" < "3";  
//true  
  
"11" < 3;  
// Confronto numerico  
// false  
  
"C" < "abc";  
// false  
  
a = 1;  
// true
```

<< (spostamento a sinistra bit a bit)

Disponibilità

Flash Player 5.

Uso

```
espressione1 << espressione2
```

Parametri

espressione1 Un numero o un'espressione da spostare a sinistra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione1* ed *espressione2* in interi a 32 bit, quindi sposta a sinistra tutti i bit contenuti in *espressione1* di un numero di posizioni corrispondente al numero intero risultante dalla conversione di *espressione2*. Se al termine dell'operazione rimangono delle posizioni di bit vuote, in esse viene inserito uno 0. Quando un valore viene spostato a sinistra di una posizione, lo spostamento equivale a una moltiplicazione per 2.

Esempio

Nell'esempio seguente, il numero intero 1 viene spostato a sinistra di dieci bit.

```
x = 1 << 10
```

Il risultato di questa operazione è $x = 1024$. Ciò è dovuto al fatto che il numero decimale 1 equivale al numero binario 1, il numero binario 1 spostato a sinistra di 10 bit equivale al numero binario 1000000000 e quest'ultimo equivale al numero decimale 1024.

Nell'esempio seguente, il numero intero 7 viene spostato a sinistra di otto bit.

```
x = 7 << 8
```

Il risultato di questa operazione è $x = 1792$. Ciò è dovuto al fatto che il numero decimale 7 equivale al numero binario 111, il numero binario 111 spostato a sinistra di 8 bit equivale al numero binario 1110000000 e quest'ultimo equivale al numero decimale 1792.

Vedere anche

>>= (spostamento a destra bit a bit e assegnazione), >> (spostamento a destra bit a bit), <<= (spostamento a sinistra bit a bit e assegnazione)

<<= (spostamento a sinistra bit a bit e assegnazione)

Disponibilità

Flash Player 5.

Uso

```
espressione1 <<= espressione2
```

Parametri

espressione1 Un numero o un'espressione da spostare a sinistra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); esegue un'operazione di spostamento a sinistra bit a bit e memorizza il contenuto come risultato in *espressione1*. Le due espressioni seguenti si equivalgono:

```
A <<= B  
A = (A << B)
```

Vedere anche

<< (spostamento a sinistra bit a bit), >>= (spostamento a destra bit a bit e assegnazione), >> (spostamento a destra bit a bit)

<= (minore o uguale a)

Disponibilità

Flash Player 4.

File Flash 4:

$x \leq y$

File Flash 5 convertito:

`Number(x) <= Number(y)`

Uso

espressione1 <= *espressione2*

Parametri

espressione1, *espressione2* Un numero o una stringa.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); consente di confrontare due espressioni e di determinare se il valore di *espressione1* è inferiore al valore di *espressione2*. In questo caso, l'operatore restituisce `true`. Se *espressione1* è maggiore di *espressione2*, l'operatore restituisce `false`. Le espressioni stringa vengono valutate attraverso l'ordine alfabetico, tutte le lettere maiuscole precedono le lettere minuscole.

In Flash 5, l'operatore <= (minore di o uguale a) è un operatore di confronto in grado di gestire diversi tipi di dati. In Flash 4, <= è un operatore numerico. I file di Flash 4 importati in ambiente di creazione di Flash 5 vengono sottoposti a un processo di conversione per mantenere l'integrità del tipo di dati. L'esempio seguente illustra la conversione di un file Flash 4 contenente un'operazione numerica che richiede un confronto del tipo.

Esempio

Nell'esempio seguente vengono mostrati due casi in cui i valori restituiti sono `true` e `false` per confronti numerici e di stringhe:

```
5 <= 10;
// true

2 <= 2;
// true

10 <= 3;
// false

"Paolo" <= "Dino";
// true

"Dino" <= "Paolo";
// false

"11" <= "3";
//true

"11" <= 3;
// Confronto numerico
// false

"C" <= "abc";
// false

"A" <= "a";
// true
```

⟷ (diseguaglianza)

Disponibilità

Flash 2.

Uso

espressione1 <> *espressione2*

Parametri

espressione1, *espressione2* Un numero, una stringa, un valore booleano, una variabile, un oggetto, una matrice o una funzione.

Valori restituiti

Nessuno.

Descrizione

Operatore (diseguaglianza); verifica l'esatto opposto dell'operatore `==`. Se *espressione1* è uguale a *espressione2*, il risultato è `false`. Come nel caso dell'operatore `==`, l'*uguaglianza* dipende dal tipo di dati sottoposti a confronto.

- Numeri, stringhe e valori booleani vengono confrontati come valore.
- Variabili, oggetti, matrici e funzioni vengono confrontati come riferimento.

Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore `!=`.

Vedere anche

!=(diseguaglianza)

= (assegnazione)**Disponibilità**

Flash Player 4.

File Flash 4:

`x = y`

File Flash 5 convertito:

`Numero(x) == Numero(y)`

Uso

espressione1 = *espressione2*

Parametri

espressione1 Una variabile, un elemento di una matrice oppure una proprietà di un oggetto.

espressione2 Un valore di qualsiasi tipo.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione); assegna il tipo di valore di *espressione2* (l'argomento posto a destra) alla variabile, all'elemento di matrice o alla proprietà di *espressione1*.

In Flash 5, = è un operatore di assegnazione, mentre l'operatore == viene usato per valutare l'uguaglianza. In Flash 4, = è un operatore di uguaglianza numerica. I file di Flash 4 importati in ambiente di creazione di Flash 5 vengono sottoposti a un processo di conversione per mantenere l'integrità del tipo di dati.

Esempio

Nell'esempio seguente, si usa l'operatore di assegnazione per assegnare il tipo di dati numero alla variabile x.

`x = 5`

L'esempio seguente usa l'operatore di assegnazione per assegnare il tipo di dati stringa alla variabile x.

`x = "Ciao!"`

Vedere anche

== (uguaglianza)

-= (assegnazione sottrazione)**Disponibilità**

Flash Player 4.

Uso

espressione1 = *espressione2*

Parametri

espressione1, espressione2 Un numero o un'espressione che restituisce un numero.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione aritmetica composta); assegna a *espressione1* il valore di *espressione1* - *espressione2*. Ad esempio, le due istruzioni riportate di seguito sono identiche:

```
x -= y;  
x = x - y;
```

Le espressioni di tipo stringa devono essere convertite in numeri, in caso contrario viene restituita la variabile NaN.

Esempio

Nell'esempio seguente, l'operatore -= viene usato per sottrarre 10 da 5 e assegnare il risultato alla variabile x.

```
x = 5;  
y = 10;  
x -= y;  
trace(x);  
//Restituisce -5
```

Esempio

L'esempio seguente mostra la conversione delle stringhe in numeri.

```
x = "5";  
y = "10";  
x -= y;  
trace(x);  
// Restituisce -5
```

== (uguaglianza)

Disponibilità

Flash Player 5.

Uso

```
espressione1 == espressione2
```

Parametri

espressione1, espressione2 Un numero, una stringa, un valore booleano, una variabile, un oggetto, una matrice o una funzione.

Valori restituiti

Nessuno.

Descrizione

Operatore (uguaglianza); verifica l'uguaglianza tra due espressioni. Il risultato è true se le espressioni sono uguali.

L'*uguaglianza* dipende dal tipo di dati del parametro:

- Numeri e valori booleani vengono confrontati in base al valore e considerati uguali se il loro valore è identico.

- Le espressioni di stringhe si equivalgono se hanno lo stesso numero di caratteri e i caratteri sono identici.
- Variabili, oggetti, matrici e funzioni vengono confrontati come riferimento. Due variabili sono uguali se fanno riferimento allo stesso oggetto, matrice o funzione. Due matrici distinte non vengono mai considerate uguali, anche se sono costituite dallo stesso numero di elementi.

Esempio

L'esempio seguente usa l'operatore `==` in un'istruzione `if`:

```
a = "Dario" , b = "Dario";
if (a == b){
    trace("Dario è Dario");
}
```

Esempio

Gli esempi riportati di seguito mostrano i risultati delle operazioni di confronto dei tipi misti.

```
x = "5"; y = "5";
trace(x == y);
// true
```

```
x = "5"; y = "66";
trace(x ==y);
// false
```

```
x = "Mauro"; y = "Carlo";
trace(x == y);
//false
```

Vedere anche

`!=(disuguaglianza)`, `=== (uguaglianza rigorosa)`, `!== (disuguaglianza rigorosa)`

=== (uguaglianza rigorosa)

Disponibilità

Flash Player 6.

Uso

espressione1 === *espressione2*

Descrizione

Operatore; verifica l'uguaglianza di due espressioni; l'operatore di uguaglianza rigorosa si comporta come l'operatore di uguaglianza ma i tipi di dati non vengono convertiti. Il risultato è `true` se entrambe le espressioni, inclusi i tipi di dati corrispondenti, sono uguali.

L'*uguaglianza* dipende dal tipo di dati del parametro:

- Numeri e valori booleani vengono confrontati in base al valore e considerati uguali se il loro valore è identico.
- Le espressioni di stringhe si equivalgono se hanno lo stesso numero di caratteri e i caratteri sono identici.
- Variabili, oggetti, matrici e funzioni vengono confrontati come riferimento. Due variabili sono uguali se fanno riferimento allo stesso oggetto, matrice o funzione. Due matrici distinte non vengono mai considerate uguali, anche se sono costituite dallo stesso numero di elementi.

Esempio

Il codice riportato di seguito mostra il valore restituito dalle operazioni che usano operatori di uguaglianza, uguaglianza rigorosa e disuguaglianza rigorosa.

```
s1 = new String("4");
s2 = new String("4");
s3 = new String("Ciao");
n = new Number(4);
b = new Boolean(true);
```

```
s1 == s2; // true
s1 == s3; // false
s1 == n; // true
s1 == b; // false
```

```
s1 === s2; // true
s1 === s3; // false
s1 === n; // false
s1 === b; // false
```

```
s1 !== s2; // false
s1 !== s3; // true
s1 !== n; // true
s1 !== b; // true
```

Vedere anche

== (uguaglianza), !=(disuguaglianza), === (uguaglianza rigorosa)

> (maggiore di)

Disponibilità

Flash Player 5.

Uso

espressione1 > *espressione2*

Parametri

espressione1, *espressione2* Un numero intero, un numero in virgola mobile oppure una stringa.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); confronta due espressioni e determina se *espressione1* è maggiore di *espressione2* (true) oppure se *espressione1* è inferiore o uguale a *espressione2* (false).

>= (maggiore di o uguale a)

Disponibilità

Flash Player 4.

File Flash 4:

$x < y$

File Flash 5 convertito:

`Numero(x) > Numero(y)`

Uso

espressione1 >= *espressione2*

Parametri

espressione1, *espressione2* Una stringa, un numero intero, un numero in virgola mobile.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); confronta due espressioni e determina se *espressione1* è maggiore o uguale a *espressione2* (*true*) oppure se *espressione1* è minore di *espressione2* (*false*).

In Flash 5, l'operatore > (maggiore di o uguale a) è un operatore di confronto in grado di gestire diversi tipi di dati. In Flash 4, > è un operatore numerico. I file di Flash 4 importati in ambiente di creazione di Flash 5 vengono sottoposti a un processo di conversione per mantenere l'integrità del tipo di dati.

>> (spostamento a destra bit a bit)

Disponibilità

Flash Player 5.

Uso

espressione1 >> *espressione2*

Parametri

espressione1 Un numero o un'espressione da spostare a destra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); converte *espressione1* ed *espressione2* in numeri interi a 32 bit, quindi sposta a destra tutti i bit contenuti in *espressione1* per il numero di posizioni indicate dal numero intero risultante dalla conversione di *espressione2*. I bit spostati a destra vengono eliminati. Per mantenere il segno dell'*espressione* originale, nei bit a sinistra viene inserito uno 0 se il bit più significativo (il bit più a sinistra) di *espressione1* è 0 oppure un 1 se il bit più significativo è 1. Lo spostamento di un valore di una posizione a destra equivale a una divisione per due con eliminazione del resto.

Esempio

Nell'esempio seguente, 65535 viene convertito in un numero intero a 32 bit e spostato di 8 bit sulla destra.

```
x = 65535 >> 8
```

Il risultato di tale operazione è il seguente:

```
x = 255
```

Il decimale 65535 equivale al numero binario 1111111111111111 (sedici 1), il numero binario 1111111111111111 spostato a destra di 8 bit equivale al numero binario 11111111, che corrisponde al decimale 255. Il bit più significativo è 0 perché i numeri interi sono a 32 bit; pertanto, il bit inserito è 0.

Nell'esempio seguente, -1 viene convertito in un numero intero a 32 bit, 1 viene cioè spostato di un bit a destra.

```
x = -1 >> 1
```

Il risultato di tale operazione è il seguente:

```
x = -1
```

Il decimale -1 equivale al numero binario 11111111111111111111111111111111 (trentadue 1); lo spostamento a destra di un bit fa sì che il bit meno significativo (posto all'estrema destra) venga eliminato e che al posto del bit più significativo venga inserito un 1. Il risultato è il numero binario 11111111111111111111111111111111 (trentadue 1), che rappresenta il numero intero a 32 bit -1.

Vedere anche

>>= (spostamento a destra bit a bit e assegnazione)

>>= (spostamento a destra bit a bit e assegnazione)

Disponibilità

Flash Player 5.

Uso

```
espressione1 ==>> espressione2
```

Parametri

espressione1 Un numero o un'espressione da spostare a sinistra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); esegue un'operazione di spostamento a destra bit a bit e memorizza il contenuto come risultato in *espressione1*.

Esempio

Le due espressioni seguenti si equivalgono:

```
A >>= B
```

```
A = (A >> B)
```

Il codice con commenti riportato di seguito usa l'operatore bit a bit (>>=) e rappresenta anche un esempio dell'uso di tutti gli operatori bit a bit.

```
function convertToBinary(number){
    var result = "";
    for (var i=0; i<32; i++) {
        // Estrae il bit meno significativo usando l'operatore AND bit a bit
        var lsb = number & 1;
        // Aggiunge questo bit alla stringa di risultato
        result = (lsb ? "1" : "0") + result;
        // Sposta il numero a destra di un bit, per accedere al bit successivo
        number >>= 1;}
    return result;
}
trace(convertToBinary(479));
// Restituisce la stringa 0000000000000000000000000111011111
// La stringa precedente è la rappresentazione binaria del
// numero decimale 479
```

Vedere anche

<< (spostamento a sinistra bit a bit)

>>> (spostamento a destra senza segno bit a bit)

Disponibilità

Flash Player 5.

Uso

```
espressione1 >>> espressione2
```

Parametri

espressione1 Un numero o un'espressione da spostare a destra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (bit a bit); la sua funzione è analoga a quella dell'operatore di spostamento a destra bit a bit (>>) tranne per il fatto che non mantiene il segno originale dell'*espressione*, poiché in corrispondenza dei bit a sinistra viene sempre inserito uno 0.

Esempio

Nell'esempio seguente, -1 viene convertito in un numero intero a 32 bit, 1 viene cioè spostato di un bit a destra.

```
x = -1 >>> 1
```

Il risultato di tale operazione è il seguente:

```
x = 2147483647
```

Ciò è dovuto al fatto che il numero decimale -1 equivale al numero binario 11111111111111111111111111111111 (trentadue 1) e che lo spostamento a destra (senza segno) di un bit determina l'eliminazione del bit meno significativo (il bit all'estrema destra) e l'inserimento di uno 0 in corrispondenza del bit più significativo (il bit all'estrema sinistra). Il risultato è il numero binario 01111111111111111111111111111111, che rappresenta il numero intero a 32 bit 2147483647.

Vedere anche

>>= (spostamento a destra bit a bit e assegnazione)

>>>= (spostamento a destra senza segno bit a bit e assegnazione)

Disponibilità

Flash Player 5.

Uso

espressione1>>> = *espressione2*

Parametri

espressione1 Un numero o un'espressione da spostare a sinistra.

espressione2 Un numero o un'espressione convertita in un numero intero da 0 a 31.

Valori restituiti

Nessuno.

Descrizione

Operatore (assegnazione composta bit a bit); esegue un'operazione di spostamento a destra bit a bit senza segno e memorizza il contenuto come risultato di *espressione1*. Le due espressioni seguenti si equivalgono:

A >>>= B
A = (A >>> B)

Vedere anche

>>> (spostamento a destra senza segno bit a bit),>>= (spostamento a destra bit a bit e assegnazione)

Accessibility (oggetto)

L'oggetto Accessibility è un insieme di metodi per la creazione di contenuti accessibili mediante ActionScript. In Flash MX, è disponibile un solo metodo.

Questo oggetto è disponibile in Flash Player 6.

Riepilogo dei metodi validi per l'oggetto Arguments

Proprietà	Descrizione
Accessibility.isActive	Indica se è attualmente attivo uno screen reader.

Accessibility.isActive

Disponibilità

Flash Player 6.

Uso

Accessibility.isActive()

Parametri

Nessuno.

Valori restituiti

Un valore booleano.

Descrizione

Metodo; indica se è attualmente attivo uno screen reader. L'uso di questo metodo è consigliato se si desidera che il filmato abbia un diverso funzionamento quando viene aperto in presenza di uno screen reader.

Vedere anche

System.capabilities.hasAccessibility

add

Disponibilità

Flash Player 4.

Uso

stringa1 add *stringa2*

Parametri

stringa1, *stringa2* Una stringa.

Valori restituiti

Nessuno.

Descrizione

Operatore; concatena (combina) due o più stringhe. L'operatore add sostituisce l'operatore & di Flash 4. Nei file Flash 4, l'operatore & viene automaticamente sostituito con l'operatore add di concatenazione delle stringhe al momento dell'importazione in ambiente Flash 5. L'operatore add in Flash 5 è diventato obsoleto e si consiglia di usare l'operatore + per la creazione di contenuti per Flash Player 5 o Flash Player 6. Usare l'operatore add per concatenare le stringhe quando si creano contenuti per Flash 4 o versioni precedenti di Player.

Vedere anche

+ (addizione)

and

Disponibilità

Flash Player 4.

Uso

condizione1 and condizione2

Parametri

condizione1,condizione2 Condizioni o espressioni che restituiscono `true` o `false`.

Valori restituiti

Nessuno.

Descrizione

Operatore; esegue un'operazione di AND logico in ambiente Flash Player 4. Se entrambe le espressioni restituiscono `true` l'intera espressione è `true`. Questo operatore è diventato obsoleto in Flash 5 e si consiglia di usare il nuovo operatore `&&`.

Vedere anche

`&&` (cortocircuito AND)

arguments (oggetto)

L'oggetto `Arguments` è una matrice di valori inviati come parametri a una delle funzioni. Tutte le volte che si richiama una funzione in ActionScript viene automaticamente creato un oggetto `Arguments` per la funzione. Viene creata anche una variabile locale `arguments` che consente di fare riferimento a tale oggetto.

L'oggetto `Arguments` è disponibile in Flash Player 6.

Riepilogo delle proprietà valide per l'oggetto Arguments

Proprietà	Descrizione
<code>arguments.callee</code>	Si riferisce alla funzione che viene richiamata.
<code>arguments.caller</code>	Si riferisce alla funzione che contiene l'istruzione.
<code>arguments.length</code>	Il numero di parametri inviati a una funzione.

arguments.callee

Disponibilità

Flash Player 5.

Uso

`arguments.callee`

Descrizione

Proprietà; si riferisce alla funzione attualmente richiamata.

Esempio

È possibile usare la proprietà `arguments.callee` per creare una funzione anonima ricorsiva, come nell'esempio seguente:

```
factorial = function (x) {
    if (x <= 1) {
        return 1;
    } else {
        return x * arguments.callee(x-1);
    }
};
```

Di seguito, viene riportato il nome di una funzione ricorsiva:

```
function factorial (x) {
    if (x <= 1) {
        return 1;
    } else {
        return x * factorial(x-1);
    }
}
```

arguments.caller

Disponibilità

Flash Player 6.

Uso

`arguments.caller`

Descrizione

Proprietà; si riferisce all'oggetto `Arguments` della funzione contenente l'istruzione.

arguments.length

Disponibilità

Flash Player 6.

Uso

`arguments.length`

Descrizione

Proprietà; il numero di parametri attualmente passati a una funzione.

Array (oggetto)

L'oggetto `Array` consente di accedere alle matrici e di gestirle. Una matrice è un oggetto le cui proprietà sono identificate da un numero che ne rappresenta la posizione all'interno della matrice. Tale numero è detto anche *indice*. Tutte le matrici hanno base zero; ciò significa che il primo elemento nella matrice è `[0]`, il secondo è `[1]` e così via. Nell'esempio seguente, `myArray` contiene i mesi dell'anno.

```
myArray[0] = "gennaio"
myArray[1] = "febbraio"
myArray[2] = "marzo"
myArray[3] = "aprile"
```

Per creare un oggetto Array, usare la funzione di costruzione `new Array` o l'operatore di accesso alla matrice (`[]`). Per accedere agli elementi di una matrice, usare il relativo operatore di accesso (`[]`).

In Flash MX, l'oggetto Array è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

Riepilogo dei metodi validi per l'oggetto Array

Metodo	Descrizione
<code>Array.concat</code>	Concatena i parametri e li restituisce come nuova matrice.
<code>Array.join</code>	Restituisce una stringa che contiene tutti gli elementi della matrice.
<code>Array.pop</code>	Rimuove l'ultimo elemento della matrice e restituisce il valore corrispondente.
<code>Array.push</code>	Aggiunge uno o più elementi in fondo alla matrice e ne restituisce la nuova dimensione.
<code>Array.reverse</code>	Inverte la direzione di una matrice.
<code>Array.shift</code>	Rimuove il primo elemento da una matrice e ne restituisce il valore corrispondente.
<code>Array.splice</code>	Estrae una sezione da una matrice e la restituisce come nuova matrice.
<code>Array.sort</code>	Ordina una matrice in posizione.
<code>Array.sortOn</code>	Ordina una matrice in base a uno dei campi.
<code>Array.splice</code>	Aggiunge e/o rimuove gli elementi di una matrice.
<code>Array.toString</code>	Restituisce un valore stringa che rappresenta gli elementi dell'oggetto Array.
<code>Array.unshift</code>	Aggiunge uno o più elementi all'inizio della matrice e ne restituisce la nuova dimensione.

Riepilogo delle proprietà valide per l'oggetto Array

Proprietà	Descrizione
<code>Array.length</code>	Restituisce le dimensioni della matrice.

Funzione di costruzione dell'oggetto Array

Disponibilità

Flash Player 5.

Uso

```
new Array();
```

```
new Array(lunghezza)
```

```
new Array(elemento0, elemento1, elemento2, ... elementoN)
```

Parametri

lunghezza Numero intero che indica il numero di elementi di una matrice. Nel caso di elementi non contigui, il parametro *lunghezza* specifica il numero d'indice dell'ultimo elemento nella matrice più 1.

elemento0...elementoN Un elenco di due o più valori arbitrari. I valori possono essere rappresentati numeri, stringhe, oggetti oppure altre matrici. Al primo elemento della matrice corrisponde sempre un indice o posizione 0.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; consente di creare una matrice. È possibile usare la funzione di costruzione per creare diversi tipi di matrice: una matrice vuota, una matrice con una lunghezza specifica contenente elementi senza valore o una matrice contenente elementi con valori specifici.

Uso 1: se non si specifica alcun parametro, viene creata una matrice con lunghezza 0.

Uso 2: se si specifica solo una lunghezza, viene creata una matrice con il numero di *lunghezza* di elementi che non contengono valori.

Uso 3: se si usano i parametri dell'*elemento* per specificare i valori, viene creata una matrice con valori specifici.

Esempio

Uso 1: l'esempio seguente crea un nuovo oggetto Array la cui dimensione iniziale è uguale a 0.

```
myArray = new Array();
```

Uso 3: l'esempio seguente crea il nuovo oggetto Array *go-gos*, con una dimensione iniziale di 5.

```
go_gos = new Array("Iolanda", "Gina", "Clara", "Carlotta", "Sara");  
trace(go_gos.join(" + "));
```

Gli elementi iniziali della matrice *go_gos* sono identificati nel modo seguente:

```
go_gos[0] = "Iolanda";  
go_gos[1] = "Gina";  
go_gos[2] = "Clara";  
go_gos[3] = "Carlotta";  
go_gos[4] = "Sara";
```

Il seguente codice aggiunge un quinto elemento alla matrice *go-gos* e modifica il primo elemento:

```
go_gos[5] = "Marta";  
go_gos[1] = "Nina";  
trace(go_gos.join(" + "));
```

Vedere anche

`Array.length, []` (accesso matrice)

Array.concat

Disponibilità

Flash Player 5.

Uso

```
myArray.concat(valore0, valore1, ... valoreN)
```

Parametri

valore0, ... valoreN Numeri, elementi o stringhe da concatenare in una nuova matrice.

Valori restituiti

Nessuno.

Descrizione

Metodo; concatena gli eventuali elementi specificati nei parametri agli elementi presenti in *myArray* e crea una nuova matrice. Se i parametri di *valore* specificano una matrice, vengono concatenati gli elementi stessi al posto della matrice. La matrice *myArray* non viene modificata.

Esempio

Il codice seguente concatena due matrici.

```
alpha = new Array("a","b","c");
numeric = new Array(1,2,3);
alphaNumeric=alpha.concat(numeric);
trace(alphaNumeric);
// Crea la matrice ["a","b","c",1,2,3]
```

Il codice seguente concatena tre matrici:

```
num1=[1,3,5];
num2=[2,4,6];
num3=[7,8,9];
nums=num1.concat(num2,num3)
trace(nums);
// Crea la matrice [1,3,5,2,4,6,7,8,9]
```

Le matrici nidificate non vengono disposte su un unico livello come le matrici normali. Gli elementi di una matrice nidificata non vengono divisi in elementi separati nella matrice *x*, come mostra l'esempio riportato di seguito.

```
a = new Array ("a","b","c");
n = new Array(1, [2, 3], 4);
// 2 e 3 sono elementi di una matrice nidificata
x = a.concat(n);
x[0] = "a"
x[1] = "b"
x[2] = "c"
x[3] = 1
x[4] = 2, 3
x[5] = 4
```

Array.join

Disponibilità

Flash Player 5.

Uso

```
myArray.join([separatore])
```

Parametri

separatore Carattere o stringa che separa gli elementi della matrice nella stringa restituita. Se si omette questo parametro, viene usata una virgola come separatore predefinito.

Valori restituiti

Nessuno.

Descrizione

Metodo; converte gli elementi di una matrice in stringhe, inserisce il separatore specificato tra gli elementi, li concatena e restituisce la stringa risultante. Una matrice nidificata è sempre separata mediante una virgola e tramite il separatore inviato al metodo *join*.

Esempio

L'esempio seguente crea una matrice con tre elementi, quindi li unisce tre volte (usando il separatore predefinito, una virgola e uno spazio, e il segno più) e li visualizza nella finestra Output:

```
a = new Array("Terra","Luna","Sole")
trace(a.join());
// Restituisce Terra, Luna, Sole
trace(a.join(" - "));
// Restituisce Terra - Luna - Sole
trace(a.join(" + "));
// Restituisce Terra + Luna + Sole
```

Array.length

Disponibilità

Flash Player 5.

Uso

```
myArray.length
```

Descrizione

Proprietà; contiene la dimensione della matrice. Questa proprietà viene aggiornata automaticamente quando alla matrice vengono aggiunti nuovi elementi. Quando si assegna un valore a un elemento di una matrice (ad esempio, `myArray[indice] = valore`), se `indice` è un numero e `indice+1` ha un valore superiore a quello della proprietà `length`, quest'ultima viene aggiornata in base a `indice + 1`.

Esempio

Il codice seguente esemplifica la modalità di aggiornamento della proprietà `length`.

```
// La lunghezza iniziale è 0
myArray = new Array();
myArray[0] = 'a';
//myArray.length viene aggiornato a 1
myArray[1] = 'b';
//myArray.length viene aggiornato a 2
myArray[9] = 'c';
//myArray.length viene aggiornato a 10
```

Array.pop

Disponibilità

Flash Player 5.

Uso

```
myArray.pop()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove l'ultimo elemento della matrice e restituisce il valore di tale elemento.

Esempio

Il codice seguente crea la matrice `myPets` che contiene quattro elementi, quindi rimuove l'ultimo di essi.

```
myPets = ["topo", "cane", "uccello", "pesce"];  
popped = myPets.pop();  
trace(popped);  
// Restituisce pesce
```

Array.push

Disponibilità

Flash Player 5.

Uso

```
myArray.push(valore,...);
```

Parametri

valore Uno o più valori da aggiungere in fondo alla matrice.

Valori restituiti

Restituisce la lunghezza della nuova matrice.

Descrizione

Metodo; aggiunge uno o più elementi in fondo alla matrice e ne restituisce la nuova dimensione.

Esempio

Nell'esempio seguente, viene creata la matrice `myPets` con due elementi, `topo` e `cane`. La seconda riga aggiunge due elementi alla matrice. In seguito, viene richiamato il metodo `push`. La variabile sottoposta al metodo `push` contiene quattro elementi. Poiché mediante il metodo `push` viene restituita la nuova lunghezza della matrice, l'azione `trace` nell'ultima riga invia la nuova lunghezza di `myPets` (4) alla finestra Output:

```
myPets = ["topo", "cane"];  
pushed = myPets.push("uccello", "pesce");  
trace(pushes);
```

Array.reverse

Disponibilità

Flash Player 5.

Uso

```
myArray.reverse();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; inverte la matrice in posizione.

Esempio

L'esempio seguente illustra l'uso del metodo `Array.reverse`:

```
var numbers = [1, 2, 3, 4, 5, 6];
trace(numbers.join());
numbers.reverse();
trace(numbers.join());
```

Output:

```
1,2,3,4,5,6
6,5,4,3,2,1
```

Array.shift

Disponibilità

Flash Player 5.

Uso

```
myArray.shift()
```

Parametri

Nessuno.

Valori restituiti

Il primo elemento della matrice.

Descrizione

Metodo; rimuove il primo elemento della matrice e lo restituisce.

Esempio

Il codice seguente crea la matrice `myPets`, quindi rimuove il primo elemento dalla matrice e lo assegna alla variabile `shifted`.

```
myPets = ["topo", "cane", "uccello", "pesce"];
shifted = myPets.shift();
trace(shifted);
// Restituisce topo
```

Vedere anche

`Array.pop`

Array.splice

Disponibilità

Flash Player 5.

Uso

```
myArray.splice(inizio, fine)
```

Parametri

inizio Un numero che specifica l'indice del punto di inizio della porzione. Se *inizio* è un numero negativo, il punto di inizio è specificato a partire dal fondo della matrice, dove -1 è l'ultimo elemento.

fine Un numero che specifica l'indice del punto finale della porzione. Se questo parametro viene omissso, la porzione include tutti gli elementi dall'inizio alla fine della matrice. Se *fine* è un numero negativo, il punto finale è specificato a partire dal fondo della matrice, dove -1 è l'ultimo elemento.

Valori restituiti

Nessuno.

Descrizione

Metodo; estrae una porzione o una sottostringa della matrice e la restituisce come nuova matrice senza modificare quella originale. La matrice restituita contiene l'elemento *inizio* e tutti gli elementi compresi tra questo e l'elemento *fine*, che è escluso.

Array.sort

Disponibilità

Flash Player 5.

Uso

```
myArray.sort([funzioneConfronto])
```

Parametri

funzioneConfronto Una funzione opzionale di confronto usata per determinare il tipo di ordinamento degli elementi di una matrice. Dati gli elementi A e B, il parametro *funzioneOrdine* può avere uno dei tre valori riportati di seguito:

- -1 se nella sequenza ordinata A precede B
- 0 se A = B
- 1 se nella sequenza ordinata A segue B

Valori restituiti

Nessuno.

Descrizione

Metodo; ordina la matrice in posizione senza farne una copia. Se viene omissso il parametro *funzioneConfronto*, l'applicazione Flash ordina gli elementi mediante l'operatore di confronto <.

Esempio

Nell'esempio seguente viene usato il metodo `Array.sort` senza specificare il parametro *funzioneConfronto*.

```
var fruits = ["arance", "mele", "fragole", "ananas", "ciliegie"];  
trace(fruits.join());  
fruits.sort();  
trace(fruits.join());
```

Output:

```
arance,mele,fragole,ananas,ciliegie  
visualizza mele,ciliegie,arance,ananas,fragole
```


Nell'esempio seguente, viene usato il metodo `Array.sort` con una funzione di ordinamento specifica.

```
var passwords = [
    "tino:zoo",
    "lino:bar",
    "aldo:carte",
    "carlo:dado",
    "franco:1234"
];
function order (a, b) {
    // Le voci da ordinare sono nel formato
    // nome:password
    // Ordina usando solo la porzione del nome
    // della voce come chiave.
    var nome1 = a.split(':')[0];
    var nome2 = b.split(':')[0];
    if (nome1 < nome2) {
        return -1;
    } else if (nome1 > nome2) {
        return 1;
    } else {
        return 0;
    }
}
for (var i=0; i< password.length; i++) {
    trace (passwords.join());
}
passwords.sort(order);
trace ("Ordinato:");
for (var i=0; i< password.length; i++) {
    trace (passwords.join());
}
```

Mediante l'esecuzione del codice precedente, viene visualizzato il risultato seguente nella finestra Output.

```
franco:1234
tino:zoo
aldo:carte
lino:bar
carlo:dado
```

Array.sortOn

Disponibilità

Flash Player 6.

Uso

`Array.sortOn(nomeCampo)`

Parametri

nomeCampo Una stringa che identifica un campo in un elemento dell'Array da usare come valore di ordinamento.

Valori restituiti

Nessuno.

Descrizione

Metodo; ordina gli elementi in una matrice basata su un campo della matrice. Se non viene inviato alcun *nomeCampo*, la funzione non ha effetto. Se vengono passati più parametri *nomeCampo*, il primo viene convertito nel valore di una stringa mentre gli altri parametri vengono ignorati.

Se il campo specificato nel parametro *nomeCampo* non è contenuto in nessuno degli elementi sottoposti a confronto, l'ordinamento usa l'impostazione predefinita del metodo `Array.sort`.

Esempio

Nell'esempio seguente, vengono creati una nuova matrice e un nuovo ordinamento in base al campo `città`:

```
var recArray = new Array();
recArray.push( { name: "ale", city: "rieti", zip: 02100 } );
recArray.push( { name: "ciro", city: "napoli", zip: 80133 } );
recArray.push( { name: "sauro", city: "civitanova", zip: 12345 } );
recArray.sortOn("city");
//Vengono restituiti i seguenti risultati:
recArray[0] = name: "sauro", city: "civitanova", zip: 94010
recArray[1] = name: "ciro", city: "napoli", zip: 80133
recArray[2] = name: "ale", city: "rieti", zip: 68144
```

Vedere anche

`Array.sort`

Array.splice

Disponibilità

Flash Player 5.

Uso

```
myArray.splice(inizio, numeriEliminati, valore0, valore1...valoreN)
```

Parametri

inizio L'indice dell'elemento della matrice in corrispondenza del quale ha inizio l'operazione di inserimento o eliminazione.

numeriEliminati Il numero di elementi da eliminare. Tale numero include l'elemento specificato dal parametro *inizio*. Se non viene specificato un valore per *numeriEliminati*, il metodo elimina tutti i valori, dall'elemento *inizio* all'ultimo elemento della matrice. Se il valore è 0, nessun elemento viene eliminato.

valore Zero o più valori da inserire nella matrice in corrispondenza del punto di inserimento indicato dal parametro *inizio*. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge e/o rimuove gli elementi da una matrice. Questo metodo modifica la matrice senza farne una copia.

Array.toString

Disponibilità

Flash Player 5.

Uso

```
myArray.toString()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce un valore stringa che rappresenta gli elementi dell'oggetto Array specificato. Ciascun elemento della matrice, dall'indice 0 fino all'indice `myArray.length-1`, viene convertito in una stringa concatenata e separata da virgole.

Esempio

Nell'esempio seguente, la matrice `myArray` viene creata, convertita in una stringa e visualizza 1,2,3,4,5 nella finestra Output.

```
myArray = new Array();
myArray[0] = 1;
myArray[1] = 2;
myArray[2] = 3;
myArray[3] = 4;
myArray[4] = 5;

trace(myArray.toString());
```

Array.unshift

Disponibilità

Flash Player 5.

Uso

```
myArray.unshift(valore1, valore2, ... valoreN)
```

Parametri

`valore1, ... valoreN` Uno o più numeri, elementi oppure variabili da inserire all'inizio della matrice.

Valori restituiti

La nuova lunghezza della matrice.

Descrizione

Metodo; aggiunge uno o più elementi all'inizio della matrice e ne restituisce la nuova dimensione.

asfunction

Disponibilità

Flash Player 5.

Uso

`asfunction:funzione, "parametro"`

Parametri

funzione Un identificatore di una funzione.

parametro Una stringa passata alla funzione indicata nel parametro *funzione*.

Valori restituiti

Nessuno.

Descrizione

Protocollo; un protocollo speciale per URL dei campi di testo HTML. Nei campi di testo HTML, il testo può contenere un collegamento ipertestuale basato su un tag HTML A. L'attributo HREF del tag A contiene un URL per un protocollo standard, ad esempio HTTP, HTTPS o FTP. Il protocollo `asfunction` è un protocollo aggiuntivo specifico dell'applicazione Flash che fa sì che il collegamento attivi una funzione ActionScript.

Esempio

Nell'esempio seguente, la funzione `MyFunc` è definita nelle prime tre righe del codice. La variabile `textField` è associata a un campo di testo HTML. Il testo "Fai clic qui!" è un collegamento ipertestuale all'interno del campo di testo. La funzione `MyFunc` viene richiamata quando l'utente seleziona il collegamento ipertestuale:

```
funzione Myfunc(arg){  
    trace ("Hai selezionato l'elemento!Argument era "+arg);  
}  
myTextField.text ="<A HREF=\"asfunction:MyFunc,Foo \">>Fai clic qui!</A>";
```

Quando si seleziona il collegamento ipertestuale, nella finestra Output vengono visualizzati i risultati seguenti:

```
Hai selezionato l'elemento! Il parametro era Foo
```

Boolean (funzione)

Disponibilità

Flash Player 5.

Uso

`Boolean(espressione)`

Parametri

espressione Un'espressione per convertire un valore booleano.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte il parametro *espressione* in un valore booleano e restituisce un valore come riportato di seguito:

- Se l'*espressione* è un valore booleano, il valore restituito è *espressione*.
- Se l'*espressione* è un numero, il valore restituito è `true` se il numero è diverso da zero, altrimenti il valore restituito è `false`.
- Se l'*espressione* è una stringa, viene attivato il metodo `toNumber` e il valore restituito è `true` se il numero non è zero, altrimenti il valore restituito è `false`.
- Se l'*espressione* non è definita, il valore restituito è `false`.
- Se l'*espressione* è un clip filmato o un oggetto, il valore restituito è `true`.

Boolean (oggetto)

L'oggetto Boolean è un semplice oggetto wrapper con le stesse funzionalità dell'oggetto Boolean standard di JavaScript. L'oggetto Boolean consente di recuperare il tipo di dati di base o la rappresentazione in formato stringa dell'oggetto Boolean. In Flash MX, l'oggetto Boolean è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

Per creare un'istanza dell'oggetto Boolean, è necessario usare la funzione di creazione `new Boolean()` prima di attivare i metodi corrispondenti.

Riepilogo dei metodi validi per l'oggetto Boolean

Metodo	Descrizione
<code>Boolean.toString</code>	Restituisce la rappresentazione in formato stringa (<code>true</code>) o (<code>false</code>) dell'oggetto Boolean.
<code>Boolean.valueOf</code>	Restituisce il tipo di valore di base dell'oggetto Boolean specificato.

Funzione di costruzione dell'oggetto Boolean

Disponibilità

Flash Player 5.

Uso

```
nuovo Boolean([x])
```

Parametri

x Qualsiasi espressione. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un'istanza dell'oggetto Boolean. Se si omette il parametro *x*, l'oggetto Boolean viene inizializzato con il valore `false`. Se si indica un valore per il parametro *x*, esso viene analizzato dal metodo che restituisce il valore booleano risultante, in base ai criteri contenuti in `Boolean` (funzione).

Nota: per mantenere la compatibilità con Flash Player 4, la gestione delle stringhe in relazione all'oggetto Boolean non è conforme allo standard ECMA-262.

Esempio

Il codice seguente crea un nuovo oggetto Boolean vuoto chiamato `myBoolean`.

```
myBoolean = new Boolean();
```

Boolean.toString

Disponibilità

Flash Player 5.

Uso

```
myBoolean.toString()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una rappresentazione in formato stringa, `true` o `false`, dell'oggetto Boolean.

Boolean.valueOf

Disponibilità

Flash Player 5.

Uso

```
Boolean.valueOf()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il tipo di valore originario dell'oggetto Boolean specificato.

break

Disponibilità

Flash Player 4.

Uso

```
break
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; questa azione è sempre associata a un ciclo (`for`, `for..in`, `do while` o `while`) oppure a una sequenza di istruzioni associate a un caso specifico nell'ambito di un'azione `switch`. L'azione `break` indica a Flash di ignorare il resto del corpo del ciclo, interromperne l'iterazione ed eseguire l'istruzione che segue l'istruzione del ciclo. Quando si usa l'azione `break`, l'interprete Flash ignora il resto delle istruzioni contenute nello stesso blocco `case` e passa all'istruzione successiva al termine dell'azione `switch`. L'azione `break` consente di uscire da una serie di cicli nidificati.

Esempio

L'esempio seguente usa l'azione `break` per uscire da un ciclo altrimenti infinito.

```
i = 0;
while (true) {
    if (i >= 100) {
        break;
    }
    i++;
}
```

Vedere anche

`for`, `for..in`, `do while`, `while`, `switch`, `case`

Button (oggetto)

Tutti i simboli dei pulsanti in un filmato Flash sono istanze dell'oggetto `Button`. È possibile assegnare a un pulsante un nome di istanza nella finestra di ispezione Proprietà e gestire i pulsanti tramite ActionScript, usando i metodi e le proprietà dell'oggetto `Button`. I nomi delle istanze `Button` vengono visualizzati in Esplora filmato e nella finestra di dialogo Inserisci percorso target nel pannello Azioni.

L'oggetto `Button` eredita proprietà o metodi dell'oggetto `Object`.

L'oggetto `Button` è supportato in Flash Player 6.

Riepilogo dei metodi validi per l'oggetto Button

Metodo	Descrizione
<code>Button.getDepth</code>	Restituisce la profondità di un'istanza di pulsante.

Riepilogo delle proprietà valide per l'oggetto Button

Proprietà	Descrizione
<code>Button._alpha</code>	Il valore di trasparenza di un'istanza di pulsante.
<code>Button.enabled</code>	Indica se il pulsante è attivo.
<code>Button._focusrect</code>	Indica se un pulsante è delimitato da un rettangolo giallo di attivazione.
<code>Button._height</code>	L'altezza di un'istanza di pulsante, in pixel.
<code>Button._highquality</code>	Indica la qualità di rendering del filmato.
<code>Button._name</code>	Il nome dell'istanza di un pulsante.
<code>Button._parent</code>	Un riferimento all'istanza del clip filmato principale di questa istanza.
<code>Button._quality</code>	Indica la qualità di rendering del filmato.

Proprietà	Descrizione
<code>Button._rotation</code>	Il grado di rotazione di un'istanza di pulsante.
<code>Button._soundbuftime</code>	Numero di secondi di precaricamento di un suono.
<code>Button.tabEnabled</code>	Indica se un pulsante è incluso nell'ordinamento di tabulazione automatico.
<code>Button.tabIndex</code>	Indica l'ordine di tabulazione di un oggetto.
<code>Button._target</code>	Il percorso target di un'istanza di pulsante.
<code>Button.trackAsMenu</code>	Indica se altri pulsanti possono ricevere eventi associati al rilascio del pulsante del mouse.
<code>Button._url</code>	L'URL del file SWF che ha creato l'istanza di pulsante.
<code>Button.useHandCursor</code>	Indica se il cursore manuale viene visualizzato quando si passa il mouse su un pulsante.
<code>Button._visible</code>	Un valore booleano che determina se un'istanza è nascosta o visibile.
<code>Button._width</code>	La larghezza di un'istanza di pulsante, in pixel.
<code>Button._x</code>	La coordinata x di un'istanza di pulsante.
<code>Button._xmouse</code>	La coordinata x del cursore relativa a un'istanza di pulsante.
<code>Button._xscale</code>	Il valore che specifica la percentuale di modifica orizzontale in scala di un'istanza di pulsante.
<code>Button._y</code>	La coordinata y di un'istanza di pulsante.
<code>Button._ymouse</code>	La coordinata y del cursore relativa a un'istanza di pulsante.
<code>Button._yscale</code>	Il valore che specifica la percentuale di modifica verticale in scala di un'istanza di pulsante.

Riepilogo eventi dell'oggetto Button

Nella tabella riportata di seguito è elencato il riepilogo degli eventi dell'oggetto Button.

Metodo	Descrizione
<code>Button.onDragOut</code>	Invocato quando si seleziona il pulsante con il mouse e si trascina il puntatore fuori dall'area del pulsante.
<code>Button.onDragOver</code>	Invocato quando il pulsante del mouse viene premuto all'esterno dell'area del pulsante e il puntatore viene nuovamente trascinato sopra il pulsante.
<code>Button.onKeyUp</code>	Invocato quando viene rilasciato un tasto.
<code>Button.onKillFocus</code>	Invocato quando un pulsante viene disattivato.
<code>Button.onPress</code>	Invocato quando il mouse viene premuto mentre il puntatore si trova sopra un pulsante.
<code>Button.onRelease</code>	Invocato quando il mouse viene rilasciato mentre il puntatore si trova sopra un pulsante.
<code>Button.onReleaseOutside</code>	Invocato quando il pulsante del mouse viene rilasciato mentre il puntatore si trova al di fuori dell'area del pulsante, dopo la pressione del pulsante mentre il puntatore è all'interno di essa.
<code>Button.onRollOut</code>	Invocato quando il puntatore del mouse viene spostato fuori dall'area del pulsante.
<code>Button.onRollOver</code>	Invocato quando il puntatore del mouse scorre sopra il pulsante.
<code>Button.onSetFocus</code>	Invocato quando un pulsante è l'elemento attivo e un tasto viene rilasciato.

Button._alpha

Disponibilità

Flash Player 6.

Uso

myButton._alpha

Descrizione

Proprietà; imposta o recupera la trasparenza alfa (*valore*) del clip filmato specificato tramite *Button*. I valori validi vanno da 0 (completamente trasparente) a 100 (completamente opaco). Gli oggetti di un pulsante con *_alpha* impostata su 0 sono attivi anche se sono invisibili.

Esempio

L'esempio seguente mostra l'impostazione della proprietà *_alpha* di un pulsante denominato *star* al 30%.

```
on(release) {  
    star._alpha = 30;  
}
```

Button.enabled

Disponibilità

Flash Player 6.

Uso

myButton.enabled

Descrizione

Proprietà; un valore booleano che specifica se un pulsante è attivato. Il valore predefinito è *true*.

Button._focusrect

Disponibilità

Flash Player 6.

Uso

myButton._focusrect

Descrizione

Proprietà; un valore booleano che specifica se un pulsante è delimitato da un rettangolo giallo quando è attivato tramite tastiera. Questa proprietà ha la priorità sulla proprietà globale *_focusrect*.

Button.getDepth

Disponibilità

Flash Player 6.

Uso

```
myButton.getDepth()
```

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce la profondità di un'istanza di pulsante.

Button._height

Disponibilità

Flash Player 6.

Uso

```
myButton._height
```

Descrizione

Proprietà; imposta e richiama l'altezza del pulsante in pixel.

Esempio

Il codice seguente imposta l'altezza e la larghezza di un pulsante quando si fa clic con il pulsante del mouse.

```
myButton._width = 200;  
myButton._height = 200;
```

Button._highquality

Disponibilità

Flash Player 6.

Uso

```
myButton._highquality
```

Descrizione

Proprietà (globale); specifica il livello di antialiasing applicato al filmato corrente. Specificare 2 (Ottima) per applicare un livello di alta qualità con smussatura delle bitmap sempre attiva. Specificare 1 (Alta qualità) per applicare l'antialiasing; in tal modo le bitmap verranno smussate se il filmato non contiene animazioni. Specificare 0 (Bassa qualità) per impedire l'antialiasing.

Esempio

```
_highquality = 1;
```

Vedere anche

```
_quality, toggleHighQuality
```

Button._name

Disponibilità

Flash Player 6.

Uso

myButton._name

Descrizione

Proprietà; restituisce il nome dell'istanza del pulsante specificato mediante *myButton*.

Button.onDragOut

Disponibilità

Flash Player 6.

Uso

myButton.onDragOut

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si seleziona un pulsante con il mouse e si trascina il puntatore all'esterno dell'area del pulsante.

Button.onDragOver

Disponibilità

Flash Player 6.

Uso

myButton.onDragOver

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si preme il pulsante del mouse e si trascina il puntatore all'esterno, quindi nuovamente all'interno dell'area del pulsante.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onKeyDown` che invia un'azione trace alla finestra Output:

```
myButton.onDragOver = function () {  
    trace ("richiamato onDragOver");  
};
```

Vedere anche

`Button.onKeyUp`

Button.onKeyDown

Disponibilità

Flash Player 6.

Uso

`myButton.onKeyDown`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un pulsante viene attivato dalla pressione di un tasto.

L'evento `onKeyDown` viene invocato senza parametri. È possibile usare i metodi `Key.getAscii` e `Key.getCode` per determinare quale tasto è stato premuto.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onKeyDown` che invia un'azione trace alla finestra Output:

```
myButton.onKeyDown = function () {  
    trace ("richiamato onKeyDown");  
};
```

Vedere anche

`Button.onKeyUp`

Button.onKeyUp

Disponibilità

Flash Player 6.

Uso

`myButton.onKeyUp`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un pulsante è l'elemento attivo e un tasto viene rilasciato. L'evento `onKeyUp` viene invocato senza parametri. È possibile usare i metodi `Key.getAscii` e `Key.getCode` per determinare quale tasto è stato premuto.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onKeyPress` che invia un'azione trace alla finestra Output:

```
myButton.onKeyUp = function () {  
    trace ("richiamato onKeyUp");  
};
```

Button.onKillFocus

Disponibilità

Flash Player 6.

Uso

```
myButton.onKillFocus = function (nuovoAttivo) {  
    istruzioni;  
};
```

Parametri

nuovoAttivo L'oggetto che viene attivato.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un pulsante non è più attivato dalla tastiera. Il metodo `onKillFocus` riceve il parametro *nuovoAttivo*, che corrisponde al nuovo oggetto attivato. Se nessun oggetto viene attivato, il valore di *nuovoAttivo* è `null`.

Button.onPress

Disponibilità

Flash Player 6.

Uso

```
myButton.onPress
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si preme un tasto. È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onPress` che invia un'azione trace alla finestra `Output`.

```
myButton.onPress = function () {  
    trace ("richiamato onPress");  
};
```

Button.onRelease

Disponibilità

Flash Player 6.

Uso

```
myButton.onRelease
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si rilascia un pulsante.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onRelease` che invia un'azione trace alla finestra `Output`.

```
myButton.onRelease = function () {  
    trace ("richiamato onRelease");  
};
```

Button.onReleaseOutside

Disponibilità

Flash Player 6.

Uso

```
myButton.onReleaseOutside
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si rilascia il pulsante del mouse mentre il puntatore si trova all'esterno dell'area del pulsante, selezionato quando il puntatore si trovava all'interno dell'area.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onReleaseOutside` che invia un'azione trace alla finestra Output.

```
myButton.onReleaseOutside = function () {  
    trace ("richiamato onReleaseOutside");  
};
```

Button.onRollOut

Disponibilità

Flash Player 6.

Uso

```
myButton.onRollOut
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore viene trascinato all'esterno dell'area di un pulsante.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onRollOut` che invia un'azione trace alla finestra Output.

```
myButton.onRollOut = function () {  
    trace ("richiamato onRollOut");  
};
```

Button.onRollOver

Disponibilità

Flash Player 6.

Uso

```
myButton.onRollOver
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore viene trascinato fuori dall'area del pulsante.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onRollOver` che invia un'azione trace alla finestra `Output`.

```
myButton.onRollOver = function () {  
    trace ("richiamato onRollOver");  
};
```

Button.onSetFocus

Disponibilità

Flash Player 6.

Uso

```
myButton.onSetFocus = function(vecchioAttivo){  
    istruzioni;  
};
```

Parametri

vecchioAttivo L'oggetto attivato precedentemente da tastiera.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si attiva un pulsante da tastiera. Il parametro *vecchioAttivo* si riferisce all'oggetto attivato precedentemente da tastiera. Ad esempio, se l'utente preme il tasto di tabulazione per spostare il punto attivo da un campo di testo a un pulsante, *vecchioAttivo* contiene l'istanza del campo di testo.

Se precedentemente non è stato attivato alcun oggetto, *vecchioAttivo* contiene un valore `null`.

Button._parent

Disponibilità

Flash Player 6.

Uso

```
_parent.property
```

Descrizione

Proprietà; specifica o restituisce un riferimento al clip filmato contenente l'oggetto o il clip filmato corrente. L'oggetto attuale è quello contenente il codice di ActionScript che si riferisce a `_parent`. Usare `_parent` per specificare il percorso relativo a clip filmato o oggetti gerarchicamente superiori all'oggetto o al clip filmato attuale. La proprietà `_parent` consente di risalire nell'elenco di visualizzazione di diversi livelli, come nell'esempio seguente:

```
_parent._parent._alpha = 20;
```

Vedere anche

```
_root, targetPath
```


Button._quality

Disponibilità

Flash Player 6.

Uso

myButton._quality

Descrizione

Proprietà (globale); imposta o recupera la qualità di rendering usata per un filmato. È sempre disponibile una copia dei caratteri dispositivo che non vengono alterati dalla proprietà `_quality`.

È possibile impostare la proprietà `_quality` con i valori riportati di seguito:

- "LOW" Bassa qualità di rendering. I grafici non sono sottoposti ad antialiasing e le bitmap non vengono smussate.
- "MEDIUM" Media qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 2x2 ma le bitmap non vengono smussate. Questo valore è adatto a filmati che non contengono testo.
- "HIGH" Alta qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 e le bitmap vengono smussate se il filmato è statico. Questa è l'impostazione predefinita per la qualità di rendering usata da Flash.
- "BEST" Ottima qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 e le bitmap vengono sempre smussate.

Esempio

Nell'esempio seguente, la qualità di rendering è impostata su LOW:

```
myButton._quality = "LOW";
```

Vedere anche

`_highquality`, `toggleHighQuality`

Button._rotation

Disponibilità

Flash Player 6.

Uso

myButton._rotation

Descrizione

Proprietà; specifica la rotazione del pulsante in gradi.

Button._soundbuftime

Disponibilità

Flash Player 6.

Uso

myButton._soundbuftime

Descrizione

Proprietà (globale); numero intero che specifica in secondi il tempo di prebuffer di un suono prima che abbia inizio lo streaming.

Button.tabEnabled

Disponibilità

Flash Player 6.

Uso

myButton.tabEnabled

Descrizione

Proprietà; è possibile impostarla su un'istanza degli oggetti MovieClip, Button o TextField. Per impostazione predefinita, questa proprietà è di tipo "undefined".

Se la proprietà `tabEnabled` risulta `undefined` o `true`, l'oggetto a cui essa è associata viene incluso nell'ordinamento di tabulazione automatico. L'oggetto viene incluso nell'ordinamento di tabulazione personalizzato se anche la proprietà `tabIndex` è impostata su un valore. Se `tabEnabled` è `false`, l'oggetto non viene incluso nell'ordinamento di tabulazione automatico. Nel caso di un clip filmato, se `tabEnabled` è `false`, gli elementi secondari del clip filmato possono essere ancora inclusi nell'ordinamento di tabulazione automatico, finché anche la proprietà `tabChildren` è impostata su `false`.

Se `tabEnabled` è `undefined` o `true` e viene impostata la proprietà `tabIndex`, l'oggetto è incluso nell'ordinamento di tabulazione personalizzato. Se `tabEnabled` è `false`, l'oggetto non viene incluso nell'ordinamento di tabulazione personalizzato, anche se viene impostata la proprietà `tabIndex`. Se `tabEnabled` è impostata su `false` in un clip filmato, gli elementi secondari del filmato possono ancora essere inclusi nell'ordinamento di tabulazione automatico.

Vedere anche

`Button.tabIndex`

Button.tabIndex

Disponibilità

Flash Player 6.

Uso

myButton.tabIndex

Descrizione

Proprietà; consente di personalizzare l'ordinamento di tabulazione degli oggetti in un filmato. È possibile impostare la proprietà `tabIndex` su un pulsante, clip filmato o istanza del campo di testo; per impostazione predefinita, tale proprietà è `undefined`.

Se qualsiasi oggetto attualmente visualizzato nel filmato Flash contiene una proprietà `tabIndex`, l'ordinamento di tabulazione automatico viene disattivato e l'ordinamento di tabulazione viene calcolato in base alle proprietà `tabIndex` degli oggetti del filmato. L'ordinamento di tabulazione personalizzato riguarda solo gli oggetti che hanno proprietà `tabIndex`.

La proprietà `tabIndex` può corrispondere a un numero intero non negativo. Gli oggetti vengono ordinati in base alle relative proprietà `tabIndex` in ordine ascendente. Un oggetto `tabIndex 1` precede un oggetto `tabIndex 2`. Se i due oggetti hanno lo stesso `tabIndex`, il primo nell'ordinamento di tabulazione è `undefined`.

L'ordinamento di tabulazione personalizzato definito dalla proprietà `tabIndex` è *flat*. Questo significa che non vengono considerate le relazioni gerarchiche degli oggetti nel filmato Flash. Tutti gli oggetti nel filmato Flash con proprietà `tabIndex` vengono inseriti nell'ordine di tabulazione, stabilito in base all'ordine dei valori `tabIndex`. Se due oggetti hanno lo stesso valore `tabIndex`, il primo è `undefined`. Non usare lo stesso valore `tabIndex` per più oggetti.

Button._target

Disponibilità

Flash Player 6.

Uso

`myButton._target`

Descrizione

Proprietà (sola lettura); restituisce il percorso target dell'istanza di pulsante specificata nel parametro *pulsante*.

Button.trackAsMenu

Disponibilità

Flash Player 6.

Uso

`myButton.trackAsMenu`

Descrizione

Proprietà; valore booleano che indica se altri pulsanti o clip filmato possono o meno ricevere eventi associati al rilascio del pulsante del mouse. Questa proprietà consente di creare dei menu. È possibile attribuire la proprietà `trackAsMenu` a qualsiasi pulsante e oggetto di clip filmato. Se la proprietà `trackAsMenu` non esiste, il comportamento predefinito è `false`.

È possibile modificare la proprietà `trackAsMenu` in qualsiasi momento, il pulsante modificato assume immediatamente il nuovo comportamento.

Vedere anche

`MovieClip.trackAsMenu`

Button._url

Disponibilità

Flash Player 6.

Uso

`myButton._url`

Descrizione

Proprietà (sola lettura); recupera l'URL del file SWF che ha creato il pulsante.

Button.useHandCursor

Disponibilità

Flash Player 6.

Uso

`myButton.useHandCursor`

Descrizione

Proprietà; un valore booleano che, se impostato su `true`, indica se viene visualizzato il cursore Mano quando si trascina il puntatore su un pulsante. Il valore predefinito di `useHandCursor` è `true`. Se la proprietà `useHandCursor` è impostata su `false`, viene usato il cursore Freccia.

È possibile modificare la proprietà `useHandCursor` in qualsiasi momento, il pulsante modificato assume immediatamente il comportamento del nuovo cursore. La proprietà `useHandCursor` è deducibile da un oggetto prototipo.

Button._visible

Disponibilità

Flash Player 6.

Uso

`myButton._visible`

Descrizione

Proprietà; un valore booleano che indica se il pulsante specificato dal parametro *Pulsante* è visibile. I pulsanti non visibili (la proprietà `_visible` è impostata su `false`) sono disabilitati.

Button._width

Disponibilità

Flash Player 6.

Uso

`myButton._width`

Descrizione

Proprietà; imposta e recupera la larghezza del pulsante, in pixel.

Esempio

Nell'esempio seguente, vengono impostate le proprietà di altezza e larghezza di un pulsante.

```
myButton._width=200;  
myButton._height=200;
```

Vedere anche

`MovieClip._width`

Button._x

Disponibilità

Flash Player 6.

Uso

`myButton._x`

Descrizione

Proprietà; un numero intero che imposta la coordinata x di un pulsante rispetto alle coordinate del clip filmato principale. Se un clip filmato è nella linea temporale principale, il sistema di coordinate corrispondente considera l'angolo superiore sinistro dello stage come $(0, 0)$. Se il pulsante è all'interno di un clip filmato a cui sono state applicate trasformazioni, il pulsante rientra nel sistema di coordinate locali del clip che lo contiene. Nel caso di un clip filmato ruotato di 90° in senso antiorario, il pulsante incluso eredita un sistema di coordinate con la stessa rotazione. Le coordinate del pulsante si riferiscono alla posizione del punto di registrazione.

Vedere anche

`Button._xscale`, `Button._y`, `Button._yscale`

Button._xmouse

Disponibilità

Flash Player 6.

Uso

`myButton._xmouse`

Descrizione

Proprietà (sola lettura); restituisce la coordinata x della posizione del mouse relativa al pulsante.

Vedere anche

`Button._ymouse`

Button._xscale

Disponibilità

Flash Player 6.

Uso

`myButton._xscale`

Descrizione

Proprietà; determina la scala orizzontale (*percentuale*) del pulsante applicata dal punto di registrazione del pulsante. Il punto di registrazione predefinito è $(0,0)$.

Il ridimensionamento in scala del sistema di coordinate locale modifica le impostazioni delle proprietà `_x` e `_y`, che sono definite in pixel. Se, ad esempio, il clip filmato principale è ridimensionato in scala al 50%, l'impostazione della proprietà `_x` sposta un oggetto nel pulsante per la metà del numero di pixel del filmato al 100%.

Vedere anche

`Button._x`, `Button._y`, `Button._yscale`

Button._y

Disponibilità

Flash Player 6.

Uso

`myButton._y`

Descrizione

Proprietà; imposta la coordinata *y* del pulsante rispetto alle coordinate locali del clip filmato principale. Se un pulsante è nella linea temporale principale, il sistema di coordinate considera l'angolo superiore sinistro dello stage come (0, 0). Se il pulsante è all'interno di un altro clip filmato a cui sono state applicate trasformazioni, il pulsante rientra nel sistema di coordinate locali del filmato che lo contiene. Nel caso di un clip filmato ruotato di 90° in senso antiorario, il pulsante incluso eredita un sistema di coordinate con la stessa rotazione. Le coordinate del pulsante si riferiscono alla posizione del punto di registrazione.

Vedere anche

`Button._x`, `Button._xscale`, `Button._yscale`

Button._ymouse

Disponibilità

Flash Player 6.

Uso

`myButton._ymouse`

Descrizione

Proprietà (sola lettura); indica la coordinata *y* della posizione del mouse relativa la pulsante.

Vedere anche

`Button._xmouse`

Button._yscale

Disponibilità

Flash Player 6.

Uso

`myButton._yscale`

Descrizione

Proprietà; determina la scala orizzontale (*percentuale*) del pulsante applicata dal suo punto di registrazione. Il punto di registrazione predefinito è (0,0).

Vedere anche

`Button._y`, `Button._x`, `Button._xscale`

call

Disponibilità

Flash Player 4. Questa azione è diventata obsoleta in Flash 5 ed è consigliato l'uso della nuova azione `function`.

Uso

`call(fotogramma)`

Parametri

fotogramma L'etichetta o il numero di un fotogramma nella linea temporale.

Valori restituiti

Nessuno.

Descrizione

Azione; esegue lo script nel fotogramma richiamato senza spostare l'indicatore di riproduzione su di esso. Le variabili locali non esisteranno più al termine dell'esecuzione dello script.

Vedere anche

`function`

call (funzione)

Disponibilità

Flash Player 6

Uso

`object.function([parametri])`

Parametri

oggetto Un oggetto (ad esempio un clip filmato) in cui è stata definita la funzione.

funzione Un identificatore che specifica una funzione definita dall'utente.

parametri Un parametro opzionale che indica tutti i parametri richiesti dalla funzione.

Valori restituiti

Nessuno.

Descrizione

Azione; consente di usare i campi dei parametri per richiamare una funzione definita dall'utente in Modalità normale nel pannello Azioni.

case

Disponibilità

Flash Player 4.

Uso

`case espressione: istruzioni`

Parametri

espressione Un'espressione qualsiasi.

istruzioni Un'istruzione qualsiasi.

Valori restituiti

Nessuno.

Descrizione

Parola chiave; definisce una condizione per l'azione `switch`. Le istruzioni contenute nel parametro *istruzioni* vengono eseguite se il parametro *espressione*, che segue la parola chiave `case`, equivale al parametro *espressione* dell'azione `switch` in base al criterio di uguaglianza rigorosa (`===`).

Se si usa l'azione `case` fuori da un'istruzione `switch`, genera un errore e lo script non viene compilato.

Vedere anche

`switch`, `default`, `break`, `===` (uguaglianza rigorosa)

chr

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta in Flash 5 ed è consigliato l'uso del nuovo metodo `String.fromCharCode`.

Uso

```
chr(numero)
```

Parametri

numero Un numero di codice ASCII.

Valori restituiti

Nessuno.

Descrizione

Funzione stringa; converte i numeri di codice ASCII in caratteri.

Esempio

Nell'esempio seguente, il numero 65 viene convertito nella lettera *A* e assegnato alla variabile `myVar`.

```
myVar = chr(65);
```

Vedere anche

`String.fromCharCode`

clearInterval

Disponibilità

Flash Player 6.

Uso

```
clearInterval( IDintervallo )
```

Parametri

IDintervallo Un oggetto restituito dall'attivazione della funzione `setInterval`.

Valori restituiti

Nessuno.

Descrizione

Azione; annulla l'attivazione della funzione `setInterval`.

Esempio

Nell'esempio seguente, l'intervallo viene inizialmente richiamato, quindi annullato:

```
function callback() {
    trace("richiamato intervallo");
}
var intervalID;
intervalID = setInterval( callback, 1000 );

// Successivamente
clearInterval( intervalID );
```

Vedere anche

`setInterval`

Color (oggetto)

L'oggetto `Color` consente di impostare il valore dei colori RGB e la trasformazione del colore dei clip filmato e di recuperare i valori dopo l'impostazione.

Prima di richiamare i metodi dell'oggetto `Color`, è necessario creare un'istanza usando la funzione di costruzione `new Color()`.

L'oggetto `Color` è supportato da Flash 5 Player e versioni successive.

Riepilogo dei metodi validi per l'oggetto Color

Metodo	Descrizione
<code>Color.getRGB</code>	Restituisce il valore RGB numerico impostato dall'ultima attivazione di <code>setRGB</code> .
<code>Color.getTransform</code>	Restituisce i dati relativi alla trasformazione impostati dall'ultima attivazione di <code>setTransform</code> .
<code>Color.setRGB</code>	Imposta la rappresentazione in formato esadecimale del valore RGB per l'oggetto <code>Color</code> .
<code>Color.setTransform</code>	Imposta la trasformazione dei colori per l'oggetto <code>Color</code> .

Funzione di costruzione per l'oggetto Color

Disponibilità

Flash Player 5.

Uso

```
new Color(target)
```

Parametri

target Il nome di istanza di un clip filmato.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un'istanza dell'oggetto `Color` per il clip filmato specificato dal parametro *target*. È possibile usare i metodi dell'oggetto `Color` per modificare il colore dell'intero clip filmato.

Esempio

Nell'esempio seguente, viene creata un'istanza dell'oggetto `Color` (`myColor`) per il clip filmato `myMovieClip` e impostato il valore RGB:

```
myColor = new Color(myMovieClip);  
myColor.setRGB(0xff9933);
```

Color.getRGB

Disponibilità

Flash Player 5.

Uso

```
myColor.getRGB();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce i valori numerici impostati dall'ultima attivazione di `setRGB`.

Esempio

Il codice seguente recupera il valore RGB per l'istanza dell'oggetto `Color` `myColor`, che viene convertito in una stringa esadecimale e assegnato alla variabile `value`.

```
value = myColor.getRGB().toString(16);
```

Vedere anche

`Color.setRGB`

Color.getTransform

Disponibilità

Flash Player 5.

Uso

```
myColor.getTransform()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce i valori di trasformazione impostati dall'ultima attivazione di `setTransform`.

Vedere anche

`Color.setTransform`

Color.setRGB

Disponibilità

Flash Player 5.

Uso

```
myColor.setRGB(0xRRGGBB)
```

Parametri

0xRRGGBB Il colore esadecimale o RGB da impostare. *RR*, *GG* e *BB* sono costituiti ciascuno da due cifre esadecimali che specificano l'offset di ciascun componente del colore. *0x* comunica al compilatore di ActionScript che il numero è un valore esadecimale.

Descrizione

Metodo; specifica un colore RGB per un'istanza dell'oggetto Color. Richiamando questo metodo si ridefiniscono tutte le precedenti impostazioni definite dal metodo `setTransform`.

Valori restituiti

Nessuno.

Esempio

Questo esempio imposta il valore del colore RGB per il clip filmato `myMovie`. Per vedere il funzionamento di questo codice, posizionare un clip filmato sullo stage con il nome dell'istanza `myMovie`. Quindi posizionare il codice riportato di seguito sul Fotogramma 1 nella linea temporale principale e scegliere Controlli > Prova filmato.

```
myColor = new Color(myMovie);  
myColor.setRGB(0x993366);
```

Vedere anche

`Color.setTransform`

Color.setTransform

Disponibilità

Flash Player 5.

Uso

```
myColor.setTransform(oggettoTrasformazioneColore);
```

Parametri

oggettoTrasformazioneColore Un oggetto creato con la funzione di creazione `new Object`. L'istanza dell'oggetto `Object` deve avere le seguenti proprietà che specificano i valori di trasformazione: *ra*, *rb*, *ga*, *gb*, *ba*, *bb*, *aa*, *ab*. Tali proprietà vengono descritte di seguito.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta le informazioni di trasformazione del colore per un'istanza dell'oggetto Color. Il parametro *oggettoTrasformazioneColore* si riferisce a un oggetto generico creato mediante la funzione di creazione `new Object`. Dispone di parametri che specificano la percentuale e i valori di offset dei componenti rosso, verde, blu e alpha (trasparenza) di un colore, immessa nel formato *0xRRGGBBAA*.

I parametri per un oggetto di trasformazione del colore corrispondono alle impostazioni visualizzate nella finestra di dialogo Effetto avanzato e sono definite come indicato di seguito:

- *ra* è la percentuale del componente rosso (da -100 a 100).
- *rb* è l'offset del componente rosso (da -255 a 255).
- *ga* è la percentuale del componente verde (da -100 a 100).
- *gb* è l'offset del componente verde (da -255 a 255).
- *ba* è la percentuale del componente blu (da -100 a 100).
- *bb* è l'offset del componente blu (da -255 a 255).
- *aa* è la percentuale di alfa (da -100 a 100).
- *ab* è l'offset di alfa (da -255 a 255).

Per creare un parametro *oggettoTrasformazioneColore*, attenersi alla procedura indicata di seguito:

```
myColorTransform = new Object();
myColorTransform.ra = 50;
myColorTransform.rb = 244;
myColorTransform.ga = 40;
myColorTransform.gb = 112;
myColorTransform.ba = 12;
myColorTransform.bb = 90;
myColorTransform.aa = 40;
myColorTransform.ab = 70;
```

È inoltre possibile usare la seguente sintassi per creare un parametro

oggettoTrasformazioneColore:

```
myColorTransform = { ra: '50', rb: '244', ga: '40', gb: '112', ba: '12', bb:
'90', aa: '40', ab: '70' }
```

Esempio

Nell'esempio seguente, viene creata una nuova istanza dell'oggetto Color per un filmato target e un oggetto generico, detto *myColorTransform*, avente le proprietà definite in precedenza.

Mediante il metodo *setTransform* si invia l'*oggettoTrasformazioneColore* a un oggetto Color. Per usare questo codice in un documento Flash (FLA), posizionarlo sul Fotogramma 1 della linea temporale principale e posizionare il clip filmato sullo stage con il nome di istanza *myMovie*, come esemplificato nel codice riportato di seguito:

```
// Crea un oggetto Color chiamato myColor per il target myMovie
myColor = new Color(myMovie);
// Crea un oggetto di trasformazione del colore chiamato myColorTransform
// usando l'oggetto Object generico
myColorTransform = new Object();
// Imposta i valori per myColorTransform
myColorTransform = { ra: '50', rb: '244', ga: '40', gb: '112', ba: '12', bb:
'90', aa: '40', ab: '70'};
// Associa l'oggetto di trasformazione del colore all'oggetto Color
// creato per myMovie
myColor.setTransform(myColorTransform);
```

continue

Disponibilità

Flash Player 4.

Uso

`continue`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; appare in vari tipi di istruzioni cicliche. Il comportamento cambia a seconda del tipo di ciclo.

In un ciclo `while`, l'azione `continue` indica all'applicazione Flash di ignorare il resto del corpo del ciclo e di passare all'inizio del ciclo, dove la condizione viene verificata.

In un ciclo `do...while`, l'azione `continue` indica all'applicazione Flash di ignorare il resto del corpo del ciclo e di passare alla fine del ciclo, dove la condizione viene verificata.

In un ciclo `for`, l'azione `continue` indica all'applicazione Flash di ignorare il resto del corpo del ciclo e di passare alla valutazione dell'espressione finale del ciclo `for`.

In un ciclo `for...in`, l'azione `continue` indica all'applicazione Flash di ignorare il resto del corpo del ciclo e di tornare all'inizio del ciclo, dove viene elaborato il valore successivo nell'enumerazione.

Vedere anche

`do while`, `for`, `for...in`, `while`

CustomActions (oggetto)

I metodi dell'oggetto `CustomActions` consentono a un filmato Flash riprodotto dallo strumento di creazione di codice di Flash di controllare tutte le azioni personalizzate registrate con tale strumento. Un filmato Flash può installare e disinstallare azioni personalizzate, recuperare la definizione XML di un'azione personalizzata e recuperare un elenco delle azioni personalizzate registrate.

È possibile usare questi metodi per creare filmati Flash che siano estensioni dello strumento di creazione di codice di Flash. Un filmato di estensione di questo tipo potrebbe, ad esempio, usare Flash Application Protocol per navigare all'interno di un archivio UDDI e scaricare i servizi Web nella casella degli strumenti Azioni.

Riepilogo dei metodi validi per l'oggetto CustomActions

Metodo	Descrizione
<code>CustomActions.get</code>	Legge i contenuti di un file di definizione XML delle azioni personalizzate.
<code>CustomActions.install</code>	Installa un nuovo file di definizione XML delle azioni personalizzate.
<code>CustomActions.list</code>	Restituisce un elenco di tutte le azioni personalizzate registrate.
<code>CustomActions.uninstall</code>	Rimuove un file di definizione XML delle azioni personalizzate.

CustomActions.get

Disponibilità

Flash Player 6.

Uso

```
CustomActions.get(nomeAzioniPersonalizzate)
```

Parametri

nomeAzioniPersonalizzate Il nome della definizione delle azioni personalizzate da recuperare.

Valori restituiti

Nessuno.

Descrizione

Metodo; legge i contenuti del file della definizione XML delle azioni personalizzate denominato *nomeAzioniPersonalizzate*.

Il nome del file della definizione deve essere semplice, senza estensione .xml e senza separatori di directory (':', '/' oppure '\').

Se non è possibile trovare il file della definizione specificato da *nomeAzioniPersonalizzate*, viene restituito un valore `undefined`. Se è stata individuata la definizione XML delle azioni personalizzate specificata da *nomeAzioniPersonalizzate*, essa viene letta interamente e restituita come stringa.

CustomActions.install

Disponibilità

Flash Player 6.

Uso

```
CustomActions.install(nomeAzioniPersonalizzate, definizioneXMLPersonalizzata)
```

Parametri

nomeAzioniPersonalizzate Il nome della definizione delle azioni personalizzate da installare.

definizioneXMLPersonalizzata Il testo della definizione XML da installare.

Valori restituiti

Nessuno.

Descrizione

Metodo; installa un nuovo file di definizione XML delle azioni personalizzate indicate dal parametro *nomeAzioniPersonalizzate*. I contenuti del file vengono specificati dalla stringa *definizioneXMLPersonalizzata*.

Il nome del file della definizione deve essere semplice, senza estensione .xml e senza separatori di directory (':', '/' oppure '\').

Se esiste già un file delle azioni personalizzate denominato *nomeAzioniPersonalizzate*, esso viene sovrascritto.

Se durante l'installazione si verifica un errore, viene restituito un valore `false`; in caso contrario viene restituito un valore `true` a indicare che l'installazione dell'azione personalizzata ha avuto esito positivo.

Se la directory `Configuration/ActionsPanel/CustomActions` non esiste quando viene invocato questo metodo, viene creata.

CustomActions.list

Disponibilità

Flash Player 6.

Uso

```
CustomActions.list()
```

Parametri

Nessuno.

Valori restituiti

Una matrice.

Descrizione

Metodo; restituisce un oggetto Array contenente i nomi di tutte le azioni personalizzate registrate mediante lo strumento di creazione di codice. Gli elementi della matrice sono nomi semplici, senza estensione `.xml` e separatori di directory (ad esempio, `:"`, `/'` oppure `\"`). Se non sono presenti azioni personalizzate registrate, il metodo `list` restituisce una matrice di lunghezza zero. Se si verifica un errore, il metodo `list` restituisce il valore `undefined`.

CustomActions.uninstall

Disponibilità

Flash Player 6.

Uso

```
CustomActions.uninstall(nomeAzioniPersonalizzate)
```

Parametri

nomeAzioniPersonalizzate Il nome della definizione delle azioni personalizzate da disinstallare.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove la definizione XML delle azioni personalizzate denominata *nomeAzioniPersonalizzate*.

Il nome del file della definizione deve essere semplice, senza estensione `.xml` e senza separatori di directory (`:'`, `/'` oppure `\"`).

Se non vengono rilevate azioni personalizzate, viene restituito un valore `false`. Se le azioni personalizzate sono state rimosse correttamente, viene restituito un valore `true`.

Date (oggetto)

L'oggetto Date consente di recuperare i valori relativi all'ora e alla data standard (ora di Greenwich, attualmente denominata tempo universale coordinato UTC) su cui è in esecuzione Flash Player. I metodi dell'oggetto Date non sono statici, ma sono validi solo per l'istanza dell'oggetto Date specificata quando viene richiamato il metodo. Il metodo `Date.UTC` costituisce un'eccezione; si tratta infatti di un metodo statico.

L'oggetto Date gestisce l'ora legale a seconda del sistema operativo e della versione di Flash Player. Nei sistemi operativi riportati di seguito, Flash Player 6 gestisce l'ora legale nei seguenti modi:

- Windows: l'oggetto Date è impostato per visualizzare automaticamente l'ora legale. Questo oggetto rileva se nelle impostazioni internazionali correnti è applicata l'ora legale e individua gli orari e le date di transizione. Tali date vengono applicate anche al passato e al futuro; pertanto, è possibile che l'ora legale non venga calcolata correttamente per le date trascorse, poiché le date di transizione erano diverse.
- Mac OS 8 e 9: l'oggetto Date usa l'ora legale corrente, indipendentemente dalla data o dall'ora calcolata. Ad esempio, negli Stati Uniti (ora costa del Pacifico), nel mese di agosto quando è in vigore l'ora legale (DST), un oggetto Date contenente la data del 1° gennaio 2001 conserva l'ora legale anche se essa non è in vigore in gennaio. Questo problema non è risolvibile in Mac OS 8 o 9, poiché in tali sistemi non è disponibile un database dei fusi orari.
- Mac OS X: l'oggetto Date è impostato per visualizzare automaticamente l'ora legale. Il database delle informazioni in Mac OS X viene usato per determinare se è necessario applicare l'ora legale a una qualsiasi ora e data passate e future.

Nei sistemi operativi riportati di seguito, Flash Player 5 gestisce l'ora legale nei seguenti modi:

- Mac OS 8 e 9: il funzionamento è identico a quello descritto per Flash Player 6.
- Windows: vengono sempre applicati i criteri USA per l'ora legale. Questo comporta transizioni non corrette per gli altri paesi in cui è in vigore l'ora legale con altre date di transizione. L'applicazione Flash è in grado di rilevare correttamente se nelle impostazioni internazionali correnti è applicata l'ora legale.

Prima di richiamare i metodi dell'oggetto Date, è necessario creare un'istanza dell'oggetto Date usando la funzione di costruzione corrispondente.

Per l'oggetto Date, è richiesto Flash Player 5.

Riepilogo dei metodi validi per l'oggetto Date

Metodo	Descrizione
Date.getDate	Restituisce il giorno del mese in base all'ora locale.
Date.getDay	Restituisce il giorno della settimana in base all'ora locale.
Date.getFullYear	Restituisce l'indicazione dell'anno a quattro cifre in base all'ora locale.
Date.getHours	Restituisce l'ora in base all'ora locale.
Date.getMilliseconds	Restituisce i millisecondi in base all'ora locale.
Date.getMinutes	Restituisce i minuti in base all'ora locale.
Date.getMonth	Restituisce il mese in base all'ora locale.
Date.getSeconds	Restituisce i secondi in base all'ora locale.
Date.getTime	Restituisce il numero di millisecondi dal 1° gennaio 1970, tempo universale coordinato UTC.
Date.getTimezoneOffset	Restituisce la differenza, in minuti, tra l'ora locale del computer e il tempo universale coordinato UTC.
Date.getUTCDate	Restituisce il giorno del mese (la data) in base al tempo universale coordinato UTC.
Date.getUTCDay	Restituisce il giorno della settimana in base al tempo universale coordinato UTC.
Date.getUTCFullYear	Restituisce l'indicazione dell'anno a quattro cifre in base al tempo universale coordinato UTC.
Date.getUTCHours	Restituisce l'ora in base al tempo universale coordinato UTC.
Date.getUTCMilliseconds	Restituisce i millisecondi in base al tempo universale coordinato UTC.
Date.getUTCMinutes	Restituisce i minuti in base al tempo universale coordinato UTC.
Date.getUTCMonth	Restituisce il mese in base al tempo universale coordinato UTC.
Date.getUTCSeconds	Restituisce i secondi in base al tempo universale coordinato UTC.
Date.getYear	Restituisce l'anno in base all'ora locale.
Date.setDate	Imposta il giorno del mese in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setFullYear	Imposta l'anno di quattro cifre in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setHours	Imposta l'ora in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setMilliseconds	Imposta i millisecondi in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setMinutes	Imposta i minuti in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setMonth	Imposta il mese in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setSeconds	Imposta i secondi in base all'ora locale. Restituisce la nuova data in millisecondi.
Date.setTime	Imposta la data in millisecondi. Restituisce la nuova data in millisecondi.
Date.setUTCDate	Imposta la data in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
Date.setUTCFullYear	Imposta l'anno in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
Date.setUTCHours	Imposta l'ora in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.

Metodo	Descrizione
<code>Date.setUTCMilliseconds</code>	Imposta i millisecondi in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
<code>Date.setUTCMinutes</code>	Imposta i minuti in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
<code>Date.setUTCMonth</code>	Imposta il mese in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
<code>Date.setUTCSeconds</code>	Imposta i secondi in base al tempo universale coordinato UTC. Restituisce la nuova data in millisecondi.
<code>Date.setYear</code>	Imposta l'anno in base all'ora locale.
<code>Date.toString</code>	Restituisce un valore stringa che rappresenta la data e l'ora memorizzate nell'oggetto <code>Date</code> specificato.
<code>Date.UTC</code>	Restituisce il numero di millisecondi tra la mezzanotte del 1 gennaio 1970, tempo universale coordinato UTC, e l'ora specificata.

Funzione di costruzione per l'oggetto `Date`

Disponibilità

Flash Player 5.

Uso

```
new Date()
```

```
new Date(anno [, mese [, data [, ora [, minuti [, secondi [, millisecondi ]]]]]]) )
```

Parametri

anno Valore da 0 a 99 indicante un anno dal 1900 al 1999. Per l'indicazione di altri anni, usare il formato esteso a quattro cifre.

mese Un numero intero da 0 (gennaio) a 11 (dicembre).

data Un numero intero da 1 a 31. Questo parametro è opzionale.

ora Un numero intero da 0 a 23.

minuti Un numero intero da 0 a 59. Questo parametro è opzionale.

secondi Un numero intero da 0 a 59. Questo parametro è opzionale.

millisecondi Un numero intero da 0 a 999. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Oggetto; costruisce un nuovo oggetto `Date` che indica la data e l'ora attuali o un'altra data specificata.

Esempio

L'esempio seguente recupera la data e l'ora correnti.

```
now = new Date();
```

Nell'esempio seguente, viene creato un nuovo oggetto `Date` per il compleanno di Gary (7 agosto 1974).

```
gary_birthday = new Date (7, 7, 74);
```

Nell'esempio seguente, viene creato un nuovo oggetto `Date`, si concatenano i valori restituiti dall'esecuzione dei metodi dell'oggetto `Date` `getMonth`, `getDate` e `getFullYear` che vengono visualizzati nel campo di testo specificato dalla variabile `dateTextField`.

```
myDate = new Date();  
dateTextField = ((myDate.getMonth() + 1) + "/" + myDate.getDate() + "/" +  
    myDate.getFullYear());
```

Date.getDate

Disponibilità

Flash Player 5.

Uso

```
myDate.getDate()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il giorno del mese (un intero da 1 a 31) dell'oggetto `Date` specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getDay

Disponibilità

Flash Player 5.

Uso

```
myDate.getDay()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il giorno della settimana (0 corrisponde a domenica, 1 a lunedì e così via) dell'oggetto `Date` specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getFullYear

Disponibilità

Flash Player 5.

Uso

```
myDate.getFullYear()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'anno per esteso (un numero in formato a quattro cifre, ad esempio 2000) dell'oggetto Date specificato, in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Esempio

L'esempio seguente usa la funzione di costruzione per creare un nuovo oggetto Date e invia il valore restituito dal metodo `getFullYear` alla finestra Output.

```
myDate = new Date();  
trace(myDate.getFullYear());
```

Date.getHours

Disponibilità

Flash Player 5.

Uso

```
myDate.getHours()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'ora (un intero da 0 a 23) dell'oggetto Date specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getMilliseconds

Disponibilità

Flash Player 5.

Uso

```
myDate.getMilliseconds()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i millisecondi (un intero da 0 a 999) dell'oggetto Date specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getMinutes

Disponibilità

Flash Player 5.

Uso

```
myDate.getMinutes()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i minuti (un intero da 0 a 59) dell'oggetto Date specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getMonth

Disponibilità

Flash Player 5.

Uso

```
myDate.getMonth()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il mese (0 corrisponde a gennaio, 1 a febbraio e così via) dell'oggetto Date specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getSeconds

Disponibilità

Flash Player 5.

Uso

```
myDate.getSeconds()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i secondi (un intero da 0 a 59) dell'oggetto Date specificato in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.getTime

Disponibilità

Flash Player 5.

Uso

```
myDate.getTime()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di millisecondi dalla mezzanotte del 1° gennaio 1970, tempo universale coordinato UTC, per l'oggetto Date specificato. Questo metodo consente di rappresentare l'ora di un determinato istante confrontando due o più oggetti Date.

Date.getTimezoneOffset

Disponibilità

Flash Player 5.

Uso

```
myDate.getTimezoneOffset()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i minuti di differenza tra l'ora locale del computer e il tempo universale coordinato UTC.

Esempio

Nell'esempio seguente, viene visualizzata la differenza tra l'ora legale locale di San Francisco e il tempo universale coordinato UTC. L'ora legale viene considerata nel valore restituito solo se la data specificata nell'oggetto Date ricade nel periodo in cui viene applicata l'ora legale.

```
trace(new Date().getTimezoneOffset());  
// 420 viene visualizzato nella finestra Output  
// (7 ore * 60 minuti/ore = 420 minuti)  
// Questo esempio indica l'ora legale per la costa del Pacifico  
// (PDT, GMT-0700).  
// Il risultato varia a seconda della località e del periodo dell'anno.
```

Date.getUTCDate

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCDate()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il giorno del mese (data) dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCDay

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCDate()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il giorno della settimana dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCFullYear

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCFullYear()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'anno in formato a quattro cifre dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCHours

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCHours()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'ora dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCMilliseconds

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCMilliseconds()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i millisecondi dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCMinutes

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCMinutes()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i minuti dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCMonth

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCMonth()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il mese dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getUTCSeconds

Disponibilità

Flash Player 5.

Uso

```
myDate.getUTCSeconds()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce i secondi dell'oggetto Date specificato in base al tempo universale coordinato UTC.

Date.getYear

Disponibilità

Flash Player 5.

Uso

```
myDate.getYear()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'anno dell'oggetto Date specificato in base al tempo locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player. All'anno viene sottratta la cifra 1900. Ad esempio, l'anno 2000 è rappresentato da 100.

Date.setDate

Disponibilità

Flash Player 5.

Uso

```
myDate.setDate(data)
```

Parametri

data Un numero intero da 1 a 31.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta il giorno del mese per l'oggetto Date specificato in base all'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setFullYear

Disponibilità

Flash Player 5.

Uso

```
myDate.setFullYear(anno [, mese [, data]] )
```

Parametri

anno Un numero nel formato a quattro cifre che specifica l'anno. I numeri a due cifre non rappresentano gli anni. Ad esempio, 99 non rappresenta l'anno 1999, ma l'anno 99.

mese Un numero intero da 0 (gennaio) a 11 (dicembre). Questo parametro è opzionale.

data Un numero intero da 1 a 31. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta l'anno dell'oggetto Date specificato in base all'ora locale e restituisce la nuova data in millisecondi. Se i parametri *mese* e *data* sono specificati, vengono impostati anch'essi in base all'ora locale. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Richiamando questo metodo non si modificano gli altri campi dell'oggetto Date specificato. Tuttavia, se in seguito alla attivazione di questo metodo cambia il giorno della settimana, è possibile che i metodi `getUTCDay` e `getDay` restituiscano un nuovo valore.

Date.setHours

Disponibilità

Flash Player 5.

Uso

myDate.setHours(ora)

Parametri

ora Un numero intero da 0 a 23.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta le ore per l'oggetto Date specificato in base all'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setMilliseconds

Disponibilità

Flash Player 5.

Uso

myDate.setMilliseconds(millisecondi)

Parametri

millisecondi Un numero intero da 0 a 999.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i millisecondi per l'oggetto Date specificato in base all'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setMinutes

Disponibilità

Flash Player 5.

Uso

myDate.setMinutes(minuti)

Parametri

minuti Un numero intero da 0 a 59.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i minuti per l'oggetto Date specificato in base all'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setMonth

Disponibilità

Flash Player 5.

Uso

```
myDate.setMonth(mese [, data ])
```

Parametri

mese Un numero intero da 0 (gennaio) a 11 (dicembre).

data Un numero intero da 1 a 31. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta il mese per l'oggetto Date specificato nell'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setSeconds

Disponibilità

Flash Player 5.

Uso

```
myDate.setSeconds(secondi)
```

Parametri

secondi Un numero intero da 0 a 59.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i secondi per l'oggetto Date specificato nell'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.setTime

Disponibilità

Flash Player 5.

Uso

```
myDate.setTime(millisecondi)
```

Parametri

millisecondi Un valore intero in cui a 0 corrisponde a 0:00 (ora di Greenwich) del 1° gennaio 1970.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta la data per l'oggetto Date specificato in millisecondi da mezzanotte del 1° gennaio 1970 e restituisce la nuova data in millisecondi.

Date.setUTCDate

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCDate(data)
```

Parametri

data Un numero intero da 1 a 31.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta la data per l'oggetto Date specificato in base all'ora universale e restituisce la nuova data in millisecondi. Richiamando questo metodo non si modificano gli altri campi dell'oggetto Date specificato, ma i metodi `getUTCDay` e `getDay` potrebbero indicare un nuovo valore per il giorno della settimana in seguito alla attivazione di questo metodo.

Date.setUTCFullYear

Disponibilità

Flash Player 5.

Uso

```
oggettoData.setUTCFullYear(anno [, mese [, data]])
```

Parametri

anno L'anno nel formato a quattro cifre, ad esempio 2000.

mese Un numero intero da 0 (gennaio) a 11 (dicembre). Questo parametro è opzionale.

data Un numero intero da 1 a 31. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta l'anno per l'oggetto Date specificato (*mydate*) nell'ora universale e restituisce la nuova data in millisecondi.

Facoltativamente, questo metodo può anche impostare il mese e la data rappresentati dall'oggetto Date specificato. Non viene modificato nessun altro campo dell'oggetto Date. Se in seguito all'attivazione del metodo `setUTCFullYear` cambia il giorno della settimana, è possibile che i metodi `getUTCDay` e `getDay` riportino un nuovo valore.

Date.setUTCHours

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCHours(ora [, minuti [, secondi [, millisecondi]])
```

Parametri

ora Un numero intero da 0 a 23.

minuti Un numero intero da 0 a 59. Questo parametro è opzionale.

secondi Un numero intero da 0 a 59. Questo parametro è opzionale.

millisecondi Un numero intero da 0 a 999. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta l'ora per l'oggetto Date specificato nell'ora universale e restituisce la nuova data in millisecondi.

Date.setUTCMilliseconds

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCMilliseconds(millisecondi)
```

Parametri

millisecondi Un numero intero da 0 a 999.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i millisecondi per l'oggetto Date specificato nell'ora universale e restituisce la nuova data in millisecondi.

Date.setUTCMinutes

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCMinutes(minuti [, secondi [, millisecondi]])
```

Parametri

minuti Un numero intero da 0 a 59.

secondi Un numero intero da 0 a 59. Questo parametro è opzionale.

millisecondi Un numero intero da 0 a 999. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i minuti per l'oggetto Date specificato nell'ora universale e restituisce la nuova data in millisecondi.

Date.setUTCMonth

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCMonth(mese [, data])
```

Parametri

mese Un numero intero da 0 (gennaio) a 11 (dicembre).

data Un numero intero da 1 a 31. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta il mese e (opzionalmente) il giorno (la data), per l'oggetto Date specificato nell'ora universale e restituisce la nuova data in millisecondi. Richiamando questo metodo non si modificano gli altri campi dell'oggetto Date specificato. Tuttavia, se quando si richiama il metodo `setUTCMonth` il giorno viene modificato specificando il parametro *data*, è possibile che i metodi `getUTCDay` e `getDay` riportino un nuovo valore.

Date.setUTCSeconds

Disponibilità

Flash Player 5.

Uso

```
myDate.setUTCSeconds(secondi [, millisecondi])
```

Parametri

secondi Un numero intero da 0 a 59.

millisecondi Un numero intero da 0 a 999. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta i secondi per l'oggetto Date specificato con l'ora universale e restituisce la nuova data in millisecondi.

Date.setYear

Disponibilità

Flash Player 5.

Uso

```
myDate.setYear(anno)
```

Parametri

anno Se *anno* è un numero intero compreso tra 0 e 99, `setYear` imposta l'anno su 1900 + *anno*; in caso contrario l'anno corrisponde al valore del parametro *anno*.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; imposta l'anno per l'oggetto `Date` specificato con l'ora locale e restituisce la nuova data in millisecondi. L'ora locale è determinata dal sistema operativo in cui si esegue Flash Player.

Date.toString

Disponibilità

Flash Player 5.

Uso

```
myDate.toString()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce un valore stringa per l'oggetto `Date` specificato in un formato leggibile e restituisce la nuova data in millisecondi.

Esempio

L'esempio seguente restituisce le informazioni contenute nell'oggetto `dateOfBirth` di tipo `Date` in formato stringa.

```
var dateOfBirth = new Date(74, 7, 7, 18, 15);  
trace (dateOfBirth.toString());
```

Output (ora solare Pacifico):

```
Mer 7 Ago 18:15:00 GMT-0700 1974
```


Date.UTC

Disponibilità

Flash Player 5.

Uso

`Date.UTC(anno, mese [, data [, ora [, minuti [, secondi [, millisecondi]]]])`

Parametri

anno Un numero nel formato a quattro cifre, ad esempio 2000.

mese Un numero intero da 0 (gennaio) a 11 (dicembre).

data Un numero intero da 1 a 31. Questo parametro è opzionale.

ora Un numero intero da 0 a 23.

minuti Un numero intero da 0 a 59. Questo parametro è opzionale.

secondi Un numero intero da 0 a 59. Questo parametro è opzionale.

millisecondi Un numero intero da 0 a 999. Questo parametro è opzionale.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di millisecondi tra la mezzanotte del 1° gennaio 1970 (ora universale) e l'ora specificata nei parametri. Si tratta di un metodo statico che viene richiamato attraverso la funzione di costruzione dell'oggetto `Date` e non attraverso un oggetto `Date` specifico. Questo metodo consente di creare un oggetto `Date` che usa l'ora universale, mentre la funzione di costruzione `Date` usa l'ora locale.

Esempio

L'esempio seguente crea un nuovo oggetto `Date` `gary_birthday` definito in base all'ora universale. Ecco la variazione dell'ora universale per l'esempio usato per la funzione di costruzione `new Date`.

```
gary_birthday = new Date(Date.UTC(1974, 7, 8));
```

default

Disponibilità

Flash Player 6.

Uso

`default: istruzioni`

Parametri

istruzioni Un'istruzione qualsiasi.

Valori restituiti

Nessuno.

Descrizione

Parola chiave; definisce il caso predefinito per un'azione `switch`. Le istruzioni vengono eseguite se il parametro *Espressione* dell'azione `switch` non equivale (applicando il criterio di uguaglianza rigorosa) a nessuno dei parametri *Espressione* che seguono le parole chiave `case` per una determinata azione `switch`.

Un'azione `switch` non richiede un caso di tipo `default`. Non è necessario specificare un caso di tipo `default` come ultima voce in elenco. L'uso di un'azione `default` non associata a un'azione `switch` genera un errore e lo script non viene compilato.

Esempio

Nell'esempio seguente, l'espressione `A` non equivale all'espressione `B` o `D`. Per questo motivo, la parola chiave viene eseguita e l'azione `trace` inviata alla finestra `Output`.

```
switch ( A ) {  
    case B:  
        C;  
        break;  
    case D:  
        E;  
        break;  
    default:  
        trace ("non è stato rilevato alcun caso specifico");  
}
```

Vedere anche

`switch`, `case`, `break`

delete

Disponibilità

Flash Player 5.

Uso

`delete` *referimento*

Parametri

referimento Il nome della variabile o dell'oggetto da eliminare.

Valori restituiti

Nessuno.

Descrizione

Operatore; elimina definitivamente l'oggetto o la variabile specificata dal parametro *referimento* e restituisce `true` se l'oggetto viene eliminato correttamente; in caso contrario, restituisce `false`. Questo operatore consente di liberare la memoria usata dagli script. L'operatore `delete` viene generalmente usato come istruzione, come nell'esempio seguente:

```
delete x;
```

L'operatore `delete` potrebbe non avere effetto e restituire il valore `false` se il parametro *referimento* non esiste oppure se non è possibile eliminarlo, come nel caso degli oggetti e delle proprietà predefinite, nonché delle variabili dichiarate con `var`. Non è possibile usare l'operatore `delete` per rimuovere clip filmato.

Esempio

Nell'esempio seguente, un oggetto viene creato, usato e, infine, eliminato quando non è più necessario.

```
account = new Object();
account.name = 'Lea';
account.balance = 10000;

delete account;
```

Esempio

L'esempio seguente elimina una proprietà di un oggetto:

```
// Crea il nuovo oggetto "account"
account = new Object();
// Assegna la proprietà name all'account
account.name = 'Lea';
// Cancella la proprietà
delete account.name;
```

Esempio

Il seguente è un altro esempio di eliminazione di una proprietà di un oggetto:

```
// Crea un oggetto Array con lunghezza 0
array = new Array();
// Array.length diventa 1
array[0] = "abc";
// Aggiunge un altro elemento alla matrice; .Array.length diventa 2
array[1] = "def";
// Aggiunge un altro elemento alla matrice; .Array.length diventa 3
array[2] = "ghi";
// array[2] viene eliminato, ma Array.length rimane invariata
delete array[2];
```

Nell'esempio seguente, viene mostrato il comportamento dell'operatore `delete` sui riferimenti agli oggetti.

```
// Creare un nuovo oggetto e assegnare la variabile ref1
// come riferimento all'oggetto
ref1 = new Object();
ref1.name = "Mara";
// Copiare la variabile di riferimento in una nuova variabile
// ed eliminare ref1
ref2 = ref1;
delete ref1;
```

Se la variabile `ref1` non fosse stata copiata in `ref2`, l'oggetto sarebbe stato eliminato insieme a `ref1` poiché non sarebbero più stati presenti riferimenti a esso. Eliminando `ref2`, non vi sarebbero più riferimenti all'oggetto; pertanto, verrebbe eliminato definitivamente e la memoria impiegata sarebbe nuovamente disponibile.

Vedere anche

`var`

do while

Disponibilità

Flash Player 4.

Uso

```
do {  
    istruzione/i  
} while (condizione)
```

Parametri

condizione La condizione da valutare.

istruzione/i L'istruzione o le istruzioni da eseguire fino a quando il parametro *condizione* restituisce true.

Valori restituiti

Nessuno.

Descrizione

Azione; esegue le istruzioni, quindi valuta la condizione in un ciclo, finché la condizione risulta true.

Vedere anche

break, continue

duplicateMovieClip

Disponibilità

Flash Player 4.

Uso

```
duplicateMovieClip(target, nuovonome, profondità)
```

Parametri

target Il percorso target del clip filmato da duplicare.

nuovonome Identificatore univoco per il clip filmato duplicato.

profondità Livello di profondità univoco per il clip filmato duplicato. Il livello di profondità corrisponde all'ordine di impilamento dei clip filmato duplicati, simile all'ordine dei livelli nella linea temporale; i clip filmato con un livello di profondità basso sono nascosti sotto quelli con un ordine di impilamento più elevato. È necessario assegnare a ciascun clip filmato duplicato un livello di profondità univoco per impedire che vada a sostituirsi a filmati presenti in livelli di profondità già occupati.

Valori restituiti

Nessuno.

Descrizione

Azione; crea un'istanza di clip filmato durante la riproduzione del filmato. Nei clip filmato, l'indicatore di riproduzione inizia sempre dal fotogramma 1, a prescindere dal punto in cui l'indicatore si trova nel clip filmato originale (o "principale"). Le variabili nel clip filmato principale non vengono copiate nel clip filmato duplicato. L'eliminazione del clip filmato principale comporta l'eliminazione del clip filmato duplicato. Per eliminare un'istanza di clip filmato creata con l'azione `duplicateMovieClip`, usare l'azione o metodo `removeMovieClip`.

Esempio

L'istruzione seguente duplica l'istanza di clip filmato `flower` dieci volte. La variabile `i` viene usata per creare un nuovo nome dell'istanza e una profondità univoca per ciascun clip filmato duplicato.

```
on (release) {
    amount = 10;
    while(amount>0) {
        duplicateMovieClip (_root.flower, "mc"+i, i);
        setProperty ("mc"+i, _x, random(275));
        setProperty ("mc"+i, _y, random(275));
        setProperty ("mc"+i, _alpha, random(275));
        setProperty ("mc"+i, _xscale, random(50));
        setProperty ("mc"+i, _yscale, random(50));
        i++;
        amount--;
    }
}
```

Vedere anche

`MovieClip.duplicateMovieClip`, `removeMovieClip`, `MovieClip.removeMovieClip`

else

Disponibilità

Flash Player 4.

Uso

Istruzione `else`
`else {...istruzione/i...}`

Parametri

condizione Espressione che restituisce `true` o `false`.

istruzioni Una serie di istruzioni alternative da eseguire se la condizione specificata nell'istruzione `if` è `false`.

Valori restituiti

Nessuno.

Descrizione

Azione; specifica le istruzioni da eseguire se la condizione nell'istruzione `if` restituisce `false`.

Vedere anche

`if`

else if

Disponibilità

Flash Player 4.

Uso

```
if (condizione){
    istruzione1/i;
} else if (condizione){
    istruzione1/i;
}
```

Parametri

condizione Espressione che restituisce true o false.

istruzioni Una serie di istruzioni alternative da eseguire se la condizione specificata nell'istruzione if è false.

Valori restituiti

Nessuno.

Descrizione

Azione; valuta una condizione e specifica le istruzioni da eseguire se la condizione nell'istruzione iniziale if restituisce false. Se la condizione else if è true, l'interprete Flash esegue le istruzioni che seguono tale condizione all'interno di parentesi graffe ({}). Se la condizione else if è false, Flash ignora le istruzioni all'interno delle parentesi graffe ed esegue le istruzioni che seguono le parentesi. Usare l'azione else if per creare codice condizionale negli script.

Esempio

L'esempio seguente usa azioni else if per verificare se ciascun lato di un oggetto si trova all'interno di un determinato limite:

```
// Se l'oggetto oltrepassa i limiti fissati,  
// inviarlo nuovamente e invertirne la velocità  
    if (this._x>rightBound) {  
        this._x = rightBound;  
        xInc = -xInc;  
    } else if (this._x<leftBound) {  
        this._x = leftBound;  
        xInc = -xInc;  
    } else if (this._y>bottomBound) {  
        this._y = bottomBound;  
        yInc = -yInc;  
    } else if (this._y<topBound) {  
        this._y = topBound;  
        yInc = -yInc;  
    }  
}
```

Vedere anche

if

#endinitclip

Disponibilità

Flash Player 6.

Uso

```
#endinitclip
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; indica la fine di un blocco di azioni di inizializzazione di un componente.

Esempio

```
#initclip  
...le azioni di inizializzazione del componente vanno inserite qui...  
#endinitclip
```

Vedere anche

```
#initclip
```

eq (uguale; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore `==` (uguaglianza).

Uso

```
espressione1 eq espressione2
```

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Valori restituiti

Nessuno.

Descrizione

Operatore di confronto; verifica l'uguaglianza di due espressioni e restituisce il valore `true` se la rappresentazione in formato stringa dell'*espressione1* è uguale a quella dell'*espressione2*; in caso contrario, il valore restituito dall'operatore è `false`.

Vedere anche

`==` (uguaglianza)

escape

Disponibilità

Flash Player 5.

Uso

```
escape(espressione)
```

Parametri

espressione L'espressione da convertire in stringa e da codificare in un formato con codifica URL.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte il parametro in una stringa in formato con codifica URL, dove tutti i caratteri non alfanumerici vengono sostituiti dalle sequenze esadecimali di escape precedute da `%`.

Esempio

L'esecuzione del codice seguente dà come risultato: Ciao%7B%5BMondo%5D%7D.

```
escape("Ciao{[Mondo]}");
```

Vedere anche

unescape

eval

Disponibilità

Flash Player 5 o versione successiva per avvalersi delle funzionalità complete. La funzione `eval` può essere usata in fase di esportazione in Flash Player 4, ma è necessario usare la sintassi della barra rovesciata `e`, inoltre, è possibile accedere solo alle variabili ma non alle proprietà o agli oggetti.

Uso

```
eval(espressione);
```

Parametri

espressione Una stringa contenente il nome di una variabile, una proprietà, un oggetto o un clip filmato da recuperare.

Valori restituiti

Nessuno.

Descrizione

Funzione; accede alle variabili, proprietà, oggetti o clip filmato in base al loro nome. Se *espressione* è una variabile o una proprietà, viene restituito il valore corrispondente. Se *espressione* è un oggetto o un clip filmato, viene restituito un riferimento a esso. Se l'elemento definito in *espressione* non è reperibile, viene restituito un valore indefinito.

In Flash 4, la funzione `eval` era usata per simulare le matrici, in Flash 5 si consiglia a tale scopo l'uso dell'oggetto `Array`.

È possibile usare la funzione `eval` anche per impostare e recuperare in modo dinamico il valore di una variabile o il nome di un'istanza. Tuttavia, è possibile eseguire tale operazione anche mediante l'operatore di accesso alla matrice (`[]`).

Nota: l'azione `eval` di ActionScript non equivale alla funzione `eval` di JavaScript e non può essere usata per valutare le istruzioni.

Esempio

Gli esempi seguenti usano la funzione `eval` per determinare il valore dell'espressione "piece" + x. Poiché il risultato è il nome di una variabile, `piece3`, la funzione `eval` restituisce il valore della variabile e lo assegna a y:

```
piece3 = "pericoloso";  
x = 3;  
  
y = eval("piece" + x);  
trace(y);  
  
// Output: pericoloso
```

Vedere anche

Array (oggetto)

evaluate

Disponibilità

Flash Player 5.

Uso

istruzione

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; crea una nuova riga vuota e inserisce un punto e virgola (;) per la creazione di istruzioni da valutare nel pannello Azioni.

false

Disponibilità

Flash Player 5.

Uso

true

Descrizione

Valore booleano univoco che rappresenta il contrario di true.

Vedere anche

true

FCheckBox (componente)

Il componente Casella di controllo nell'ambiente di creazione di Flash fornisce la funzionalità di trascinamento per l'aggiunta di caselle di controllo ai documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri principali. I metodi del componente FCheckBox consentono di controllare le caselle di controllo in fase di esecuzione: è possibile creare caselle di controllo, controllare quelle generate durante la creazione di codice, impostare o ridefinire i parametri principali e configurare nuove opzioni di runtime. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

Il componente Casella di controllo è supportato in Flash Player 6.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di ActionScript; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Per informazioni sull'uso del componente Casella di controllo, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" nel capitolo "Uso dei componenti" della *Guida all'uso di Flash*.

Riepilogo dei metodi per il componente FCheckBox

Metodo	Descrizione
<code>FCheckBox.setEnabled</code>	Restituisce <code>true</code> se la casella di controllo è attivata, <code>false</code> se invece è disattivata.
<code>FCheckBox.getLabel</code>	Restituisce l'etichetta applicata alla casella di controllo in forma di stringa.
<code>FCheckBox.getValue</code>	Restituisce <code>true</code> se la casella di controllo è selezionata, <code>false</code> se invece non è selezionata.
<code>FCheckBox.registerSkinElement</code>	Registra un elemento skin in una proprietà.
<code>FCheckBox.setChangeHandler</code>	Specifica un gestore delle modifiche da richiamare quando il valore della casella di controllo viene modificato.
<code>FCheckBox.setEnabled</code>	Determina se la casella di controllo è attivata o disattivata.
<code>FCheckBox.setLabel</code>	Specifica il testo per l'etichetta della casella di controllo.
<code>FCheckBox.setLabelPlacement</code>	Specifica se l'etichetta viene visualizzata a sinistra o a destra della casella di controllo.
<code>FCheckBox.setSize</code>	Imposta la larghezza in pixel della casella di controllo e la ridisegna.
<code>FCheckBox.setStyleProperty</code>	Imposta una sola proprietà di stile per un componente.
<code>FCheckBox.setValue</code>	Seleziona o deseleziona la casella di controllo e attiva la funzione di gestore delle modifiche.

FCheckBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Valore booleano che indica se l'istanza della casella di controllo è attivata (`true`) o disattivata (`false`).

Descrizione

Metodo; indica se l'istanza della casella di controllo è attivata o disattivata.

Esempio

Il seguente codice restituisce lo stato attivato di `checkBox1` nella finestra Output.

```
trace(checkBox1.setEnabled());
```

Vedere anche

`FCheckBox.setValue`

FCheckBox.getLabel

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.getLabel()
```

Parametri

Nessuno.

Valori restituiti

Stringa di testo.

Descrizione

Metodo; recupera l'etichetta della casella di controllo.

Esempio

Il seguente codice restituisce l'etichetta di checkBox1.

```
checkBox1.getLabel();
```

Vedere anche

FCheckBox.setLabel

FCheckBox.getValue

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.getValue()
```

Parametri

Nessuno.

Valori restituiti

Valore booleano che indica se la casella di controllo è selezionata (`true`) o deselezionata (`false`).

Descrizione

Metodo; indica se la casella di controllo è selezionata.

Esempio

Il seguente codice restituisce il valore selezionato di KowalczykBox nella finestra Output.

```
trace(KowalczykBox.getValue());
```

Vedere anche

FCheckBox.setValue

FCheckBox.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di FStyleFormat.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto FStyleFormat. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto FStyleFormat.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente FCheckBox usa gli elementi skin nella cartella FCheckBox Skins.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice registra l'elemento skin personalizzato `customChk_mc` nella proprietà `check` nel primo fotogramma del livello Leggimi dell'elemento skin `fcx_check` nella cartella FCheckBox Skins nella libreria.

```
check1.registerSkinElement(customChk_mc, "controllo");
```

FCheckBox.setChangeHandler

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setChangeHandler(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione di gestore da eseguire se il valore della casella di controllo viene modificato. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un gestore delle modifiche da richiamare quando il valore della casella di controllo viene modificato. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente che è stato modificato come parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore delle modifiche specificato durante la creazione del codice.

Per ulteriori informazioni, consultare "Creazione delle funzioni del gestore delle modifiche per i componenti" nel capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice specifica `myHandler` come la funzione richiamata se il valore di `checkBox1` viene modificato. Poiché il parametro *posizione* non è specificato, `myHandler` deve trovarsi nella stessa linea temporale dell'istanza del componente.

Il parametro `componente` in `myHandler` viene automaticamente compilato con l'istanza di un componente (il componente modificato in seguito all'input dell'utente e che specifica `myHandler` come gestore delle modifiche). Le azioni definite in `myHandler` determinano la visualizzazione di un messaggio nella finestra Output indicante che il componente è stato selezionato, quando l'utente seleziona una casella di controllo.

```
checkBox1.setChangeHandler("myHandler");
function myHandler(component){
trace(component._name + " è stato selezionato ");
}
```

Se nell'esempio precedente la funzione `myHandler` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
check1.setChangeHandler("myHandler", _parent._parent._parent);
```

Il seguente codice crea la funzione `myHandler` in un'istanza di `myObject` (della classe `Object`) e quindi specifica `myHandler` come funzione per `check1`.

```
My Object = new Object();
myObject.myHandler = function(component){
trace(component._name + " è stato selezionato ");
}

check1.setChangeHandler("myHandler", myObject);
```

FCheckBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setEnabled(attiva)
```

Parametri

attiva Valore booleano che indica se la casella di controllo è attivata (`true`) o disattivata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; indica se la casella di controllo è attivata (*true*) o disattivata (*false*). Se una casella di controllo è disattivata, non accetta l'interazione dell'utente mediante mouse o tastiera. Se questo parametro viene omissso, il metodo viene impostato automaticamente su *true*.

Esempio

Il seguente codice disattiva `checkBox1`.

```
checkBox1.setEnabled(false);
```

FCheckBox.setLabel

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setLabel(etichetta)
```

Parametri

etichetta Stringa che specifica l'etichetta del testo associato a una casella di controllo.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica l'etichetta del testo associato a una casella di controllo. Per impostazione predefinita, l'etichetta viene visualizzata a destra della casella di controllo. Se si richiama questo metodo, viene sovrascritto il valore del parametro *etichetta* specificato durante la creazione del codice.

Esempio

Il seguente codice consente di applicare l'etichetta "Invia altre informazioni" a `checkBox1`.

```
checkBox1.setLabel("Invia altre informazioni");
```

Vedere anche

`FCheckBox.getLabel`, `FCheckBox.setLabelPlacement`

FCheckBox.setLabelPlacement

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setLabelPlacement(posizioneEtichetta)
```

Parametri

posizioneEtichetta Stringa di testo; specificare "left" o "right".

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se l'etichetta viene visualizzata a sinistra o a destra della casella di controllo. Se si richiama tale metodo, viene sovrascritto il valore del parametro posizione etichetta impostato durante la creazione del codice.

Esempio

Il seguente codice posiziona l'etichetta per `checkBox1` a sinistra della casella di controllo.

```
checkBox1.setLabelPlacement("left");
```

Vedere anche

`FCheckBox.setLabel`

FCheckBox.setSize

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setSize(larghezza)
```

Parametri

larghezza Numero intero che specifica la larghezza in pixel della casella di controllo.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica la larghezza in pixel della casella di controllo e la ridisegna. Non è possibile impostare l'altezza dei componenti della casella di controllo. Se si richiama questo metodo, vengono sovrascritti i valori di ridimensionamento in scala della larghezza applicati durante la creazione del codice.

Per ulteriori informazioni, consultare la sezione "Ridimensionamento dei componenti Casella di controllo" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice imposta la larghezza di `checkBox1` su 200 pixel.

```
checkBox1.setSize(200);
```

FCheckBox.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto `FStyleFormat`.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di `FStyleFormat` per una singola istanza della casella di controllo. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di `FStyleFormat` per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice imposta la proprietà `shadow` relativa a `checkBox1` su `0x000000` (nero).

```
checkBox1.setStyleProperty("shadow", 0x000000);
```

Vedere anche

`FStyleFormat` (oggetto)

FCheckBox.setValue

Disponibilità

Flash Player 6.

Uso

```
myCheckBox.setValue(selezione)
```

Parametri

selezione Valore booleano che indica se la casella di controllo è selezionata (`true`) o deselezionata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; seleziona o deseleziona *myCheckBox* e attiva l'eventuale funzione di gestore delle modifiche specificata in fase di esecuzione. Il valore predefinito è `true`.

Sebbene richiamare questo metodo sovrascriva il valore del parametro valore iniziale impostato durante la creazione del codice, non usare il metodo a tale scopo, in quanto esso attiva anche la funzione di gestore delle modifiche associata. Per impostare il parametro valore iniziale di una casella di controllo in fase di esecuzione, usare `FCheckBox.setStyleProperty`.

Esempio

Il seguente codice seleziona l'istanza di `checkBox1` e attiva tutte le funzioni di gestore delle modifiche specificate.

```
checkBox1.setValue(true);
```

Vedere anche

`FCheckBox.getValue`

FComboBox (componente)

Il componente Casella combinata, nell'ambiente di creazione di Flash, fornisce la funzionalità di trascinamento per l'aggiunta di elenchi a discesa a scorrimento a selezione singola ai documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri principali. I metodi del componente FComboBox consentono di controllare le caselle combinate in fase di esecuzione: è possibile creare caselle combinate, controllare quelle generate durante la creazione di codice, impostare o ridefinire i parametri principali e configurare nuove opzioni di runtime. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

Il componente Casella combinata consente di creare caselle combinate sia statiche che modificabili. Una casella combinata statica è un elenco a discesa a scorrimento che consente all'utente di selezionare le voci in esso contenute. Una casella combinata modificabile è un elenco a discesa a scorrimento con un campo di testo nella parte superiore. È possibile consentire agli utenti di immettere il testo nell'apposito campo in modo che la casella combinata scorra fino alla voce desiderata, oppure usare il campo di testo per impostare il testo che verrà visualizzato in fase di esecuzione.

Entrambe le versioni del componente Casella combinata (statica e modificabile) elencano le voci dall'alto verso il basso usando un sistema di indicizzazione con base zero. Se il numero di voci contenute nella casella di riepilogo combinata crea un elenco a discesa che oltrepassa lo spazio disponibile al di sotto del componente, l'elenco si apre verso l'alto invece che verso il basso.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di ActionScript; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Il componente Casella combinata è supportato in Flash Player 6 e versioni successive.

Per informazioni sull'uso del componente Casella combinata, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare le sezioni "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi per il componente FComboBox

Metodo	Descrizione
<code>FComboBox.addItem</code>	Aggiunge una nuova voce alla fine della casella di riepilogo combinata.
<code>FComboBox.addItemAt</code>	Aggiunge una nuova voce alla casella di riepilogo combinata all'indice specificato.
<code>FComboBox.setEnabled</code>	Restituisce <code>true</code> se la casella combinata è attivata, <code>false</code> se invece è disattivata.
<code>FComboBox.getItemAt</code>	Restituisce la voce all'indice specificato come oggetto con le proprietà <code>label</code> e <code>data</code> .
<code>FComboBox.getLength</code>	Restituisce il numero di voci elencate nella casella combinata.
<code>FComboBox.getRowCount</code>	Restituisce il numero di righe visibili nella casella combinata.
<code>FComboBox.getScrollPosition</code>	Restituisce l'indice della voce nella parte superiore della casella combinata.
<code>FComboBox.getSelectedIndex</code>	Restituisce l'indice della voce attualmente selezionata.
<code>FComboBox.getSelectedItem</code>	Restituisce la voce attualmente selezionata come oggetto con le proprietà <code>label</code> e <code>data</code> .

Metodo	Descrizione
<code>FComboBox.getValue</code>	Restituisce il testo del campo di immissione per le caselle combinate modificabili; restituisce l'etichetta o i dati della voce selezionata, se si tratta di caselle statiche.
<code>FComboBox.registerSkinElement</code>	Registra un elemento skin in una proprietà.
<code>FComboBox.removeAll</code>	Rimuove tutte le voci dalla casella combinata.
<code>FComboBox.removeItemAt</code>	Rimuove la voce all'indice specificato.
<code>FComboBox.replaceItemAt</code>	Sostituisce l'etichetta e i dati di una voce all'indice specificato.
<code>FComboBox.setChangeHandler</code>	Assegna una funzione da richiamare ogni volta che una voce viene selezionata o che l'utente immette un testo nell'apposito campo.
<code>FComboBox.setDataProvider</code>	Registra un oggetto esterno nel componente come sorgente di dati.
<code>FComboBox.setEditable</code>	Indica se la casella combinata è modificabile (<code>true</code>) o statica (<code>false</code>).
<code>FComboBox.setEnabled</code>	Indica se la casella combinata è attivata (<code>true</code>) o disattivata (<code>false</code>).
<code>FComboBox.setItemSymbol</code>	Registra un ID di concatenamento del simbolo da usare per la visualizzazione delle voci in elenco nella casella combinata.
<code>FComboBox.setRowCount</code>	Determina il numero di voci visualizzate nella casella combinata senza la barra di scorrimento.
<code>FComboBox.setSelectedIndex</code>	Seleziona la voce all'indice specificato.
<code>FComboBox.setSize</code>	Imposta la larghezza in pixel della casella combinata.
<code>FComboBox.setStyleProperty</code>	Imposta un'unica proprietà di stile per l'istanza di un componente.
<code>FComboBox.setValue</code>	Specifica il testo visualizzato nell'apposito campo nella parte superiore di una casella combinata modificabile.
<code>FComboBox.sortItemsBy</code>	Visualizza le voci dell'elenco in ordine alfabetico o numerico in base all'etichetta o ai dati.

FComboBox.addItem

Disponibilità

Flash Player 6.

Uso

```
myComboBox.addItem(etichetta [, dati])
```

Parametri

etichetta Stringa di testo da visualizzare nella casella di riepilogo combinata.

dati Il valore da associare alla voce dell'elenco. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge una nuova voce con l'etichetta e i dati specificati alla fine della casella di riepilogo combinata e aggiorna l'elenco. I *dati* possono essere rappresentati da qualsiasi oggetto Flash, stringa, valore booleano, numero intero, oggetto o clip filmato.

Per ottenere i risultati migliori in termini di prestazioni e tempi di caricamento, evitare di aggiungere più di 400 voci in ciascun fotogramma. Questa regola è valida a prescindere dal fatto che si aggiungano voci a una sola o a più caselle di riepilogo combinate.

Esempio

Il seguente codice aggiunge la voce Carlo, con il valore associato Intelligente, alla fine dell'elenco della casella combinata `teacherList`.

```
teacherList.addItem("Carlo", Intelligente);
```

Il seguente codice aggiunge il numero massimo di voci consigliate per un singolo fotogramma (400) a `comboBox1`:

```
for (i=0; i<400; i++) {  
    comboBox1.addItem(i);  
}
```

Il seguente codice aggiunge il numero massimo di voci consigliate per un singolo fotogramma (400) a `listBox1` e a `comboBox2`:

```
for (i=0; i<200; i++) {  
    listBox1.addItem(i);  
    comboBox2.addItem(i);  
}
```

Vedere anche

`FComboBox.addItemAt`, `FComboBox.getItemAt`, `FComboBox.replaceItemAt`,
`FComboBox.setDataProvider`, `FComboBox.sortItemsBy`

FComboBox.addItemAt

Disponibilità

Flash Player 6.

Uso

```
myComboBox.addItemAt(indice, etichetta [, dati])
```

Parametri

indice Numero intero che specifica la posizione nella quale aggiungere la voce.

etichetta Stringa che identifica la voce dell'elenco nella casella combinata.

dati Il valore da associare alla voce dell'elenco. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge una nuova voce, con l'etichetta specificata e i dati opzionali associati, alla casella di riepilogo combinata nella posizione specificata. Il parametro `Dati` può essere rappresentato da qualsiasi oggetto `Flash`, stringa, valore booleano, numero intero, oggetto o clip filmato. Man mano che vengono aggiunte voci, l'elenco viene aggiornato e la barra di scorrimento ridimensionata.

Il componente `Casella combinata` usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Per ottenere i risultati migliori in termini di prestazioni e tempi di caricamento, evitare di aggiungere più di 400 voci in ciascun fotogramma. Questa regola è valida a prescindere dal fatto che si aggiungano voci a una sola o a più caselle di riepilogo combinate.

Esempio

Il seguente codice aggiunge la voce `Giacomo`, con il valore associato `Campione`, come quinta voce dell'elenco della casella combinata `Favorites`.

```
Favorites.addItemAt(4, "Giacomo", Campione);
```

Per gli esempi sul caricamento di un numero elevato di voci, consultare `FComboBox.addItem`.

Vedere anche

`FComboBox.getItemAt`, `FComboBox.removeItemAt`, `FComboBox.replaceItemAt`,
`FComboBox.setDataProvider`, `FComboBox.sortItemsBy`

FComboBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Valore booleano che indica se la casella combinata è attivata (`true`) o disattivata (`false`).

Descrizione

Metodo; indica se la casella combinata è attivata.

Esempio

Il seguente codice usa `setEnabled` per determinare se `comboBox1` è attivata o disattivata e visualizza il risultato nella finestra `Output`.

```
trace(comboBox1.setEnabled());
```

Vedere anche

`FComboBox.setEnabled`

FComboBox.getItemAt

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getItemAt(indice)
```

Parametri

indice Numero intero che specifica la posizione di una voce nella casella combinata.

Valori restituiti

Un oggetto.

Descrizione

Metodo; restituisce la voce all'indice specificato come oggetto con le proprietà `label` e `data`.

Il componente Casella combinata usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice restituisce in forma di stringa l'etichetta della voce all'indice 4 in `comboBox1`.

```
trace(comboBox1.getItemAt(4).label);
```

Il seguente codice restituisce i dati associati alla voce all'indice 4 in `comboBox2`. Il valore restituito dipende dal tipo di dati e può essere rappresentato da un oggetto, una stringa, un riferimento a un clip filmato o da un altro valore.

```
trace(comboBox2.getItemAt(4).data);
```

Il seguente codice restituisce un oggetto contenente il valore dell'etichetta e dei dati associati alla voce all'indice 4 in `comboBox3`.

```
trace(comboBox3.getItemAt(4));
```

Vedere anche

`FComboBox.getSelectedItem`

FComboBox.getLength

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getLength()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di voci contenute nella casella di riepilogo combinata.

Esempio

Il seguente codice recupera il numero di voci contenute nell'elenco di `listMain` e lo memorizza nella variabile `len`.

```
var len = listMain.getLength();
```

Vedere anche

`FComboBox.addItem`, `FComboBox.addItemAt`

FComboBox.getRowCount

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getRowCount()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di righe visibili nella casella combinata.

Esempio

Il seguente codice restituisce il numero di righe visibili in `toyList` e imposta il valore sulla variabile `rowCount`.

```
var rowCount = toyList.getRowCount();
```

Vedere anche

`FComboBox.setRowCount`

FComboBox.getScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getScrollPosition()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce la posizione nell'indice della voce attualmente visualizzata nella parte superiore della casella di riepilogo combinata.

Il componente casella combinata usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice recupera l'indice della voce che si trova attualmente all'inizio dell'elenco in `toyList` e la memorizza nella variabile `scrollPos`.

```
var scrollPos = toyList.getScrollPosition();
```

Vedere anche

`FComboBox.setSelectedIndex`

FComboBox.getSelectedIndex

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getSelectedIndex()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero o `undefined`.

Descrizione

Metodo; restituisce l'indice della voce attualmente selezionata nella casella combinata oppure restituisce `undefined`, nel caso in cui nessuna voce sia selezionata.

Le voci sono elencate nella casella combinata dall'alto verso il basso usando un indice con base zero.

Esempio

Il seguente codice recupera l'indice della voce attualmente selezionata in `toyList` e la memorizza nella variabile `selIndex`.

```
var selIndex = toyList.getSelectedIndex();
```

Vedere anche

`FComboBox.setSelectedIndex`

FComboBox.getSelectedItem

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getSelectedItem()
```

Parametri

Nessuno.

Valori restituiti

Un oggetto o `undefined`.

Descrizione

Metodo; restituisce la voce attualmente selezionata in forma di oggetto con le proprietà `label` e `data` oppure restituisce `undefined`, nel caso in cui nessuna voce sia selezionata.

Esempio

Il seguente codice recupera l'etichetta e i dati della voce attualmente selezionata in `comboBox1`.

```
trace(comboBox1.getSelectedItem());
```

Il seguente codice recupera l'etichetta della voce attualmente selezionata in `comboBox2`.

```
trace(comboBox2.getSelectedItem().label);
```

Il seguente codice recupera i dati della voce attualmente selezionata in `comboBox3`.

```
trace(comboBox3.getSelectedItem().data);
```

Vedere anche

`FComboBox.setSelectedIndex`

FComboBox.getValue

Disponibilità

Flash Player 6.

Uso

```
myComboBox.getValue()
```

Parametri

Nessuno.

Valori restituiti

Stringa di testo.

Descrizione

Metodo; restituisce il testo nel campo che si trova nella parte superiore della casella combinata, se quest'ultima è di tipo modificabile. Nel caso in cui la casella sia di tipo statico (e dunque non modificabile), questo metodo restituisce i dati associati alla voce selezionata o l'etichetta della voce, se non vi sono dati associati.

Esempio

Il seguente codice restituisce i dati o l'etichetta della voce attualmente selezionata in `menuMain`.

```
trace(menuMain.getValue())
```

Vedere anche

`FComboBox.setValue`

FComboBox.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myComboBox.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di `FStyleFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto `FStyleFormat`. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto `FStyleFormat`.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente `FComboBox` usa gli elementi skin nella cartella `FComboBox Skins`.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice registra l'elemento skin personalizzato `boundBox_mc` nella proprietà `background` nel primo fotogramma del livello Leggimi dell'elemento skin `FBoundingBox` nella cartella `Global Skins`.

```
toysMenu.registerSkinElement(boundBox_mc, "background");
```

FComboBox.removeAll

Disponibilità

Flash Player 6.

Uso

```
myComboBox.removeAll();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove tutte le voci della casella di riepilogo combinata, la aggiorna e ridimensiona la barra di scorrimento. Le caselle combinate prive di voci vengono visualizzate senza la barra di scorrimento. Questo metodo non può essere usato se la casella combinata è disattivata.

Esempio

Il seguente codice rimuove tutte le voci da `menuMain`.

```
menuMain.removeAll();
```

Vedere anche

`FComboBox.removeItemAt`

FComboBox.removeItemAt

Disponibilità

Flash Player 6.

Uso

```
myComboBox.removeItemAt(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da rimuovere.

Valori restituiti

Un oggetto contenente la voce rimossa.

Descrizione

Metodo; restituisce la voce rimossa all'indice specificato e aggiorna l'elenco. Quando una voce viene rimossa dall'elenco, gli indici delle voci successive vengono aggiornati in modo da indicare la nuova posizione di ciascuna voce nell'elenco. Se all'indice specificato non esiste alcuna voce, il metodo restituisce `undefined`.

Il componente Casella combinata usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice rimuove la quinta voce dall'elenco in `menuMain`.

```
menuMain.removeItemAt(4);
```

Vedere anche

`FComboBox.removeAll`

FComboBox.replaceItemAt

Disponibilità

Flash Player 6.

Uso

```
myComboBox.replaceItemAt(indice, etichetta [, dati])
```

Parametri

indice Numero intero che specifica la posizione nell'elenco di una voce.

etichetta Stringa che specifica una nuova etichetta per la voce dell'elenco.

dati Il nuovo valore da associare alla voce dell'elenco. Questo parametro è opzionale; se non viene specificato, qualsiasi dato attualmente specificato per la voce rimane valido.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiorna la voce all'indice specificato con l'etichetta e i dati specificati. Se la voce all'indice specificato ha un valore di dati associato e non viene specificato un valore per il parametro `dati`, il valore dei dati della voce in elenco non viene modificato.

Il componente Casella combinata usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice aggiorna la quinta voce nella casella combinata `Favorites` con etichetta `Nicola` e valore di dati `7439`. Se non è stato specificato nessun dato per la voce dell'elenco, a `Nicola` viene associato il valore di dati della voce dell'elenco esistente.

```
Favorites.replaceItemAt(4, "Nicola", "7439");
```

Vedere anche

`FComboBox.addItemAt`, `FComboBox.getItemAt`, `FComboBox.setDataProvider`,
`FComboBox.sortItemsBy`

FComboBox.setChangeHandler

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setChangeHandler(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione gestore da eseguire se la selezione nella casella combinata viene modificata. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un gestore delle modifiche da richiamare quando la selezione nella casella combinata viene modificata. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente che è stato modificato come parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore delle modifiche specificato durante la creazione del codice.

Per ulteriori informazioni, consultare la sezione "Creazione delle funzioni del gestore delle modifiche per i componenti" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice specifica `myHandler` come funzione richiamata se il valore di `toyList` viene modificato. Poiché il parametro `posizione` non è specificato, `myHandler` deve trovarsi nella stessa linea temporale dell'istanza del componente.

Il parametro `componente` in `myHandler` viene automaticamente compilato con l'istanza di un componente (il componente modificato in seguito all'input dell'utente e che specifica `myHandler` come gestore delle modifiche). Le azioni definite in `myHandler` determinano la visualizzazione dell'etichetta della voce nella finestra Output quando l'utente seleziona una voce dell'elenco.

```
toyList.setChangeHandler("myHandler");
function myHandler(component){
    trace(toyList.getSelectedItem().label);
}
```

Se nell'esempio precedente la funzione `myHandler` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
toyList.setChangeHandler("myHandler", _parent._parent._parent);
```

Il seguente codice crea la funzione `myHandler` in un'istanza di `myObject` (della classe `Object`) e quindi specifica `myHandler` come funzione per `toyList`.

```
myObject = new Object();
myObject.myHandler = function(component){
    trace(toyList.getSelectedItem().label);
}

toyList.setChangeHandler("myHandler", myObject);
```

FComboBox.setDataProvider

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setDataProvider(dataProvider)
```

Parametri

dataProvider Matrice di stringhe di testo contenente le voci da aggiungere; istanza dell'oggetto Array che specifica le voci da aggiungere o istanza della classe `DataProvider`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto esterno (*dataProvider*) come sorgente di dati per il componente Casella combinata. Se *dataProvider* è un'istanza dell'oggetto Array, l'oggetto può specificare `label`, `data` o entrambe le proprietà, dal momento che le proprietà dell'oggetto e il contenuto della matrice possono essere copiati nella casella combinata come etichette, dati o entrambi. Se *dataProvider* è un'istanza della classe `DataProvider`, è necessario implementare l'interfaccia API di `DataProvider` definita nel simbolo `DataProvider` nella cartella Flash UI Components/Core Assets/ClassTree nella libreria. L'interfaccia API di `DataProvider` è riservata agli utenti esperti e ai programmatori; per gli altri utenti si consiglia di usare una matrice o un oggetto Array.

Esempio

Il seguente codice specifica l'oggetto Array `peopleList1` come fornitore di dati per `comboBox1`.

```
comboBox1.setDataProvider(peopleList1);
```

Il seguente codice crea la matrice `peopleList` per la visualizzazione delle etichette delle voci elencate in `comboBox1`.

```
peopleList = new Array();
peopleList[0] = "BBianchi";
peopleList[1] = "CRossi";
peopleList[2] = "MRusso";
peopleList[3] = "MLucidi";
```

Il seguente codice crea la matrice `itemList1`, che specifica etichetta e dati relativi alle voci dell'elenco. In alternativa, è possibile usare l'oggetto Array come fornitore di dati per `comboBox1`.

```
itemList1 = new Array();
per (i=0; i<10; i++) {
    // Creazione di un elemento reale
    var myItem = new Object();
    myItem.label = "voce" + i;
    myItem.data = 75;
    // Inserimento nella matrice
    itemList1[i] = myItem;
}
```

Il seguente codice specifica `comboData`, un'istanza della classe `DataProvider`, come fornitore di dati per `comboBox1`.

```
comboBox1.setDataProvider(comboData);
```

Il codice seguente crea una nuova istanza della classe `DataProvider`, quindi aggiunge le etichette delle voci usando il metodo `addItem` di `DataProvider`.

Nota: il metodo `addItem` è solo uno dei metodi della classe `DataProvider`. I programmatori interessati all'uso della classe `DataProvider` devono fare riferimento al simbolo `DataProvider` nella cartella `Flash UI Components/CoreAssets/ClassTree` presente nella libreria prima di usare i metodi.

```
comboData = new DataProviderClass();

comboData.addItem("Debora");
comboData.addItem("Delia");
comboData.addItem("Vasco");
comboData.addItem("Alice");
```

Vedere anche

`FComboBox.addItem`, `FComboBox.replaceItemAt`, `FComboBox.sortItemsBy`

FComboBox.setEditable

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setEditable(modificabile)
```

Parametri

modificabile Valore booleano che indica se la casella combinata è modificabile (`true`) o statica (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; indica se la casella combinata è modificabile (`true`) o statica (`false`). Una casella combinata modificabile contiene un campo di testo; quando l'utente immette il testo, la casella scorre alla voce contenente lo stesso testo. Tale campo può essere usato anche per visualizzare il testo mediante `FComboBox.setValue`. Se si richiama questo metodo, viene sovrascritto il valore del parametro modificabile impostato durante la creazione del codice.

Esempio

Il seguente codice attiva un campo per l'immissione di testo nella parte superiore di `menuMain`.

```
menuMain.setEditable(true);
```

Vedere anche

`FComboBox.setValue`

FComboBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setEnabled(attiva)
```

Parametri

attiva Valore booleano che indica se la casella combinata è attivata (`true`) o disattivata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; determina se la casella combinata è attivata (`true`) o disattivata (`false`). Se una casella combinata è disattivata, non accetta l'interazione dell'utente mediante mouse o tastiera. Se si omette il parametro, questo metodo imposta il valore predefinito su `true`.

Esempio

Il seguente codice disattiva `menuMain`.

```
menuMain.setEnabled(false);
```

Vedere anche

`FListBox.setEnabled`

FComboBox.setItemSymbol

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setItemSymbol(IDsimbolo)
```

Parametri

IDsimbolo L'ID di concatenamento di un simbolo grafico per la visualizzazione del contenuto della casella combinata.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un simbolo grafico per la visualizzazione delle voci della casella di riepilogo combinata. Il valore predefinito è il simbolo `FComboBoxItem` contenuto nella libreria. Questo metodo è rivolto a programmatori e utenti esperti.

FComboBox.setRowCount

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setRowCount(righe)
```

Parametri

righe Il numero massimo di righe che l'elenco a discesa può visualizzare senza eseguire uno scorrimento.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta il numero di voci che possono essere visualizzate nell'elenco a discesa della casella combinata senza che venga eseguito uno scorrimento. Il valore minimo per il parametro *righe* è 3. Se si richiama questo metodo, viene sovrascritto il valore del parametro conteggio righe impostato durante la creazione del codice.

Esempio

Il seguente codice imposta il numero di voci visualizzate nell'elenco a discesa di `menuMain` su 4.

```
menuMain.setRowCount(4);
```

Vedere anche

`FComboBox.setSize`

FComboBox.setSelectedIndex

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setSelectedIndex(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da selezionare.

Valori restituiti

Nessuno.

Descrizione

Metodo; seleziona la voce specificata e aggiorna la casella combinata per mostrare la voce come selezionata. Richiamare questo metodo non ha effetto sullo stato corrente dell'elenco a discesa (aperto o chiuso). Tale metodo, inoltre, non può essere usato se la casella combinata è disattivata.

Il componente Casella combinata usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice seleziona la quarta voce nell'elenco di `menuMain`.

```
menuMain.setSelectedIndex(3);
```

Vedere anche

`FComboBox.setRowCount`

FComboBox.setSize

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setSize(larghezza)
```

Parametri

larghezza Numero intero che specifica la larghezza in pixel della casella combinata.

Valori restituiti

Nessuno.

Descrizione

Metodo; ridimensiona la casella combinata in base alla larghezza specificata (non è possibile impostare l'altezza di un componente Casella combinata). Usare questo metodo per ridimensionare in modo programmatico la casella combinata e aggiornarla in fase di esecuzione.

Esempio

Il seguente codice imposta o ridimensiona la larghezza di `menuMain` su 100 pixel.

```
menuMain.setSize(100);
```

Vedere anche

`FComboBox.setRowCount`

FComboBox.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto `FStyleFormat`.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di `FStyleFormat` per una singola istanza della casella combinata. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di `FStyleFormat` per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice imposta la proprietà `arrow` relativa a `comboBox1` su `0x000000` (nero).

```
comboBox1.setStyleProperty("arrow", 0x000000);
```

Vedere anche

`FStyleFormat` (oggetto)

FComboBox.setValue

Disponibilità

Flash Player 6.

Uso

```
myComboBox.setValue(testoModificabile)
```

Parametri

testoModificabile Stringa che specifica il testo che deve essere visualizzato nell'apposito campo di una casella combinata modificabile.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica il testo visualizzato nel campo di immissione nella parte superiore di una casella combinata modificabile. Se viene richiamato questo metodo, l'utente può comunque immettere il testo nell'apposito campo.

Questo metodo può essere usato solo con caselle combinate modificabili. Prima di richiamare il metodo, è necessario specificare il valore `true` per il parametro `modificabile` (che è impostato automaticamente su `false`) durante la creazione di codice, oppure usare `FComboBox.setEditable` per impostare il parametro su `true`.

Esempio

Il seguente codice immette la stringa `Gabino` nel campo superiore della casella combinata `surnameMenu`.

```
surnameMenu.setValue("Gabino");
```

Vedere anche

`FComboBox.getValue`

FComboBox.sortItemsBy

Disponibilità

Flash Player 6.

Uso

```
myComboBox.sortItemsBy(nomeCampo, ordine)
```

Parametri

nomeCampo Stringa che specifica il nome del campo usato per l'ordinamento. Il nome normalmente usato è "etichetta" o "dati".

ordine Stringa che specifica se ordinare le voci in ordine ascendente ("ASC") o discendente ("DESC").

Valori restituiti

Nessuno.

Descrizione

Metodo; elenca le voci nella casella combinata in ordine alfabetico o di numero, in base al tipo di ordinamento specificato e usando il nome del campo specificato. Se le voci in *nomeCampo* sono costituite da una combinazione di stringhe di testo e numeri interi, questi ultimi compaiono nell'elenco per primi. Il parametro *nomeCampo* è generalmente *etichetta* o *dati* ma è possibile assegnare un qualsiasi valore di base, a seconda delle proprie esigenze.

Esempio

Il seguente codice visualizza l'elenco di voci nella casella combinata `surnameMenu` in ordine ascendente usando le etichette delle voci.

```
surnameMenu.sortItemsBy("etichetta", "ASC");
```

Vedere anche

`FComboBox.addItemAt`, `FComboBox.replaceItemAt`, `FComboBox.setDataProvider`

FListBox (componente)

Il componente Casella di riepilogo nell'ambiente di creazione di Flash fornisce la funzionalità di trascinamento per l'aggiunta di caselle di riepilogo a discesa a scorrimento a selezione singola ai documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri principali. I metodi del componente `FListBox` consentono di controllare le caselle di riepilogo in fase di esecuzione: è possibile creare caselle di riepilogo, controllare quelle generate durante la creazione del codice, impostare o ridefinire i parametri principali e configurare nuove opzioni di runtime. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di ActionScript; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Il componente Casella di riepilogo è supportato in Flash Player 6 e versioni successive.

Per informazioni sull'uso del componente Casella di controllo, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare le sezioni "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi per il componente `FListBox`

Metodo	Descrizione
<code>FListBox.addItem</code>	Aggiunge una nuova voce alla fine della casella di riepilogo.
<code>FListBox.addItemAt</code>	Aggiunge una nuova voce alla casella di riepilogo all'indice specificato.
<code>FListBox.setEnabled</code>	Restituisce <code>true</code> se la casella di riepilogo è attivata, <code>false</code> se invece è disattivata.
<code>FListBox.getItemAt</code>	Restituisce l'etichetta e il valore della voce all'indice specificato.
<code>FListBox.getLength</code>	Restituisce il numero di voci nella casella di riepilogo.
<code>FListBox.getRowCount</code>	Restituisce il numero di voci visibili nella casella di riepilogo.
<code>FListBox.getScrollPosition</code>	Restituisce l'indice della voce nella parte superiore della casella di riepilogo.
<code>FListBox.getSelectedIndex</code>	Restituisce l'indice dell'ultima voce selezionata.
<code>FListBox.getSelectedIndices</code>	Restituisce gli indici di più voci selezionate.
<code>FListBox.getSelectedItem</code>	Restituisce l'etichetta e il valore della voce selezionata.
<code>FListBox.getSelectedItems</code>	Restituisce l'etichetta e il valore di più voci selezionate.
<code>FListBox.selectMultiple</code>	Restituisce <code>true</code> se è consentita la selezione multipla, <code>false</code> se è consentita la selezione singola.
<code>FListBox.getValue</code>	Restituisce l'etichetta della voce selezionata o qualsiasi altra informazione associata.
<code>FListBox.registerSkinElement</code>	Registra un elemento skin in una proprietà.
<code>FListBox.removeAll</code>	Rimuove tutte le voci dalla casella di riepilogo.
<code>FListBox.removeItemAt</code>	Rimuove la voce all'indice specificato.
<code>FListBox.replaceItemAt</code>	Sostituisce l'etichetta e i dati di una voce a un indice specificato con una nuova etichetta e nuovi dati.
<code>FListBox.setAutoHideScrollbar</code>	Determina se la barra di scorrimento è nascosta (<code>true</code>) o visualizzata (<code>false</code>) quando il numero di voci nella casella di riepilogo non rende necessario eseguire lo scorrimento.
<code>FListBox.setChangeHandler</code>	Assegna una funzione da richiamare ogni volta che la selezione viene modificata.
<code>FListBox.setDataProvider</code>	Associa un oggetto esterno alla casella di riepilogo.
<code>FListBox.setEnabled</code>	Indica se la casella di riepilogo è attivata (<code>true</code>) o disattivata (<code>false</code>).
<code>FListBox.setItemSymbol</code>	Registra un ID di concatenamento del simbolo da usare per la visualizzazione delle voci nella casella di riepilogo.
<code>FListBox.setRowCount</code>	Determina il numero di voci visualizzate nella casella di riepilogo.
<code>FListBox.setScrollPosition</code>	Determina lo scorrimento della casella di riepilogo in modo che la voce all'indice specificato sia visualizzata all'inizio dell'elenco.
<code>FListBox.setSelectedIndex</code>	Seleziona la voce all'indice specificato e aggiorna la casella di riepilogo.
<code>FListBox.setSelectedIndices</code>	Seleziona le voci agli indici specificati e aggiorna la casella di riepilogo.
<code>FListBox.selectMultiple</code>	Determina se all'utente è consentito selezionare più voci dell'elenco (<code>true</code>) oppure se non può eseguire questa operazione (<code>false</code>).

Metodo	Descrizione
<code>FListBox.setSize</code>	Imposta la larghezza e l'altezza in pixel della casella di riepilogo.
<code>FListBox.setStyleProperty</code>	Imposta una sola proprietà di stile per un componente.
<code>FListBox.setWidth</code>	Imposta la larghezza in pixel della casella di riepilogo.
<code>FListBox.sortItemsBy</code>	Visualizza le voci della casella di riepilogo in ordine alfabetico o numerico in base all'etichetta o ai dati.

FListBox.addItem

Disponibilità

Flash Player 6.

Uso

```
myListBox.addItem(etichetta [, dati])
```

Parametri

etichetta Stringa di testo che specifica la voce da aggiungere all'elenco.

dati Valore associato alla voce di elenco. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge una nuova voce con l'etichetta e i dati specificati (opzionali) alla fine della casella di riepilogo, aggiorna la casella e ridimensiona la barra di scorrimento. Il parametro Dati può essere rappresentato da qualsiasi oggetto Flash, stringa, valore booleano, numero intero, oggetto o clip filmato.

Per ottenere i risultati migliori in termini di prestazioni e tempi di caricamento, evitare di aggiungere più di 400 voci in ciascun fotogramma. Questa regola è valida a prescindere dal fatto che si aggiungano voci a una sola o a più caselle di riepilogo.

Esempio

Il seguente codice aggiunge Livia all'elenco di voci visualizzate nella casella di riepilogo `coolGirls`.

```
coolGirls.addItem("Livia");
```

Il seguente codice aggiunge il numero massimo di voci consigliate per un singolo fotogramma (400) a `listBox1`:

```
for (i=0; i<400; i++) {  
    listBox1.addItem(i);  
}
```

Il seguente codice aggiunge il numero massimo di voci consigliate per un singolo fotogramma (400) a `listBox1` e a `comboBox2`:

```
for (i=0; i<200; i++) {  
    listBox1.addItem(i);  
    comboBox2.addItem(i);  
}
```

Vedere anche

`FListBox.addItemAt`, `FListBox.getItemAt`, `FListBox.removeItemAt`,
`FListBox.replaceItemAt`, `FListBox.sortItemsBy`

FListBox.addItemAt

Disponibilità

Flash Player 6.

Uso

```
myListBox.addItemAt(indice, etichetta [, dati])
```

Parametri

indice Numero intero che specifica l'indice al quale aggiungere la voce.

etichetta Stringa di testo che specifica l'etichetta della voce.

dati Valore associato alla voce di elenco. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge una nuova voce con l'etichetta specificata e i dati associati (opzionali) all'indice specificato e aggiorna la casella di riepilogo. Il parametro dati può essere rappresentato da qualsiasi oggetto Flash, stringa, valore booleano, numero intero, oggetto o clip filmato.

L'indice del componente Casella di riepilogo è con base zero, pertanto la voce di indice 0 viene visualizzata in prima posizione nell'elenco.

Per ottenere i risultati migliori in termini di prestazioni e tempi di caricamento, evitare di aggiungere più di 400 voci in ciascun fotogramma. Questa regola è valida a prescindere dal fatto che si aggiungano voci a una sola o a più caselle di riepilogo.

Esempio

Il seguente codice aggiunge la voce *Davide* con il valore associato *amico* come quinta voce nella casella di riepilogo `peopleList`.

```
peopleList.addItemAt(4, "Davide", amico);
```

Per gli esempi sul caricamento di un numero elevato di voci, consultare `FListBox.addItem`.

Vedere anche

`FListBox.getSelectedItem`, `FListBox.removeItemAt`, `FListBox.replaceItemAt`, `FListBox.sortItemsBy`

FListBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myListBox.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Valore booleano che indica se la casella di riepilogo è attivata (`true`) o disattivata (`false`).

Descrizione

Metodo; indica se la casella di riepilogo è attivata.

Esempio

Il seguente codice usa `getEnabled` per determinare se `listMenu` è attivata o disattivata e visualizza il risultato nella finestra **Output**.

```
trace(listMenu.getEnabled());
```

Vedere anche

`FListBox.setEnabled`

FListBox.getItemAt

Disponibilità

Flash Player 6.

Uso

```
myListBox.getItemAt(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da recuperare.

Valori restituiti

Un oggetto o `undefined`.

Descrizione

Metodo; restituisce la voce all'indice specificato come oggetto con le proprietà `label` e `data`. Se all'indice specificato non esiste nessuna voce, il metodo restituisce `undefined`.

L'indice del componente Casella di riepilogo è con base zero, pertanto la voce di indice 0 viene visualizzata in prima posizione nell'elenco.

Esempio

Il seguente codice restituisce l'etichetta della voce all'indice 4 in `comboBox1` nella finestra **Output**.

```
trace(listMenu1.getItemAt(4).label);
```

Il seguente codice restituisce i dati o il valore associati alla voce all'indice 4 in `listMenu2` nella finestra **Output**.

```
trace(listMenu2.getItemAt(4).data);
```

Il seguente codice restituisce un oggetto contenente il valore dell'etichetta e dei dati associati alla voce all'indice 4 in `listMenu3` nella finestra **Output**.

```
trace(listMenu3.getItemAt(4));
```

Vedere anche

`FListBox.getSelectedItem`

FListBox.getLength

Disponibilità

Flash Player 6.

Uso

```
myListBox.getLength()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di voci nella casella di riepilogo.

Esempio

Il seguente codice restituisce il numero di voci contenute in `phoneList`.

```
phoneList.getLength();
```

Vedere anche

`FListBox.setSize`

FListBox.getRowCount

Disponibilità

Flash Player 6.

Uso

```
myListBox.getRowCount()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di righe visibili nella casella di riepilogo. Questo metodo è utile per determinare il numero di righe visualizzate in una casella di riepilogo misurata in pixel.

Esempio

Il seguente codice restituisce il numero di righe visibili in `toyList` e imposta il valore sulla variabile `rowCount`.

```
var rowCount = toyList.getRowCount();
```

Vedere anche

`FListBox.setRowCount`, `FListBox.setSize`

FListBox.getScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myListBox.getScrollPosition()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'indice della voce attualmente visualizzata nella parte superiore della casella di riepilogo.

Esempio

Il seguente codice restituisce l'indice della voce all'inizio di `staffList`.

```
staffList.getScrollPosition();
```

Vedere anche

`FListBox.setScrollPosition`

FListBox.getSelectedIndex

Disponibilità

Flash Player 5

Uso

```
myListBox.getSelectedIndex()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero o `undefined`.

Descrizione

Metodo; restituisce l'indice della voce attualmente selezionata in una casella di riepilogo a selezione singola, l'ultima voce selezionata in una casella di riepilogo a selezione multipla oppure restituisce `undefined` nel caso in cui nessuna voce sia selezionata. Per recuperare gli indici di tutte le voci selezionate in una casella di riepilogo a selezione multipla, usare

`FListBox.getSelectedIndices`.

Esempio

Il seguente codice restituisce l'indice della voce attualmente selezionata nella casella di riepilogo a selezione singola `nationList`.

```
nationList.getSelectedIndex();
```

Vedere anche

`FListBox.setSelectedIndices`, `FListBox.setSelectMultiple`

FListBox.getSelectedIndices

Disponibilità

Flash Player 6.

Uso

```
myListBox.getSelectedIndices()
```

Parametri

Nessuno.

Valori restituiti

Una matrice o `undefined`.

Descrizione

Metodo; restituisce in forma di matrice gli indici delle voci attualmente selezionate in una casella di riepilogo a selezione multipla oppure restituisce `undefined` nel caso in cui non vi siano voci selezionate.

Esempio

Il seguente codice restituisce in forma di matrice gli indici delle voci attualmente selezionate nella casella di riepilogo a selezione multipla `groceryList`.

```
groceryList.getSelectedIndices();
```

Vedere anche

`FListBox.getSelectedIndex`, `FListBox.setSelectMultiple`

FListBox.getSelectedItem

Disponibilità

Flash Player 6.

Uso

```
myListBox.getSelectedItem()
```

Parametri

Nessuno.

Valori restituiti

Un oggetto o `undefined`.

Descrizione

Metodo; restituisce la voce attualmente selezionata come oggetto con le proprietà `label` e `data`. Se sono state selezionate più voci, il metodo restituisce la voce dell'elenco selezionata per ultima; se non è stata selezionata alcuna voce, il metodo restituisce `undefined`. Per recuperare le informazioni su tutte le voci selezionate in una casella di riepilogo a selezione multipla, usare `FListBox.getSelectedItems`.

Esempio

Il seguente codice restituisce l'etichetta della voce attualmente selezionata in `listBox1`.

```
trace(listBox1.getSelectedItem().label);
```

Il seguente codice restituisce i dati o il valore associati alla voce attualmente selezionata in `listBox2`.

```
trace(listBox2.getSelectedItem().data);
```

Il seguente codice restituisce un oggetto contenente il valore dell'etichetta e dei dati associati alla voce attualmente selezionata in `listBox3`.

```
trace(listBox3.getSelectedItem());
```

Vedere anche

`FListBox.getItemAt`

FListBox.getSelectedItems

Disponibilità

Flash Player 6.

Uso

```
myListBox.getSelectedItems()
```

Parametri

Nessuno.

Valori restituiti

Una matrice o `undefined`.

Descrizione

Metodo; restituisce la voce attualmente selezionata come matrice di oggetti con le proprietà `label` e `data` oppure restituisce `undefined`, nel caso in cui non sia selezionata alcuna voce.

Questo metodo può essere usato solo per ottenere le voci selezionate in una casella di riepilogo a selezione multipla. Per recuperare le informazioni sulla voce attualmente selezionata in una casella di riepilogo a selezione singola, usare `FListBox.getSelectedItem`.

Esempio

Il seguente codice recupera le voci attualmente selezionate in `wishList` e le memorizza nella variabile `myObjArray`.

```
var myObjArray = wishList.getSelectedItems();
```

Vedere anche

`FListBox.getSelectedItem`, `FListBox.setSelectMultiple`

FListBox.getSelectedMultiple

Disponibilità

Flash Player 6.

Uso

```
myListBox.getSelectMultiple()
```

Parametri

Nessuno.

Valori restituiti

Un valore booleano.

Descrizione

Metodo; indica se agli utenti è consentito selezionare più voci (*true*) o una sola voce (*false*) tra quelle contenute nella casella di riepilogo.

Esempio

Il seguente codice verifica se `wishList` consente la selezione di più voci.

```
if (wishList.getSelectMultiple()) {  
}
```

Vedere anche

`FListBox.setSelectMultiple`

FListBox.getValue

Disponibilità

Flash Player 6.

Uso

```
myListBox.getValue()
```

Parametri

Nessuno.

Valori restituiti

L'etichetta o i dati associati a una voce selezionata.

Descrizione

Metodo; restituisce le informazioni sulla voce attualmente selezionata nella casella di riepilogo. Se alla voce non sono associati dati, il metodo restituisce l'etichetta della voce; in caso contrario, restituisce i dati.

Esempio

Il seguente codice restituisce l'etichetta della voce selezionata in `nationList`.

```
trace(nationList.getValue());
```

Vedere anche

`FListBox.getItemAt`

FListBox.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myListBox.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di `FStyleFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto `FStyleFormat`. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto `FStyleFormat`.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente `FListBox` usa gli elementi skin nella cartella `FListBox Skins`.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice registra l'elemento skin personalizzato `boundBox_mc` nella proprietà `background` nel primo fotogramma del livello Leggimi dell'elemento skin `FBoundingBox` nella cartella `Global Skins` nella libreria.

```
toysMenu.registerSkinElement(boundBox_mc, "background");
```

Vedere anche

`FStyleFormat` (oggetto)

FListBox.removeAll

Disponibilità

Flash Player 6.

Uso

```
myListBox.removeAll()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove tutte le voci dalla casella di riepilogo, la aggiorna e ridimensiona la barra di scorrimento.

Esempio

Il seguente codice rimuove tutte le voci da `wishList`.

```
wishList.removeAll();
```

Vedere anche

`FListBox.removeItemAt`

FListBox.removeItemAt

Disponibilità

Flash Player 6.

Uso

```
myListBox.removeItemAt(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da rimuovere.

Valori restituiti

Nessuno o `undefined`.

Descrizione

Metodo; rimuove la voce all'indice specificato, aggiorna gli indici delle voci in elenco seguenti la voce rimossa (in modo da indicare la nuova posizione della voce nell'elenco) e infine aggiorna la casella di riepilogo e ridimensiona la barra di scorrimento. Se all'indice specificato non esiste alcuna voce, il metodo restituisce `undefined`.

L'indice del componente Casella di riepilogo è con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice rimuove la quinta voce dell'elenco da `wishList`.

```
wishList.removeItemAt(4);
```

Vedere anche

`FListBox.addItemAt`

FListBox.replaceItemAt

Disponibilità

Flash Player 6.

Uso

```
myListBox.replaceItemAt(indice, etichetta [,dati])
```

Parametri

indice Numero intero che specifica la posizione nell'elenco di una voce.

etichetta Stringa che specifica una nuova etichetta per la voce dell'elenco.

dati Il nuovo valore da associare alla voce dell'elenco. Questo parametro è opzionale; se non viene specificato, qualsiasi dato attualmente associato alla voce rimane valido.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiorna la voce all'indice specificato con l'etichetta e i dati specificati. Se la voce all'indice specificato ha un valore di dati associato e non viene specificato un valore per il parametro dati, il valore di dati della voce in elenco non viene modificato.

Il componente Casella di riepilogo usa un indice con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il seguente codice aggiorna la quinta voce nella casella di riepilogo `Favorites` con la nuova etichetta `Fortunato` e il nuovo valore `Gatto`. Se il valore `Gatto` non fosse stato specificato e l'informazione associata alla quinta voce dell'elenco fosse stata `Cane`, il valore per `Fortunato` sarebbe stato `Cane` (valore errato in quanto si tratta di un gatto).

```
Favorites.replaceItemAt(4, "Fortunato", "Gatto");
```

Vedere anche

`FListBox.addItemAt`, `FListBox.getItemAt`

FListBox.setAutoHideScrollbar

Disponibilità

Flash Player 6.

Uso

```
myListBox.setAutoHideScrollbar(nascondiScorrimento)
```

Parametri

nascondiScorrimento Valore booleano che specifica se la barra di scorrimento è nascosta quando non è necessario usarla (`true`) o è sempre visualizzata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se la barra di scorrimento è nascosta quando le voci nella casella di riepilogo possono essere visualizzate senza usare la barra stessa (`true`) oppure se la barra è sempre visualizzata (`false`). Se questo metodo è impostato su `false` e il numero di voci non richiede l'uso della barra di scorrimento, quest'ultima appare disattivata, ovvero visualizzata in grigio.

Esempio

Il seguente codice nasconde la barra di scorrimento per `wishList` quando le voci nella casella di riepilogo possono essere visualizzate senza usare la funzione di scorrimento.

```
wishList.setAutoHideScrollbar(true);
```

FListBox.setChangeHandler

Disponibilità

Flash Player 6.

Uso

```
myListBox.setChangeHandler(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione gestore da eseguire se la selezione nella casella di riepilogo viene modificata. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un gestore delle modifiche da richiamare quando la selezione nella casella di riepilogo viene modificata. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente modificata in parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore delle modifiche specificato durante la creazione del codice.

Per ulteriori informazioni, consultare la sezione "Creazione delle funzioni del gestore delle modifiche per i componenti" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il seguente codice specifica `myHandler` come funzione richiamata se il valore di `listBox1` viene modificato. Poiché il parametro *posizione* non è specificato, `myHandler` deve trovarsi nella stessa linea temporale dell'istanza del componente.

Il parametro `componente` in `myHandler` viene automaticamente compilato con l'istanza di un componente (il componente modificato in seguito all'input dell'utente e che specifica `myHandler` come gestore delle modifiche). Le azioni definite in `myHandler` determinano la visualizzazione dell'etichetta della voce nella finestra Output quando l'utente seleziona una voce dell'elenco.

```
listBox1.setChangeHandler("myHandler");  
function myHandler(componente){  
    trace(listBox1.getSelectedItem().label);  
}
```

Se nell'esempio precedente la funzione `myHandler` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
listBox1.setChangeHandler("myHandler", _parent._parent._parent);
```


Il seguente codice crea la funzione `myHandler` in un'istanza di `myObject` (della classe `Object`) e quindi specifica `myHandler` come funzione per `listBox1`.

```
someListener = new Object();
myObject.myHandler = function(component){
    trace(listBox1.getSelectedItem().label);
}

listBox1.setChangeHandler("myHandler", myObject);
```

FListBox.setDataProvider

Disponibilità

Flash Player 6.

Uso

```
myListBox.setDataProvider(dataProvider)
```

Parametri

dataProvider Matrice di stringhe di testo contenente le voci da aggiungere; istanza dell'oggetto `Array` che specifica le voci da aggiungere o istanza della classe `DataProvider`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto esterno (*dataProvider*) come sorgente di dati per il componente Casella di riepilogo. Se *dataProvider* è un'istanza dell'oggetto `Array`, l'oggetto può specificare le proprietà `label`, `data` o entrambi, dal momento che le proprietà dell'oggetto e il contenuto della matrice possono essere copiati nella casella combinata come etichette, dati o in entrambi i modi. Se *dataProvider* è un'istanza della classe `DataProvider`, è necessario implementare l'interfaccia API di `DataProvider` definita nel simbolo `DataProvider` nella cartella `Flash UI Components/Core Assets/ClassTree` nella libreria. L'interfaccia API di `DataProvider` è riservata agli utenti esperti e ai programmatori; per gli altri utenti si consiglia di usare una matrice o un oggetto `Array`.

Esempio

Il seguente codice specifica l'oggetto `Array` `writerList` come fornitore di dati per `listBox1`.

```
listBox1.setDataProvider(writerList);
```

Il seguente codice crea la matrice `writerList` per la visualizzazione delle etichette delle voci elencate in `listBox1`.

```
writerList1 = new Array();
writerList[0] = "Giulio";
writerList[1] = "Maria";
writerList[2] = "Marcello";
writerList[3] = "Delia";
writerList[4] = "Stefania";
writerList[5] = "Barbara";
```

Il seguente codice crea la matrice `itemList1`, che specifica etichetta e dati relativi alle voci dell'elenco. Questo oggetto `Array` potrebbe essere usato come fornitore di dati alternativo per `listBox1`.

```
itemList1 = new Array();
per (i=0; i<10; i++) {

// Creazione di un elemento reale
    var myItem = new Object();
    myItem.label = "voce" + i;
    myItem.data = 75;

// Inserimento nella matrice
    itemList1[i] = myItem;
}
```

Il codice seguente specifica `editorList`, un'istanza della classe `DataProvider`, come fornitore di dati per `listBox1`.

```
listBox1.setDataProvider(editorList);
```

Il codice seguente crea una nuova istanza della classe `DataProvider`, quindi aggiunge le etichette delle voci usando il metodo `addItem` di `DataProvider`.

Nota: il metodo `addItem` è solo uno dei metodi della classe `DataProvider`. I programmatori interessati all'uso della classe `DataProvider` devono fare riferimento al simbolo `DataProvider` nella cartella `Flash UI Components/CoreAssets/ClassTree` presente nella libreria prima di usare i metodi.

```
editorList = new DataProviderClass();
editorList.addItem("Anna");
editorList.addItem("Rossana");
editorList.addItem("Lisa");
editorList.addItem("Rebecca");
```

Vedere anche

`FListBox.addItem`, `FListBox.replaceItemAt`, `FListBox.sortItemsBy`

FListBox.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myListBox.setEnabled(attiva)
```

Parametri

attiva Valore booleano che specifica se la casella di riepilogo è attivata (`true`) o disattivata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se la casella di riepilogo è attivata (`true`) o disattivata (`false`). Se una casella di riepilogo viene disattivata, non accetta alcuna interazione con il mouse o la tastiera da parte dell'utente. Se si omette il parametro, questo metodo imposta il valore predefinito su `true`.

Esempio

Il codice seguente disattiva `interestList`.

```
interestList.setEnabled(false);
```

Vedere anche

`FListBox.setEnabled`

FListBox.setItemSymbol

Disponibilità

Flash Player 6.

Uso

```
myListBox.setItemSymbol(IDsimbolo)
```

Parametri

IDsimbolo L'ID di concatenamento di un simbolo grafico che consente di visualizzare il contenuto della casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un simbolo grafico per visualizzare le voci nella casella di riepilogo. Il valore predefinito è il simbolo `FListBoxItem` nella libreria. Questo metodo è rivolto a programmatori e utenti esperti.

FListBox.setRowCount

Disponibilità

Flash Player 6.

Uso

```
myListBox.setRowCount(righe)
```

Parametri

righe Il numero massimo di righe visualizzate nella casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica il numero di voci visualizzate nella casella di riepilogo. Se si usa tale metodo, per impostare la larghezza della casella di riepilogo utilizzare `ListBox.setWidth`, non `FListBox.setSize`. Se si richiama `FListBox.setSize`, viene sovrascritto il valore del parametro conteggio righe impostato durante la creazione del codice. Pertanto, se si richiama tale metodo dopo aver richiamato `FListBox.setRowCount`, il filmato ignorerà l'impostazione `rowCount` e l'altezza della casella di riepilogo verrà impostata in pixel.

Esempio

Il codice seguente imposta su 4 il numero di voci visualizzate in `toyList`.

```
toyList.setRowCount(4);
```

Vedere anche

`FListBox.getRowCount`, `FListBox.setSize`

FListBox.setScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myListBox.setScrollPosition(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da visualizzare in prima posizione nella casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; determina lo scorrimento della casella di riepilogo in modo che la voce specificata venga visualizzata in prima posizione.

L'indice del componente Casella di riepilogo è con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Nel codice seguente, la quinta voce di `toyList` si trova in prima posizione nell'elenco.

```
toyList.setScrollPosition(4);
```

Vedere anche

`FListBox.getScrollPosition`

FListBox.setSelectedIndex

Disponibilità

Flash Player 6.

Uso

```
myListBox.setSelectedIndex(indice)
```

Parametri

indice Numero intero che specifica l'indice della voce da selezionare nella casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; seleziona la voce con l'indice specificato e aggiorna la casella di riepilogo.

L'indice del componente Casella di riepilogo è con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Nel codice seguente, viene selezionata la quinta voce nella casella di riepilogo.

```
toyList.setSelectedIndex(4);
```

Vedere anche

`FListBox.getSelectedIndex`, `FListBox.getSelectedIndices`

FListBox.setSelectedIndices

Disponibilità

Flash Player 6.

Uso

```
myListBox.setSelectedIndices(matriceIndice)
```

Parametri

matriceIndice Matrice di indici di voci da selezionare nella casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; seleziona le voci specificate nella matrice di indici e aggiorna la casella di riepilogo.

L'indice del componente Casella di riepilogo è con base zero, nel quale la voce all'indice 0 viene visualizzata all'inizio dell'elenco.

Esempio

Il codice seguente crea una matrice denominata `myArray` che specifica le voci da selezionare in `toyList` e passa quindi la matrice al metodo `setSelectedIndices`.

```
var myArray = new Array (1,4,5,7);  
toyList.setSelectedIndices(myArray);
```

Vedere anche

`FListBox.getSelectedIndices`

FListBox.setSelectMultiple

Disponibilità

Flash Player 6.

Uso

```
myListBox.setSelectMultiple(selezioneMultipla)
```

Parametri

selezioneMultipla Valore booleano che specifica la modalità di selezione multipla (`true`) o la modalità di selezione singola (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se all'utente è consentito selezionare nella casella di riepilogo più voci (`true`) o solo voci singole (`false`). L'impostazione predefinita è `false`. Se si richiama tale metodo, viene sovrascritto il valore del parametro selezione multipla impostato durante la creazione del codice.

Esempio

Il codice seguente attiva la selezione multipla per `toyList`.

```
toyList.setSelectMultiple(true);
```

FListBox.setSize

Disponibilità

Flash Player 6.

Uso

```
myListBox.setSize(larghezza, altezza)
```

Parametri

larghezza Numero intero che specifica in pixel la larghezza della casella di riepilogo.

altezza Numero intero che specifica in pixel l'altezza della casella di riepilogo.

Valori restituiti

Nessuno.

Descrizione

Metodo; modifica in fase di esecuzione le dimensioni della casella di riepilogo in base alla larghezza e all'altezza specificata. Se si richiama questo metodo, viene sovrascritto il valore del parametro conteggio righe impostato durante la creazione del codice. Pertanto, se si richiama tale metodo dopo aver richiamato `ListBox.setRowCount`, il filmato imposterà l'altezza della casella di riepilogo in pixel e ignorerà l'impostazione `rowCount`. Per impostare la larghezza di una casella di riepilogo durante l'uso di `setRowCount`, usare `FListBox.setWidth`.

Esempio

Il codice seguente modifica le dimensioni di `phoneList` in modo che queste corrispondano a 200 pixel in larghezza e 50 pixel in altezza.

```
phoneList.setSize(200, 50);
```

Vedere anche

`FListBox.setRowCount`, `FListBox.setWidth`

FListBox.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myListBox.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto FStyleFormat.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di FStyleFormat per una singola istanza della casella di riepilogo. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di FStyleFormat per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la proprietà di shadow di `listBox1` su `0x000000` (nero).

```
listBox1.setStyleProperty("shadow", 0x000000);
```

Vedere anche

FStyleFormat (oggetto)

FListBox.setWidth

Disponibilità

Flash Player 6.

Uso

```
myListBox.setWidth(larghezza)
```

Parametri

larghezza Valore intero che specifica in pixel la larghezza della casella di riepilogo

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica in pixel la larghezza della casella di riepilogo. Tale metodo consente di impostare la larghezza della casella di riepilogo quando si usa `FListBox.setRowCount` per determinarne l'altezza.

Esempio

Il codice seguente imposta la larghezza della casella di riepilogo `toyList` su 500 pixel.

```
toyList.setWidth(500);
```

Vedere anche

`FListBox.setSize`

FListBox.sortItemsBy

Disponibilità

Flash Player 6.

Uso

```
myListBox.sortItemsBy(nomeCampo, ordine)
```

Parametri

nomeCampo Stringa che specifica il nome del campo usato per l'ordinamento. Il nome normalmente usato è "etichetta" o "dati".

ordine Stringa che specifica se ordinare le voci in ordine ascendente ("ASC") o discendente ("DESC").

Valori restituiti

Nessuno.

Descrizione

Metodo; ordina alfabeticamente o numericamente le voci della casella di riepilogo usando l'ordine e il nome di campo specificati. Se le voci in *nomeCampo* sono costituite da una combinazione di stringhe di testo e numeri interi, questi ultimi compaiono nell'elenco per primi. Il parametro *nomeCampo* generalmente usato è "etichetta" o "dati", ma gli utenti esperti e i programmatori possono specificare qualsiasi parametro di base a seconda delle proprie esigenze.

Esempio

Il codice seguente ordina le voci della casella di riepilogo `surnameMenu` in ordine ascendente, usando le etichette delle voci di elenco.

```
surnameMenu.sortItemsBy("etichetta", "ASC");
```

Vedere anche

`FListBox.addItemAt`, `FListBox.replaceItemAt`

_focusrect

Disponibilità

Flash Player 4.

Uso

```
_focusrect = valore booleano;
```


Descrizione

Proprietà (globale); specifica se visualizzare un rettangolo giallo attorno al pulsante attivato da tastiera. Il valore predefinito è `true` e determina la visualizzazione di un rettangolo giallo attorno al pulsante o al campo di testo attualmente attivo quando l'utente preme il tasto `TAB` per spostarsi da un oggetto all'altro in un filmato. Specificare `false` se non si desidera visualizzare il rettangolo giallo. Si tratta di una proprietà di tipo globale che può essere ridefinita per istanze specifiche.

Vedere anche

`Button._focusrect`

for

Disponibilità

Flash Player 5.

Uso

```
for(inizializzazione; condizione; passaggio); {  
    istruzione/i;  
}
```

Parametri

inizializzazione Espressione da valutare prima di iniziare la sequenza di iterazioni del ciclo, in genere un'espressione di assegnazione. Per questo parametro è consentita anche un'istruzione `var`.

condizione Espressione che restituisce `true` o `false`. La condizione viene valutata prima di ciascuna iterazione del ciclo; il ciclo termina quando la condizione restituisce il valore `false`.

passaggio Espressione da valutare dopo ciascuna iterazione del ciclo, in genere un'espressione di assegnazione che usa gli operatori `++` (incremento) o `--` (decremento).

istruzione/i Una o più istruzioni da eseguire all'interno del corpo del ciclo.

Descrizione

Azione; una funzione di ciclo che valuta l'espressione *inizializzazione* una volta, quindi inizia una sequenza di iterazioni durante le quali, fino a quando il parametro *condizione* restituisce `true`, viene eseguito il parametro *istruzione* e viene valutata l'espressione *passaggio*.

Alcune proprietà non possono essere enumerate dall'azione `for` o `for..in`. È il caso, ad esempio, dei metodi predefiniti dell'oggetto `Array` (`Array.sort` e `Array.reverse`), che non sono inclusi nell'enumerazione di un oggetto `Array`, e delle proprietà dei clip filmato quali `_x` e `_y`.

Esempio

L'esempio seguente usa il ciclo `for` per inserire le voci in una matrice:

```
for(i=0; i<10; i++) {  
    array [i] = (i + 5)*10;  
    trace(array[i]);  
}
```

I seguenti risultati sono visualizzati nella finestra Output:

```
50
60
70
80
90
100
110
120
130
140
```

L'esempio seguente usa il ciclo `for` per eseguire più volte la stessa azione. Nel codice seguente, il ciclo `for` è usato per aggiungere i numeri da 1 a 100.

```
var sum = 0;
  for (var i=1; i<100; i++) {
    sum = sum + i;
  }
```

Vedere anche

`++` (incremento), `--` (decremento), `for..in`, `var`

for..in

Disponibilità

Flash Player 5.

Uso

```
for(variabileIterazione in oggetto){
  istruzione/i;
}
```

Parametri

variabileIterazione Il nome di una variabile che agisce da elemento di iterazione e fa riferimento a ciascuna proprietà di un oggetto o elemento di una matrice.

oggetto Il nome di un oggetto su cui eseguire l'iterazione.

istruzione/i Un'istruzione da eseguire per ogni iterazione.

Valori restituiti

Nessuno.

Descrizione

Azione; esegue un ciclo sulle proprietà di un oggetto o elemento di una matrice ed esegue l'*istruzione* per ciascuna proprietà di un oggetto.

Alcune proprietà non possono essere enumerate dall'azione `for` o `for..in`. È il caso, ad esempio, dei metodi predefiniti dell'oggetto `Array` (`Array.sort` e `Array.reverse`), che non sono inclusi nell'enumerazione di un oggetto `Array`, e delle proprietà dei clip filmato quali `_x` e `_y`.

L'azione `for..in` esegue le iterazioni sulle proprietà degli oggetti presenti nella catena del prototipo dell'oggetto. Se il prototipo dell'oggetto secondario è un oggetto principale, l'esecuzione dell'iterazione sulle proprietà dell'oggetto secondario con il ciclo `for..in` comporterà l'esecuzione dell'iterazione anche sulle proprietà dell'oggetto principale.

L'azione `for..in` enumera tutti gli oggetti presenti nella catena del prototipo di un oggetto. Vengono enumerate, in quest'ordine, le proprietà dell'oggetto, le proprietà del suo diretto prototipo, le proprietà del prototipo del prototipo, e così via. L'azione `for..in` non enumera due volte proprietà con il medesimo nome. Se l'oggetto secondario ha come prototipo un oggetto principale ed entrambi contengono la proprietà `prop`, l'azione `for..in` richiamata per l'oggetto secondario enumera `prop` a partire dall'oggetto secondario ma ignora la proprietà nell'oggetto principale.

Esempio

L'esempio seguente usa il ciclo `for..in` per eseguire l'iterazione sulle proprietà di un oggetto.

```
myObject = { name:'Tara', age:27, city:'San Francisco' };
for (name in myObject) {
    trace ("myObject." + name + " = " + myObject[name]);
}
```

Il risultato di questo esempio è il seguente:

```
myObject.name = Tara
myObject.age = 27
myObject.city = San Francisco
```

L'esempio seguente usa l'operatore `typeof` con il ciclo `for..in` per eseguire l'iterazione su un tipo particolare di oggetto secondario:

```
for (name in myMovieClip) {
    if (typeof (myMovieClip[name]) = "movieclip") {
        trace ("Nome oggetto secondario clip filmato: " + name);
    }
}
```

L'esempio seguente enumera gli oggetti secondari di un clip filmato e sposta ciascuno di essi al fotogramma 2 nella rispettiva linea temporale. Il clip filmato `RadioButtonGroup` è un oggetto principale con diversi oggetti secondari: `_RedRadioButton_`, `_GreenRadioButton_` e `_BlueRadioButton_`.

```
for (var name in RadioButtonGroup) {
    RadioButtonGroup[name].gotoAndStop(2);
}
```

FPushButton (componente)

Il componente Pulsante presente nell'ambiente di creazione di Flash fornisce la funzionalità di trascinamento per l'aggiunta di pulsanti a documenti Flash; offre inoltre un'interfaccia utente per l'impostazione di parametri di base. I metodi del componente `FPushButton` consentono di controllare i pulsanti in fase di esecuzione: è possibile creare pulsanti, controllare i pulsanti creati, impostare o ridefinire i parametri di base e impostare opzioni di runtime aggiuntive. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

Il componente Pulsante accetta tutte le normali interazioni eseguite tramite mouse e tastiera. I metodi di `FPushButton` possono essere usati per specificare una funzione del gestore per i pulsanti e modificare le dimensioni dei pulsanti senza distorsioni in fase di esecuzione.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di `ActionScript`; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Il componente Pulsante è supportato da Flash Player 6 e versioni successive.

Per informazioni sull'uso del componente Pulsante, l'impostazione dei parametri durante la creazione del codice e la modifica del colore e dell'aspetto dei componenti, consultare le sezioni "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi validi per il componente FPushButton

Metodo	Descrizione
<code>FPushButton.setEnabled</code>	Restituisce <code>true</code> se il pulsante è attivato, <code>false</code> se è disattivato.
<code>FPushButton.getLabel</code>	Restituisce l'etichetta del pulsante come stringa.
<code>FPushButton.registerSkinElement</code>	Registra un elemento skin in una proprietà.
<code>FPushButton.setClickHandler</code>	Specifica la funzione richiamata quando l'utente rilascia il pulsante.
<code>FPushButton.setEnabled</code>	Determina se il pulsante è attivato o disattivato.
<code>FPushButton.setLabel</code>	Imposta l'etichetta del pulsante in fase di esecuzione.
<code>FPushButton.setSize</code>	Imposta l'altezza e la larghezza del pulsante in pixel.
<code>FPushButton.setStyleProperty</code>	Imposta una sola proprietà di stile per un componente.

FPushButton.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Un valore booleano.

Descrizione

Metodo; restituisce `true` se l'istanza del pulsante è attivata, `false` se è disattivata.

Esempio

Il codice seguente restituisce lo stato attivato del pulsante `submit` nella finestra Output.

```
trace(submit.setEnabled());
```

Vedere anche

`FPushButton.setEnabled`

FPushButton.getLabel

Disponibilità

Flash Player 6.

Uso

```
myPushButton.getLabel()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce come stringa l'etichetta di testo sul pulsante.

Esempio

Il codice seguente restituisce l'etichetta del pulsante `buttonPage1` nella finestra Output.

```
trace(buttonPage1.getLabel());
```

Vedere anche

`FPushButton.setLabel`

FPushButton.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myPushButton.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di `FStyleFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto `FStyleFormat`. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto `FStyleFormat`.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Dopo averlo aggiunto al documento Flash, il componente `FPushButton` usa gli elementi skin contenuti nella cartella `FPushButton Skins` e l'elemento skin `FLabel` contenuto nella cartella `Global Skins`.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente registra l'elemento skin personalizzato `newFace_mc` nella proprietà `face` nel primo fotogramma del livello Leggimi dell'elemento skin `FLabel`. L'elemento skin `FLabel` è contenuto nella cartella `Component Skins/Global Skins` della libreria.

```
submitButton.registerSkinElement(newFace_mc, "face");
```

Vedere anche

`FStyleFormat` (oggetto)

FPushButton.setClickHandler

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setClickHandler(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione del gestore da eseguire quando l'utente rilascia il pulsante. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica una funzione del gestore da richiamare quando l'utente rilascia il pulsante. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente modificato come parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore dei clic specificato durante la creazione del codice.

Per ulteriori informazioni, consultare la sezione "Creazione delle funzioni del gestore delle modifiche per i componenti" del capitolo "Uso di componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente consente di richiamare la funzione `onClick` quando viene modificato il valore di `button1`. Non essendo specificato il parametro *posizione*, `onClick` deve trovarsi nella stessa linea temporale dell'istanza del componente. Nel parametro `componente` in `onClick` viene automaticamente inserita l'istanza di un componente (il componente modificato in seguito all'input dell'utente e che specifica `onClick` come gestore delle modifiche). Le azioni definite in `onClick` determinano la visualizzazione dell'etichetta del pulsante nella finestra `Output` al rilascio del pulsante da parte dell'utente.

```
button1.setClickHandler("onClick");  
function onClick(component){  
    trace(component._name);  
}
```

Se nell'esempio precedente la funzione `onClick` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
button1.setChangeHandler("onClick", _parent._parent._parent);
```

Il codice seguente crea la funzione `onClick` in un'istanza di `myObject` (appartenente alla classe `Object`) e specifica `onClick` come funzione per `button1`.

```
myObject = new Object();
myObject.onClick = function(component){
    trace(component._name);
}
button1.setChangeHandler("onClick", myObject);
```

FPushButton.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setEnabled(attiva)
```

Parametri

attiva Valore booleano che specifica se il pulsante è attivato (`true`) o disattivato (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; determina se il pulsante è attivato. Se un pulsante viene disattivato, non accetta alcuna interazione con il mouse o la tastiera e il testo del pulsante appare disattivato, ovvero visualizzato in grigio. Se questo parametro viene omesso, viene usato il valore `true`.

Esempio

Il codice seguente disattiva `button1`.

```
button1.setEnabled(false);
```

Vedere anche

`FPushButton.setEnabled`

FPushButton.setLabel

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setLabel(etichetta)
```

Parametri

etichetta Stringa contenente il testo da visualizzare sul pulsante.

Valori restituiti

Nessuno.

Descrizione

Metodo; applica un'etichetta di testo al pulsante in fase di esecuzione. Se si richiama tale metodo, viene sovrascritto il valore del parametro `etichetta` specificato durante la creazione del codice.

Esempio

Il codice seguente applica l'etichetta `Cleveland Rocks!` a `voteButton`.

```
voteButton.setLabel("Cleveland Rocks!");
```

Vedere anche

`FPushButton.getLabel`

FPushButton.setSize

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setSize(larghezza, altezza)
```

Parametri

larghezza Numero intero che specifica in pixel la larghezza del pulsante.

altezza Numero intero che specifica in pixel l'altezza del pulsante.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la larghezza e l'altezza del pulsante in fase di esecuzione. Se si richiama questo metodo, vengono sovrascritte le dimensioni o le proporzioni applicate durante la creazione del codice. Per ulteriori informazioni, consultare la sezione "Ridimensionamento dei componenti Pulsante" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente ridimensiona `submitButton` a 100 x 50 pixel in fase di esecuzione.

```
submitButton.setSize(100, 50);
```

FPushButton.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myPushButton.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto `FStyleFormat`.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di `FStyleFormat` per una singola istanza del pulsante. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di `FStyleFormat` per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la proprietà `face` di `submitButton` su `0xffffffff` (bianco).

```
submitButton.setStyleProperty("face", 0xffffffff);
```

Vedere anche

`FStyleFormat` (oggetto)

FRadioButton (componente)

I pulsanti di scelta sono costituiti da gruppi di pulsanti selezionabili a esclusione reciproca. Il componente Pulsante di scelta nell'ambiente di creazione di Flash fornisce funzionalità di trascinarsi della selezione, che consentono di aggiungere gruppi di pulsanti di scelta a documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri di base. I metodi del componente `FRadioButton` consentono di controllare i pulsanti di scelta in fase di esecuzione: è possibile creare pulsanti di scelta, gestire i pulsanti creati, impostare o ridefinire i parametri di base e impostare opzioni di runtime aggiuntive. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

Il componente Pulsante di scelta è supportato da Flash Player 6 e versioni successive.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di `ActionScript`; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Per informazioni sull'uso del componente Pulsante di scelta, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare le sezioni "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi validi per il componente FRadioButton

Metodo	Descrizione
<code>FRadioButton.getData</code>	Restituisce un valore dei dati relativi all'istanza di un pulsante di scelta.
<code>FRadioButton.setEnabled</code>	Restituisce <code>true</code> se il pulsante di scelta è attivato, <code>false</code> se è disattivato.
<code>FRadioButton.getLabel</code>	Restituisce l'etichetta del pulsante di scelta come stringa.
<code>FRadioButton.getState</code>	Restituisce lo stato selezionato dell'istanza di un pulsante di scelta.
<code>FRadioButton.getValue</code>	Restituisce il valore dei dati relativi al pulsante di scelta selezionato in un gruppo oppure restituisce l'etichetta nel caso in cui non sia specificato alcun dato.
<code>FRadioButton.registerSkinElement</code>	Registra un elemento skin in una proprietà di stile.
<code>FRadioButton.setChangeHandler</code>	Specifica la funzione da richiamare quando la selezione del pulsante di scelta cambia.
<code>FRadioButton.setData</code>	Imposta i dati associati all'istanza di un pulsante di scelta.
<code>FRadioButton.setEnabled</code>	Determina se il pulsante di scelta è attivato o disattivato.
<code>FRadioButton.setGroupName</code>	Specifica un nome di gruppo per l'istanza di un pulsante di scelta oppure imposta un nuovo nome per un gruppo di pulsanti di scelta.
<code>FRadioButton.setLabel</code>	Applica un'etichetta per il pulsante di scelta in fase di esecuzione.
<code>FRadioButton.setLabelPlacement</code>	Specifica se l'etichetta viene visualizzata a sinistra o a destra dei pulsanti di scelta.
<code>FRadioButton.setSize</code>	Imposta la larghezza del pulsante di scelta in pixel.
<code>FRadioButton.setState</code>	Imposta lo stato selezionato dell'istanza di un pulsante di scelta.
<code>FRadioButton.setStyleProperty</code>	Imposta una sola proprietà di stile per l'istanza di un componente.
<code>FRadioButton.setValue</code>	Seleziona un pulsante di scelta da un gruppo di pulsanti di scelta in fase di esecuzione.

FRadioButton.getData

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.getData()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce i dati associati all'istanza del pulsante di scelta specificata. Usare `FRadioButton.getValue` per ottenere i dati associati al pulsante di scelta selezionato da un gruppo di pulsanti di scelta.

Esempio

Il codice seguente restituisce i dati associati al pulsante di scelta `flashRadio` nella finestra Output.

```
trace(flashRadio.getData());
```

Vedere anche

`FRadioButton.setData`

FRadioButton.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setEnabled()  
myRadioButtonsGroup.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Un valore booleano o `undefined`.

Descrizione

Metodo; indica se è attivata l'istanza di un pulsante di scelta o un gruppo di pulsanti di scelta.

Uso 1: indica se `myRadioButton` è attivato (`true`) o disattivato (`false`).

Uso 2: indica se i pulsanti di `myRadioButtonsGroup` sono attivati (`true`) o disattivati (`false`). Se in un gruppo alcuni pulsanti risultano attivati e altri disattivati, il metodo restituisce il valore `undefined`.

Esempio

Il codice seguente restituisce lo stato attivato di `radio1` alla finestra Output.

```
trace(radio1.setEnabled());
```

Vedere anche

`FRadioButton.setEnabled`

FRadioButton.getLabel

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.getLabel()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce l'etichetta del pulsante di scelta specificato come stringa. Tale metodo non può essere usato per ottenere le etichette di un gruppo di pulsanti di scelta; la sintassi `radioButtonGroup.getLabel` non è valida.

Esempio

Il codice seguente restituisce l'etichetta dell'istanza `radio2` nella finestra Output.

```
trace(radio2.getLabel());
```

Vedere anche

`FRadioButton.setLabel`

FRadioButton.getState

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.getState()
```

Parametri

Nessuno.

Valori restituiti

Un valore booleano che indica lo stato selezionato del pulsante di scelta.

Descrizione

Metodo; restituisce un valore booleano che indica se `myRadioButton` è selezionato (`true`) o no (`false`).

Esempio

Il codice seguente restituisce lo stato selezionato del pulsante di scelta `radio1` nella finestra Output.

```
trace(radio1.getState());
```

Vedere anche

`FRadioButton.setState`

FRadioButton.getValue

Disponibilità

Flash Player 6.

Uso

```
myRadioButtonGroup.getValue()
```

Parametri

Nessuno.

Valori restituiti

Una stringa o il valore `undefined`.

Descrizione

Metodo; restituisce i dati associati al pulsante di scelta selezionato in *myRadioButtonGroup* oppure l'etichetta del pulsante di scelta, nel caso in cui non sia specificato alcun dato. Nel caso in cui non sia selezionato alcun pulsante, il metodo restituisce il valore `undefined`.

Esempio

Il codice seguente restituisce i dati associati al pulsante di scelta selezionato nel gruppo `radioGroup1` nella finestra `Output`.

```
trace(radioGroup1.getValue());
```

Vedere anche

`FRadioButton.setValue`

FRadioButton.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di `FStyleFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto `FStyleFormat`. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto `FStyleFormat`.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente `FRadioButton` usa gli elementi skin contenuti nella cartella `FRadioButton Skins` e l'elemento skin `FLabel` contenuto nella cartella `Global Skins`.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente registra l'elemento skin personalizzato `myDot_mc` nella proprietà di `FStyleFormat radioDot` del file `Leggimi` per l'elemento skin `frb_dot`, contenuto nella cartella `Skins` di `FRadioButton` presente nella libreria.

```
radio1.registerSkinElement(myDot_mc, "radioDot");
```

Vedere anche

`FStyleFormat` (oggetto)

FRadioButton.setChangeListener

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setChangeListener(nomeFunzione, [posizione])
```

```
myRadioButtongroup.setChangeListener(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione del gestore da eseguire quando il valore di un pulsante di scelta cambia. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica una funzione del gestore delle modifiche da richiamare quando cambia la selezione di un pulsante di scelta. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente che è stato modificato come parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore delle modifiche specificato durante la creazione del codice.

Uso 1: specifica la funzione da richiamare nel caso in cui l'istanza del pulsante di scelta *myRadioButton* sia selezionata o deselezionata.

Uso 2: specifica la funzione da richiamare in caso di variazione relativa al pulsante di scelta selezionato nel gruppo *radioButtongroup*. Specificare una funzione per un gruppo di pulsanti di scelta corrisponde a specificare la stessa funzione per i singoli pulsanti di scelta del gruppo mediante *myRadioButton.setChangeListener*.

Per ulteriori informazioni, consultare la sezione "Creazione delle funzioni del gestore delle modifiche per i componenti" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Uso 1: il codice seguente consente di richiamare la funzione `myHandler` quando è selezionato `radio1`.

```
radio1.setChangeHandler("myHandler");
```

Uso 2: il codice seguente consente di richiamare la funzione `onChange` quando è selezionato un pulsante di scelta del gruppo `radioGroup1`.

```
radioGroup1.setChangeHandler("onChange");
```

Il codice seguente consente di richiamare la funzione `onChange` quando viene selezionato un pulsante di scelta contenuto in `radioGroup1`. Non essendo specificato il parametro *posizione*, `onChange` deve trovarsi nella stessa linea temporale dell'istanza del componente. Il parametro componente in `onChange` viene impostato automaticamente con il componente (il componente modificato in seguito all'input dell'utente e che specifica `onChange` come gestore delle modifiche), in questo caso un pulsante di scelta contenuto nel gruppo. Le azioni definite in `onChange` determinano la visualizzazione del nome dell'istanza nella finestra Output quando un pulsante di scelta viene selezionato dall'utente.

```
radioGroup1.setChangeHandler("onChange");  
function onChange(component){  
    trace(component._name);  
}
```

Se nell'esempio precedente la funzione `onChange` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
radioGroup1.setChangeHandler("onChange", _parent._parent._parent);
```

Il codice seguente crea la funzione `onChange` in un'istanza di `myObject` (che appartiene alla classe `Object`) e specifica `onChange` come funzione per `radioGroup1`.

```
someListener = new Object();  
myObject.onChange = function(component){  
    trace(component._name);  
}  
  
radioGroup1.setChangeHandler("onChange", myObject);
```

FRadioButton.setData

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setData("dati")
```

Parametri

dati I dati da associare all'istanza del pulsante di scelta.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica i dati da associare all'istanza del pulsante di scelta. Se si richiama tale metodo, viene sovrascritto il valore del parametro `dati` impostato durante la creazione del codice.

Esempio

Il codice seguente specifica i dati `ActionScript` per l'istanza del pulsante di scelta `flashRadio`.

```
flashRadio.setData("ActionScript");
```

Vedere anche

`FRadioButton.getData`, `FRadioButton.setValue`

FRadioButton.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setEnabled(attiva)
```

```
myRadioButtonGroup.setEnabled(attiva)
```

Parametri

attiva Valore booleano che specifica se un singolo pulsante di scelta o tutti i pulsanti di un gruppo sono attivati (`true`) o disattivati (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; attiva e disattiva i pulsanti di scelta in fase di esecuzione.

Uso 1: specifica se *myRadioButton* è attivato (`true`) o disattivato (`false`).

Uso 2: specifica se tutti i pulsanti di scelta con il nome di gruppo *radioButtonGroup* sono attivati (`true`) o disattivati (`false`). Se si richiama questo metodo senza specificare alcun parametro, viene usato il parametro `true`.

Esempio

Uso 1: il codice seguente disattiva il solo pulsante di scelta `radio1` senza disattivare i restanti pulsanti del gruppo.

```
radio1.setEnabled(false);
```

Uso 2: il codice seguente disattiva tutti i pulsanti di scelta del gruppo `radioGroup1`.

```
radioGroup1.setEnabled(false);
```

Vedere anche

`FRadioButton.setEnabled`

FRadioButton.setGroupName

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setGroupName(nomeGruppo)
```

```
myRadioButtonGroup.setGroupName(nomeGruppo)
```


Parametri

nomeGruppo Stringa che specifica il nome di un gruppo di pulsanti di scelta.

Valori restituiti

Nessuno.

Descrizione

Metodo; applica un nome di gruppo all'istanza di un pulsante di scelta o a un gruppo di pulsanti di scelta in fase di esecuzione. Se si richiama tale metodo, viene sovrascritto il valore del parametro *nomeGruppo* impostato durante la creazione del codice.

Uso 1: specifica *myRadioButton* come membro del gruppo di pulsanti di scelta definiti in *nomeGruppo*.

Uso 2: specifica un nuovo nome di gruppo per tutti i pulsanti di scelta di *myRadioButtongroup*.

Esempio

Uso 1: il codice seguente specifica *Colori* come nome di gruppo dell'istanza del pulsante di scelta *radioRed*.

```
radioRed.setGroupName("Colori");
```

Uso 2: il codice seguente specifica *radioGroupToys* come nuovo nome di gruppo di tutti i pulsanti di scelta del gruppo *radioGroupGames*.

```
radioGroupGames.setGroupName("radioGroupToys");
```

FRadioButton.setLabel

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setLabel(etichetta)
```

Parametri

etichetta Stringa di testo che specifica l'etichetta visualizzata a destra del pulsante di scelta.

Valori restituiti

Nessuno.

Descrizione

Metodo; applica un'etichetta all'istanza del pulsante di scelta *myRadioButton* in fase di esecuzione. Se si richiama tale metodo, viene sovrascritto il valore del parametro *etichetta* specificato durante la creazione del codice. Tale metodo non può essere usato per impostare etichette per gruppi di pulsanti di scelta; la sintassi *radioButtongroup.setLabel* non è valida.

Esempio

Il codice seguente applica l'etichetta *Occhi marroni* a *radio1*.

```
radio1.setLabel("Occhi marroni");
```

Vedere anche

`FRadioButton.getLabel`

FRadioButton.setLabelPlacement

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setLabelPlacement(posizioneEtichetta)
```

```
myRadioButtonGroup.setLabelPlacement(posizioneEtichetta)
```

Parametri

posizioneEtichetta Stringa di testo; specificare "left" o "right".

Descrizione

Metodo; specifica se l'etichetta viene visualizzata a sinistra o a destra del pulsante di scelta. Se si richiama tale metodo, viene sovrascritto il valore del parametro *posizione etichetta* impostato durante la creazione del codice.

Uso 1: specifica la posizione dell'etichetta per un solo pulsante di scelta.

Uso 2: specifica la posizione delle etichette per tutti i pulsanti di scelta di un gruppo.

Esempio

Uso 1: il codice seguente posiziona l'etichetta per `radio1` a sinistra del pulsante di scelta.

```
radio1.setLabelPlacement("left");
```

Uso 2: il codice seguente posiziona le etichette dei pulsanti di scelta del gruppo `Colori` a destra dei pulsanti stessi.

```
Colors.setLabelPlacement("right");
```

Vedere anche

`FRadioButton.setLabel`, `FRadioButton.setLabelPlacement`

FRadioButton.setSize

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setSize(larghezza)
```

```
myRadioButtonGroup.setSize(larghezza)
```

Parametri

larghezza Numero intero che specifica in pixel la dimensione del pulsante di scelta.

Valori restituiti

Nessuno.

Descrizione

Metodo; ridisegna il pulsante di scelta e ne specifica la larghezza in pixel (non è possibile impostare l'altezza dei componenti Pulsante di scelta). Richiamare questo metodo ridisegna le proporzioni in larghezza applicate durante la creazione del codice.

Uso 1: imposta le dimensioni di un solo pulsante di scelta.

Uso 2: imposta le dimensioni di tutti i pulsanti di scelta appartenenti a un gruppo.

Per ulteriori informazioni, consultare la sezione "Ridimensionamento dei componenti Pulsante di scelta" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la larghezza di `radio1` su 200 pixel.

```
radio1.setSize(200);
```

FRadioButton.setState

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setState("seleziona")
```

Parametri

seleziona Valore booleano che indica se il pulsante di scelta è selezionato (`true`) o no (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se *myRadioButton* è selezionato (`true`) o no (`false`). In un gruppo (i cui pulsanti abbiano tutti lo stesso parametro nome gruppo) può avere lo stato iniziale di `true` (selezionato) un solo pulsante di scelta. Nel caso in cui sia specificato per tale parametro il valore `true` per più pulsanti di scelta, viene selezionato l'ultimo di tali pulsanti per il quale sia stato indicato il valore `true` per il parametro dello stato iniziale. Il valore predefinito per tale parametro è `false`.

Se si richiama questo metodo, viene sovrascritto il valore del parametro stato iniziale impostato durante la creazione del codice. Nel caso in cui, per selezionare un pulsante di scelta in fase di esecuzione, si richiami sia tale metodo che il valore `FRadioButton.setValue` e i pulsanti di scelta siano pulsanti differenti appartenenti allo stesso gruppo, viene selezionato il pulsante di scelta specificato nell'ultimo metodo richiamato.

Esempio

Il codice seguente seleziona il pulsante di scelta `radio1` in fase di esecuzione.

```
radio1.setState(true);
```

Vedere anche

`FRadioButton.getState`, `FRadioButton.getValue`, `FRadioButton.setValue`

FRadioButton.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myRadioButton.setStyleProperty(proprietàStile, valore)
```

```
myRadioButtonsGroup.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto FStyleFormat.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di FStyleFormat per una sola istanza del pulsante di scelta. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di FStyleFormat per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo di un componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la proprietà `radioDot` per `radioButton1` su `0xFF12AC` (rosa).

```
radioButton1.setStyleProperty("radioDot", 0xFF12AC);
```

Il codice seguente imposta la proprietà `radioDot` per tutti i pulsanti di `radioGroup1` su `0xFF12AC` (rosa).

```
radioGroup1.setStyleProperty("radioDot", 0xFF12AC);
```

Vedere anche

FStyleFormat (oggetto)

FRadioButton.setValue

Disponibilità

Flash Player 6.

Uso

```
myRadioButtonsGroup.setValue("dati")
```

Parametri

dati I dati associati al pulsante di scelta da selezionare.

Valori restituiti

Nessuno.

Descrizione

Metodo; seleziona il pulsante di scelta associato ai dati specificati e deseleziona contemporaneamente tutti i pulsanti eventualmente selezionati appartenenti allo stesso gruppo.

Se si richiama questo metodo, viene sovrascritto il valore del parametro valore iniziale impostato durante la creazione del codice. Nel caso in cui, per selezionare un pulsante di scelta in fase di esecuzione, si richiami sia tale metodo che il valore `FRadioButton.setValue` e i pulsanti di scelta siano pulsanti differenti appartenenti allo stesso gruppo, viene selezionato il pulsante di scelta specificato nell'ultimo metodo richiamato.

Esempio

Il codice seguente seleziona il pulsante di scelta associato ai dati `rosso` relativi al gruppo di pulsanti di scelta denominato `Colori`.

```
Colors.setValue("rosso");
```

Vedere anche

`FRadioButton.getData`, `FRadioButton.getValue`, `FRadioButton.setState`

FScrollBar (componente)

Il componente Barra di scorrimento nell'ambiente di creazione di Flash fornisce funzionalità di trascinamento della selezione che consentono di aggiungere barre di scorrimento a campi di testo dinamici e di input nei documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri di base. I metodi del componente `FScrollBar` consentono di controllare le barre di scorrimento in fase di esecuzione: è possibile creare nuove barre di scorrimento, controllare le barre di scorrimento create, impostare o ridefinire i parametri di base e impostare opzioni di runtime aggiuntive. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti. Si sconsiglia l'uso di alcuni metodi del componente `FScrollBar` per le barre di scorrimento associate ai campi di testo. A tale proposito, consultare le voci relative ai singoli metodi.

Gli utenti esperti e i programmatori possono usare il componente Barra di scorrimento con altri elementi di Flash per creare interfacce utente personalizzate.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di `ActionScript`; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Il componente Barra di scorrimento è supportato da Flash Player 6 e versioni successive.

Per informazioni sull'uso del componente Barra di scorrimento, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" nel capitolo "Uso dei componenti" della *Guida all'uso di Flash*.

Riepilogo dei metodi validi per il componente FScrollBar

Metodo	Descrizione
<code>FScrollBar.setEnabled</code>	Restituisce <code>true</code> se la barra di scorrimento è attivata, <code>false</code> se è disattivata.
<code>FScrollBar.getScrollPosition</code>	Restituisce un numero intero che rappresenta la posizione corrente della casella di scorrimento.
<code>FScrollBar.registerSkinElement</code>	Registra un elemento skin in una proprietà definita per un elemento skin presente nel livello Leggimi e posizionato nel Fotogramma 1 di un clip filmato contenuto nella libreria.

Metodo	Descrizione
<code>FScrollBar.setChangeHandler</code>	Specifica una funzione da richiamare ogni volta che la posizione di scorrimento cambia (tale metodo non può essere usato con campi di testo).
<code>FScrollBar.setEnabled</code>	Specifica se la barra di scorrimento è attivata (<code>true</code>) o disattivata (<code>false</code>).
<code>FScrollBar.setHorizontal</code>	Specifica se la barra di scorrimento è orizzontale (<code>true</code>) o verticale (<code>false</code>).
<code>FScrollBar.setLargeScroll</code>	Specifica il numero di posizioni di avanzamento dello scorrimento quando si fa clic sulla traccia.
<code>FScrollBar.setScrollContent</code>	Specifica l'istanza del campo di testo a cui si applica la barra di scorrimento.
<code>FScrollBar.setScrollPosition</code>	Imposta la posizione della casella di scorrimento mediante un numero intero compreso tra <code>minPos</code> e <code>maxPos</code> .
<code>FScrollBar.setScrollProperties</code>	Imposta le proprietà <code>pageSize</code> , <code>minPos</code> e <code>maxPos</code> per la barra di scorrimento (tale metodo non può essere usato con campi di testo).
<code>FScrollBar.setScrollTarget</code>	Specifica un campo di testo come target della barra di scorrimento.
<code>FScrollBar.setSize</code>	Imposta la lunghezza della barra di scorrimento in pixel.
<code>FScrollBar.setSmallScroll</code>	Specifica il numero di posizioni di avanzamento quando si fa clic sulla freccia di scorrimento.
<code>FScrollBar.setStyleProperty</code>	Imposta una sola proprietà di stile per un componente.

FScrollBar.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setEnabled()
```

Parametri

Nessuno.

Valori restituiti

Un valore booleano.

Descrizione

Metodo; indica se la barra di scorrimento è attivata (`true`) o disattivata (`false`).

Esempio

Il codice seguente restituisce un valore alla finestra Output che indica se `scroll1` è attivato (`true`) o disattivato (`false`).

```
trace(scroll1.setEnabled());
```

Vedere anche

`FScrollBar.setEnabled`

FScrollBar.getScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.getScrollPosition()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce un numero intero che specifica la posizione della casella di scorrimento. Il valore restituito è compreso nell'intervallo definito dalle proprietà `minPos` e `maxPos` che determinano i limiti della barra di scorrimento. Per impostare i parametri `minPos` e `maxPos`, usare `FScrollBar.setScrollProperties`.

Esempio

Il codice seguente restituisce la posizione corrente della casella di scorrimento `scroll12` alla finestra Output. Se per `scroll12` la proprietà `minPos` è impostata su 2 e `maxPos` su 25, la restituzione del valore 12 indica che la casella di scorrimento è posizionata al centro della barra di scorrimento.

```
trace(scroll12.getPosition());
```

Consultare la voce relativa a `FScrollBar.setChangeHandler` per un altro esempio dell'uso di questo metodo.

Vedere anche

`FScrollBar.setChangeHandler`, `FScrollBar.setScrollPosition`

FScrollBar.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Stringa che specifica una proprietà di `FStyleFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto `FStyleFormat`. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto `FStyleFormat`.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente `FScrollBar` usa gli elementi skin contenuti nella cartella `FScrollBar Skins` e l'elemento skin `FLabel` contenuto nella cartella `Global Skins`. La modifica di un elemento skin contenuto nella cartella `FScrollBar Skins` si applica a tutti i componenti che usano le barre di scorrimento (Casella combinata, Casella di riepilogo, Barra di scorrimento e Riquadro di scorrimento).

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente registra l'elemento skin personalizzato `NewArrow_mc` nella proprietà `arrow` nel primo fotogramma del livello Leggimi dell'elemento skin `fsb_downArrow`, contenuto nella cartella `FScrollBar Skins` presente nella libreria.

```
Scroll11.registerSkinElement(NewArrow_mc, "arrow");
```

Vedere anche

`FStyleFormat` (oggetto)

FScrollBar.setChangeHandler

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setChangeHandler(nomeFunzione, [posizione])
```

Parametri

nomeFunzione Stringa che specifica il nome della funzione del gestore da eseguire quando la casella di scorrimento viene spostata. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un gestore delle modifiche da richiamare quando si sposta la casella di scorrimento. È possibile specificare la stessa funzione del gestore delle modifiche per più componenti; la funzione accetta sempre l'istanza del componente che è stato modificato come parametro. Se si richiama questo metodo, viene sovrascritto il valore del parametro gestore delle modifiche specificato durante la creazione del codice.

Questo metodo è indirizzato a utenti esperti e programmatori che desiderano creare applicazioni e componenti personalizzati usando il componente dell'interfaccia utente di Flash Barra di scorrimento; non è possibile usare questo metodo con barre di scorrimento associate a campi di testo.

Esempio

Il codice seguente crea una casella piena sullo stage, vi applica una barra di scorrimento orizzontale, imposta le proprietà di scorrimento e specifica la funzione `mover` come gestore delle modifiche. Il gestore delle modifiche `mover` usa la posizione della barra di scorrimento per modificare la posizione `_x` del clip filmato da 50 a 250.

```
root.createEmptyMovieClip("square", 1);
_root.square._x = 50;
_root.square._y = 50;

con (_root.square) {
    moveTo(0, 0);
    beginFill(0x0066CC);
    lineTo(20, 0);
    lineTo(20, 20);
    lineTo(0, 20);
    lineTo(0, 0);
    endFill();
}

scrollBar._x = 50;
scrollBar.setHorizontal(true);
scrollBar.setScrollProperties(1, 50, 250);
scrollBar.setChangeHandler("mover");

funzione mover () {
    _root.square._x = scrollBar.getScrollPosition();
}
```

Il codice seguente specifica una funzione del gestore delle modifiche per un'istanza del componente Barra di scorrimento associato al componente Casella di riepilogo personalizzata. Il gestore delle modifiche imposta `scroll1` in modo che possa ottenere la posizione di scorrimento attuale tramite `FScrollBar.getScrollPosition`, quindi `customListBox` usa `FScrollBar.setScrollPosition` per reimpostare la posizione di scorrimento in modo che la voce presente nella posizione di scorrimento attuale venga visualizzata per prima nella casella di riepilogo personalizzata. Nel parametro `componente` viene automaticamente inserita l'istanza di un componente (il componente modificato in seguito all'input dell'utente e che specifica `myHandler` come gestore delle modifiche).

```
scroll1.setChangeHandler("myHandler");

function myHandler(component)
{
    customListBox.setScrollPosition(component.getScrollPosition());
}
```

Se nell'esempio precedente la funzione `myHandler` si fosse trovata nella linea temporale principale di terzo livello della linea temporale del componente, la prima riga del codice sarebbe stata la seguente:

```
scroll1.setChangeHandler("myHandler", _parent._parent._parent);
```

Il codice seguente crea la funzione `myHandler` in un'istanza di `myObject` (che appartiene alla classe `Object`) e specifica `myHandler` come funzione per `scroll11`.

```
myObject = new Object();
myObject.myHandler = function(component){
    customListBox.setScrollPosition(component.getScrollPosition());
}

scroll11.setChangeHandler("myHandler", myObject);
```

Vedere anche

`FScrollBar.getScrollPosition`, `FScrollBar.setScrollPosition`,
`FScrollBar.setScrollProperties`

FScrollBar.setEnabled

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setEnabled(attiva)
```

Parametri

attiva Valore booleano che specifica se la barra di scorrimento è attivata (`true`) o disattivata (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; determina se la barra di scorrimento è attivata (`true`) o disattivata (`false`). Se una barra di scorrimento viene disattivata, non accetta alcuna interazione con il mouse o la tastiera e non è più disponibile per l'utente. Se questo metodo viene richiamato senza alcun parametro, viene usato il valore `true`.

Esempio

Il codice seguente disattiva la barra di scorrimento `scroll12`.

```
scroll12.setEnabled(false);
```

Vedere anche

`FScrollBar.setEnabled`

FScrollBar.setHorizontal

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setHorizontal(scorrimentoOrizzontale)
```

Parametri

scorrimentoOrizzontale Valore booleano che specifica se la barra di scorrimento è orizzontale (`true`) o verticale (`false`).

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se la barra di scorrimento viene applicata al target orizzontalmente (`true`) o verticalmente (`false`). Questo metodo imposta il valore predefinito su `false`.

Esempio

Il codice seguente specifica che la barra di scorrimento `scrollText` è applicata orizzontalmente al suo target.

```
scrollText.setHorizontal(true);
```

Vedere anche

`FScrollBar.setSize`

FScrollBar.setLargeScroll

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setLargeScroll(largeScroll)
```

Parametri

largeScroll Numero intero che specifica il numero di posizioni da scorrere quando si fa clic una volta sulla traccia. L'impostazione predefinita è il valore impostato per `pageSize` con `FScrollBar.setScrollProperties`.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la proprietà `largeScroll` dell'istanza della barra di scorrimento in fase di esecuzione. Quando si fa clic una volta sulla traccia di scorrimento, la casella di scorrimento sposta la distanza specificata di una posizione `largeScroll`.

Esempio

Il codice seguente determina lo scorrimento di `scrollText1` di 20 posizioni quando si fa clic sulla traccia.

```
scrollText1.setLargeScroll(20);
```

Vedere anche

`FScrollBar.setSmallScroll`

FScrollBar.setScrollContent

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setScrollContent(target)
```

Parametri

target Riferimento al campo di testo per la barra di scorrimento.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica l'istanza del campo di testo cui è associata la barra di scorrimento. È necessario definire questa istanza nella stessa linea temporale e sullo stesso livello della barra di scorrimento. Se si richiama questo metodo, viene sovrascritto il valore del parametro campo di testo *target* impostato durante la creazione del codice. Il passaggio di `undefined` come valore per il parametro *target* annulla l'associazione della barra di scorrimento al campo di testo.

Esempio

Il codice seguente associa `scrollText1` al campo di testo con il nome istanza `textField1`.

```
scrollText1.setScrollContent("textField1");
```

FScrollBar.setScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setScrollPosition(posizione)
```

Parametri

posizione Numero intero compreso tra i valori `minPos` e `maxPos` della barra di scorrimento. Per ulteriori informazioni sull'impostazione delle proprietà `minPos` e `maxPos`, consultare `FScrollBar.setScrollProperties`.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica la posizione della casella di scorrimento sulla relativa barra ed esegue la funzione del gestore delle modifiche specificata con `FScrollBar.setChangeHandler`.

Esempio

Il codice seguente imposta la posizione della casella di scorrimento per `scrollText1` su 5.

```
scrollText1.setScrollPosition(5);
```

Per un altro esempio dell'uso di questo metodo, consultare `FScrollBar.setChangeHandler`.

Vedere anche

`FScrollBar.setChangeHandler`, `FScrollBar.setScrollProperties`

FScrollBar.setScrollProperties

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setScrollProperties(pageSize, minPos, maxPos)
```

Parametri

pageSize Numero intero che rappresenta il numero delle posizioni visualizzate nella pagina.

minPos Numero intero che rappresenta la posizione minima di scorrimento.

maxPos Numero intero che rappresenta la posizione massima di scorrimento.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica le proprietà *pageSize*, *minPos* e *maxPos* della barra di scorrimento e imposta la dimensione appropriata per la casella di scorrimento.

Questo metodo è indirizzato a utenti esperti e programmatori che desiderano creare componenti personalizzati; non è possibile usare questo metodo con barre di scorrimento associate a campi di testo. Quando una barra di scorrimento viene associata a un campo di testo, le proprietà di scorrimento vengono automaticamente impostate in base alle proprietà del campo di testo. Se si richiama questo metodo, viene annullata l'associazione della barra di scorrimento al campo di testo.

Esempio

Il codice seguente imposta le proprietà *pageSize*, *minPos* e *maxPos* per una barra di scorrimento associata a un componente personalizzato Casella di riepilogo. La casella di riepilogo ha 5 righe visibili e un totale di 20 voci di elenco. Poiché la casella è indicizzata da 0 a 19, la proprietà *maxPos* è data dal numero totale di voci presenti nella casella meno il numero di voci visibili.

```
scrollBar.setScrollProperties(5, 0, 15);
```

Vedere anche

FScrollBar.setScrollPosition

FScrollBar.setScrollTarget

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setScrollContent(target)
```

Parametri

target Riferimento al campo di testo per la barra di scorrimento.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica l'istanza del campo di testo cui è associata la barra di scorrimento. È necessario definire questa istanza nella stessa linea temporale e sullo stesso livello della barra di scorrimento. Se si richiama questo metodo, viene sovrascritto il valore del parametro campo di testo `target` impostato durante la creazione del codice. Il passaggio di `undefined` come valore per il parametro `target` annulla l'associazione della barra di scorrimento al campo di testo.

Esempio

Il codice seguente associa `scrollText1` al campo di testo con il nome istanza `textField1`.

```
scrollText1.setScrollTarget("textField1");
```

FScrollBar.setSize

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setSize(lunghezza)
```

Parametri

lunghezza Numero intero che specifica la lunghezza della barra di scorrimento in pixel.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la lunghezza, in pixel, della barra di scorrimento in fase di esecuzione (non è possibile impostare la larghezza dei componenti Barra di scorrimento). Se si richiama questo metodo, vengono sovrascritte tutte le impostazioni di ridimensionamento in scala e rotazione applicate durante la creazione del codice.

Questo metodo non deve essere usato con barre di scorrimento associate a campi di testo; la barra di scorrimento assume automaticamente la dimensione del campo di testo specificata durante la creazione del codice.

Per ulteriori informazioni, consultare "Ridimensionamento dei componenti Barra di scorrimento" nel capitolo "Uso dei componenti" della *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la lunghezza di `scrollText1` su 200 pixel.

```
scrollText1.setSize(200);
```

FScrollBar.setSmallScroll

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setSmallScroll(smallScroll)
```

Parametri

smallScroll Numero intero che specifica il numero di posizioni da scorrere quando si fa clic su una freccia di scorrimento. Il valore predefinito è 1.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la proprietà `smallScroll` dell'istanza della barra di scorrimento in fase di esecuzione, se il campo di testo è attivato. Quando si fa clic sulle frecce della barra di scorrimento o su un tasto freccia sulla tastiera, la casella di scorrimento sposta la distanza specificata di una posizione `smallScroll`.

Esempio

Il codice seguente determina lo scorrimento di `scrollText1` di 5 posizioni quando si fa clic su una freccia di scorrimento.

```
scrollText1.setSmallScroll(5);
```

Vedere anche

`FScrollBar.setLargeScroll`

FScrollBar.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto `FStyleFormat`.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di `FStyleFormat` per una singola istanza della barra di scorrimento. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di `FStyleFormat` per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la proprietà `arrow` di `scrollBar1` su `0x000000` (nero).

```
scrollBar1.setStyleProperty("arrow", 0x000000);
```

Vedere anche

`FStyleFormat` (oggetto)

FScrollPane (componente)

Il componente Riquadro di scorrimento nell'ambiente di creazione di Flash fornisce funzionalità di trascinamento della selezione che consentono di aggiungere riquadri di scorrimento per visualizzare clip filmato nei documenti Flash; inoltre, offre un'interfaccia utente per l'impostazione dei parametri di base. I metodi del componente FScrollPane consentono di controllare i riquadri di scorrimento in fase di esecuzione: è possibile creare riquadri di scorrimento, controllare i riquadri di scorrimento realizzati durante la creazione, impostare o ridefinire i parametri di base e impostare opzioni di runtime aggiuntive. Non è necessario usare una funzione di costruzione per accedere ai metodi dei componenti.

Il componente Riquadro di scorrimento offre barre di scorrimento verticali e orizzontali che consentono di visualizzare clip filmato di notevoli dimensioni senza occupare troppo spazio sullo stage. I controlli della tastiera e il mouse standard sono incorporati.

Nota: il componente Riquadro di scorrimento visualizza solo clip filmato; per aggiungere barre di scorrimento ai campi di testo dinamici o di input, usare il componente Barra di scorrimento. Il componente Riquadro di scorrimento non consente di visualizzare testi che usano caratteri dispositivo.

I metodi dei componenti non eseguono un controllo degli errori per tipo, come avviene con altre azioni e oggetti nativi di ActionScript; pertanto, si consiglia di convalidare i parametri prima di passarli ai metodi.

Il componente Riquadro di scorrimento è supportato in Flash Player 6 e versioni successive.

Per informazioni sull'uso del componente Pannello di scorrimento, sull'impostazione dei parametri durante la creazione di codice e sulla modifica del colore e dell'aspetto dei componenti, consultare le sezioni "Personalizzazione dei colori e del testo del componente" e "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi validi per il componente FScrollPane

Metodo	Descrizione
<code>FScrollPane.getPaneHeight</code>	Restituisce l'altezza del riquadro di scorrimento.
<code>FScrollPane.getPaneWidth</code>	Restituisce la larghezza del riquadro di scorrimento.
<code>FScrollPane.getScrollContent</code>	Restituisce un'istanza del contenuto visualizzato nel riquadro di scorrimento.
<code>FScrollPane.getScrollPosition</code>	Restituisce le coordinate x e y della posizione di scorrimento attuale.
<code>FScrollPane.loadScrollContent</code>	Carica un file SWF o JPEG nel riquadro di scorrimento.
<code>FScrollPane.refreshPane</code>	Ridimensiona le barre sul riquadro di scorrimento quando la dimensione del contenuto viene modificata.
<code>FScrollPane.registerSkinElement</code>	Registra un elemento skin in una proprietà definita per un elemento skin presente nel livello Leggimi e posizionato sul Fotogramma 1 di un clip filmato contenuto nella libreria.
<code>FScrollPane.setDragContent</code>	Imposta il contenuto del riquadro di scorrimento come trascinabile.
<code>FScrollPane.setHScroll</code>	Imposta lo stile dello scorrimento orizzontale per il riquadro di scorrimento.
<code>FScrollPane.setScrollContent</code>	Imposta un clip filmato come target del riquadro di scorrimento.

Metodo	Descrizione
<code>FScrollPane.setScrollPosition</code>	Determina lo scorrimento del riquadro alle coordinate x, y specificate.
<code>FScrollPane.setSize</code>	Imposta la larghezza e l'altezza del riquadro di scorrimento in pixel.
<code>FScrollPane.setStyleProperty</code>	Imposta una sola proprietà di stile per un componente.
<code>FScrollPane.setVerticalScroll</code>	Imposta lo stile dello scorrimento verticale per il riquadro di scorrimento.

FScrollPane.getPaneHeight

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.getPaneHeight()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che specifica l'altezza della visualizzazione del riquadro di scorrimento.

Descrizione

Metodo; restituisce l'altezza di visualizzazione del riquadro di scorrimento. È possibile usare questo metodo solo per ottenere l'altezza di un riquadro di scorrimento ridimensionato con `FScrollPane.setSize`. Questo metodo funziona solo se il riquadro di scorrimento è stato ridimensionato con `FScrollPane.setSize`, non se la dimensione è stata impostata usando le proprietà `_width` e `_height`.

Esempio

Il codice seguente ricava l'altezza e la larghezza del riquadro di scorrimento `display1` e usa i valori restituiti per ridimensionare il riquadro di scorrimento con `FScrollPane.setSize`.

```
var h = display1.getPaneHeight();  
var w = display1.getPaneWidth();  
display1.setSize(w+10, h+10);
```

Vedere anche

`FScrollPane.getPaneWidth`, `FScrollPane.setSize`

FScrollPane.getPaneWidth

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.getPaneWidth()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che specifica la larghezza della visualizzazione del riquadro di scorrimento.

Descrizione

Metodo; restituisce la larghezza della visualizzazione del riquadro di scorrimento. È possibile usare questo metodo solo per ottenere la larghezza di un riquadro di scorrimento ridimensionato con `FScrollPane.setSize`. Questo metodo funziona solo se il riquadro di scorrimento è stato ridimensionato con `FScrollPane.setSize`, non se la dimensione è stata impostata usando le proprietà `_width` e `_height`.

Esempio

Il codice seguente ricava l'altezza e la larghezza del riquadro di scorrimento `display1` e usa i valori restituiti per ridimensionare il riquadro di scorrimento con `FScrollPane.setSize`.

```
var h = display1.getPaneHeight();
var w = display1.getPaneWidth();
display1.setSize(w+10, h+10);
```

Vedere anche

`FScrollPane.getPaneHeight`, `FScrollPane.setSize`

FScrollPane.getScrollContent

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.getScrollContent()
```

Parametri

Nessuno.

Valori restituiti

Un riferimento al clip filmato nel riquadro di scorrimento.

Descrizione

Metodo; restituisce un'istanza del contenuto visualizzato nel riquadro di scorrimento.

Esempio

Il codice seguente recupera un riferimento al clip filmato contenuto all'interno di `display1`, lo memorizza in una variabile e specifica al clip filmato di passare al fotogramma 4.

```
var content = display1.getScrollContent();
content.gotoAndStop(4);
```

Vedere anche

`FScrollPane.setScrollContent`

FScrollPane.getScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.getScrollPosition()
```

Parametri

Nessuno.

Valori restituiti

Un oggetto.

Descrizione

Metodo; restituisce un oggetto con i campi `.x` e `.y` che specificano la posizione di scorrimento attuale, verticale o orizzontale, per la visualizzazione del riquadro di scorrimento.

Esempio

Il codice seguente restituisce la posizione di scorrimento attuale del riquadro di scorrimento `scroll12`.

```
trace(scroll12.getScrollPosition());
```

Vedere anche

`FScrollPane.setScrollPosition`

FScrollPane.loadScrollContent

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.loadScrollContent(URL [, nomeFunz, posizione])
```

Parametri

URL Stringa che specifica l'URL di un file SWF o JPEG da caricare nel riquadro di scorrimento.

nomeFunz Stringa che specifica il nome della funzione del gestore da eseguire quando viene caricato il contenuto del riquadro di scorrimento. Se il parametro *posizione* non viene specificato, questa funzione deve trovarsi nella stessa linea temporale dell'istanza del componente.

posizione Percorso di riferimento per un oggetto costituito da dati, clip filmato o da una linea temporale contenente la funzione specificata. Questo parametro è opzionale e usa la linea temporale principale del componente come impostazione predefinita.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica l'URL di un file SWF o JPEG da visualizzare nel riquadro di scorrimento. I parametri opzionali *nomeFunz* e *posizione* consentono di specificare una funzione del gestore delle modifiche da richiamare quando il contenuto viene caricato.

Tale URL deve trovarsi nello stesso sottodominio dell'URL che contiene il filmato Flash. Per usare file SWF o JPEG in Flash Player o provare il filmato nell'ambiente di creazione di Flash, è necessario memorizzare tutti i file SWF o JPEG nella stessa cartella assegnando loro nomi di file non contenenti alcun riferimento a specifiche della cartella o dell'unità disco.

Se si richiama questo metodo, viene sovrascritto il valore del parametro di scorrimento del contenuto impostato durante la creazione del codice.

Per ulteriori informazioni ed esempi sull'uso delle funzioni del gestore delle modifiche, consultare `FScrollBar.setChangeHandler`.

Esempio

Il codice seguente carica un file JPEG posizionato su un server in `display1`.

```
display1.loadScrollContent("http://www.browserWeb.com/Nice.jpg");
```

Il codice seguente carica un file JPEG posizionato su un server e specifica la funzione del gestore delle modifiche `load` posizionata nella linea temporale principale di secondo livello del componente `display1`.

```
display1.loadScrollContent("http://www.browserWeb.com/Nice.jpg" , "load" ,
    _parent._parent);
function load(component){
    //Il contenuto viene caricato
    component.setScrollPostion(10,10);
}
```

Vedere anche

`FScrollPane.getPaneHeight`, `FScrollPane.setScrollContent`

FScrollPane.refreshPane

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.refreshPane()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; ridimensiona le barre del riquadro di scorrimento quando si modifica il contenuto all'interno di questo riquadro. Richiamare questo metodo se il contenuto del riquadro di scorrimento viene ridimensionato usando le proprietà `_width` o `_height`.

Esempio

Il codice seguente aggiorna le barre di scorrimento di `moviePane` dopo aver aumentato la dimensione del clip filmato `myContent` nel riquadro di scorrimento.

```
var myContent = moviePane.getScrollContent();
myContent._width = 400;
moviePane.refreshPane();
```

Vedere anche

`FScrollPane.getScrollContent`

FScrollPane.registerSkinElement

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.registerSkinElement(elemento, proprietàStile)
```

Parametri

elemento Un'istanza del clip filmato.

proprietàStile Il nome di una proprietà di FStyleFormat.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un elemento skin in una proprietà di stile. Gli elementi skin vengono registrati nelle proprietà nel primo fotogramma del livello Leggimi di ciascun elemento skin contenuto nella libreria.

I componenti sono formati da elementi skin, ognuno dei quali composto da più elementi skin, registrabili individualmente in una proprietà dell'oggetto FStyleFormat. I valori di queste proprietà sono determinati dal formato di stile assegnato a un componente. Per impostazione predefinita, i componenti dell'interfaccia utente di Flash vengono assegnati all'oggetto `globalStyleFormat`, che è un'istanza dell'oggetto FStyleFormat.

Usare questo metodo per registrare proprietà ed elementi skin personalizzati nell'interfaccia utente di Flash o per personalizzare gli elementi skin di un componente modificando il codice nel primo fotogramma del livello Leggimi di un elemento skin nella libreria.

Una volta aggiunto al documento Flash, il componente FScrollPane usa gli elementi skin contenuti nella cartella FScrollBar Skins e l'elemento skin FLabel contenuto nella cartella Global Skins. La modifica di un elemento skin contenuto nella cartella FScrollBar Skins si applica a tutti i componenti che usano le barre di scorrimento (Casella combinata, Casella di riepilogo, Barra di scorrimento e Riquadro di scorrimento).

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente registra l'elemento skin personalizzato `NewArrow_mc` nella proprietà `arrow` nel primo fotogramma del livello Leggimi dell'elemento skin `fsb_downArrow`, contenuto nella cartella FScrollBar Skins presente nella libreria.

```
Panel1.registerSkinElement(NewArrow_mc, "arrow");
```

Vedere anche

FStyleFormat (oggetto)

FScrollPane.setDragContent

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setDragContent(trascinamento)
```

Parametri

trascinamento Valore booleano; `true` specifica che l'utente può modificare la visualizzazione trascinando il contenuto nel riquadro di scorrimento; `false` specifica che l'utente può modificare la visualizzazione solo usando le barre di scorrimento.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica se l'utente può modificare il riquadro di scorrimento trascinando il relativo contenuto oltre che usando le barre di scorrimento. Se si richiama questo metodo, viene sovrascritto il valore del parametro di trascinamento del contenuto impostato durante la creazione del codice.

Esempio

L'esempio seguente specifica che è possibile trascinare il contenuto del riquadro di scorrimento `display1`.

```
display1.setDragContent(true);
```

FScrollPane.setHScroll

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setHScroll(visualizzazione)
```

Parametri

visualizzazione Valore booleano che specifica se la barra di scorrimento viene visualizzata sempre (`true`) o mai (`false`) oppure una stringa che specifica che la barra di scorrimento viene visualizzata solo se necessario ("`auto`").

Se si richiama questo metodo, viene sovrascritto il valore del parametro di scorrimento orizzontale impostato durante la creazione del codice.

Valori restituiti

Nessuno.

Descrizione

Metodo; determina se una barra di scorrimento orizzontale viene visualizzata sempre (`true`), mai (`false`) o se necessario ("`auto`"). Il valore predefinito è `auto`.

Esempio

Il codice seguente nasconde la barra di scorrimento orizzontale per `display1`.

```
display1.setHScroll(false);
```

Vedere anche

`FScrollPane.setVScroll`

FScrollPane.setScrollContent

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setScrollContent(target)
```

Parametri

target Una stringa di testo che specifica l'ID di concatenamento del simbolo di un clip filmato nella libreria o un'istanza di un clip filmato.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un clip filmato da visualizzare nel riquadro di scorrimento. Se si richiama questo metodo, viene sovrascritto il valore del parametro di scorrimento del contenuto impostato durante la creazione del codice.

Esempio

L'esempio seguente specifica l'istanza del clip filmato `BetsyTacy` come `target` per `display1`.

```
display1.setScrollContent("BetsyTacy");
```

Vedere anche

`FScrollPane.getPaneHeight`, `FScrollPane.loadScrollContent`

FScrollPane.setScrollPosition

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setScrollPosition(x, y)
```

Parametri

x Numero intero che specifica il numero di pixel (da 0) da scorrere verso destra.

y Numero intero che specifica il numero di pixel (da 0) da scorrere verso il basso.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la posizione di scorrimento sulla posizione delle coordinate *x*, *y* specificate.

Esempio

L'esempio seguente scorre il contenuto di `display1` di 14 pixel verso il basso e 40 pixel verso destra.

```
display1.setScrollPosition(14,40);
```

Vedere anche

`FScrollPane.getScrollPosition`

FScrollPane.setSize

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setSize(larghezza, altezza)
```

Parametri

larghezza Numero intero che specifica la larghezza del riquadro di scorrimento in pixel.

altezza Numero intero che specifica l'altezza del riquadro di scorrimento in pixel.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta in pixel la larghezza e l'altezza di visualizzazione del riquadro di scorrimento in fase di esecuzione. Se si richiama questo metodo, vengono sovrascritte le dimensioni applicate durante la creazione del codice.

Per ulteriori informazioni, consultare la sezione "Personalizzazione degli elementi skin del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la larghezza e l'altezza di `display1` su 500 x 300 pixel.

```
display1.setSize(500, 300);
```

Vedere anche

`FScrollPane.getPaneHeight`, `FScrollPane.getPaneWidth`

FScrollPane.setStyleProperty

Disponibilità

Flash Player 6.

Uso

```
myScrollPane.setStyleProperty(proprietàStile, valore)
```

Parametri

proprietàStile Stringa che specifica una proprietà dell'oggetto `FStyleFormat`.

valore Il valore da impostare per la proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta una proprietà di `FStyleFormat` per una singola istanza del riquadro di scorrimento. Se si richiama questo metodo per specificare una proprietà, viene sovrascritta l'impostazione della proprietà nel formato di stile assegnato al componente. Se alla proprietà viene passato il valore `undefined`, tutti gli stili associati a quella proprietà vengono rimossi.

Per impostare le proprietà di `FStyleFormat` per più componenti, creare un formato di stile personalizzato. Per ulteriori informazioni sulla creazione di formati di stile personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice riportato di seguito imposta la proprietà `arrow` di `ScrollPane2` su `0x000000` (nero).

```
ScrollPane2.setStyleProperty("arrow", 0x000000);
```

Vedere anche

`FStyleFormat` (oggetto)

FScrollPane.setVScroll

Disponibilità

Flash Player 6.

Uso

```
myScrollBar.setVScroll(display)
```

Parametri

display Un valore booleano che specifica se una barra di scorrimento verticale viene visualizzata sempre (`true`) o mai (`false`) oppure una stringa che specifica che la barra di scorrimento viene visualizzata solo se necessario ("`auto`").

Valori restituiti

Nessuno.

Descrizione

Metodo; determina se una barra di scorrimento verticale viene visualizzata sempre (`true`), mai (`false`) o solo se necessario ("`auto`"). Il valore predefinito è `auto`.

Se si richiama questo metodo, viene sovrascritto il valore del parametro di scorrimento verticale impostato durante la creazione del codice.

Esempio

Il codice seguente specifica che la barra di scorrimento verticale per `display1` viene sempre visualizzata.

```
display1.setVScroll(true);
```

Vedere anche

`FScrollPane.setHScroll`

FStyleFormat (oggetto)

L'oggetto `FStyleFormat` consente di modificare o impostare le proprietà del formato di stile globale, assegnato a tutti i componenti dell'interfaccia utente di Flash per impostazione predefinita, o di creare nuovi formati personalizzati da usare con tali componenti o con componenti personalizzati creati o acquisiti da altre fonti. Il formato di stile globale, ossia l'oggetto `globalStyleFormat`, è un'istanza dell'oggetto `FStyleFormat` che definisce le proprietà di formattazione del colore e del testo usate per visualizzare tutti i componenti dell'interfaccia utente di Flash.

Per creare un nuovo formato di stile personalizzato, è sufficiente creare una nuova istanza dell'oggetto `FStyleFormat` usando la funzione di costruzione `new FStyleFormat()`, impostare le proprietà `FStyleFormat` che si desidera includere nel formato personalizzato e usare il metodo `FStyleFormat.addListener` per registrare le istanze dei componenti nel nuovo formato di stile. L'istanza di un componente è in grado di rilevare più formati di stile, ma può assumere il valore di un solo formato di stile per una proprietà specifica. Se a un formato di stile si aggiunge un componente con funzione di listener, tale componente usa il nuovo formato per le proprietà in esso specificate e il formato precedente per tutte le altre proprietà.

Non è necessario usare la funzione di costruzione `FStyleFormat` per aggiungere o rimuovere i listener oppure per impostare o modificare le proprietà nel formato di stile globale, poiché l'oggetto `globalStyleFormat` viene creato solo dopo che tutti i componenti dell'interfaccia utente di Flash sono stati posizionati nello stage.

È possibile impostare le proprietà `FStyleFormat` per una singola istanza di un componente usando il metodo `setStyleProperty` disponibile per tutti i componenti dell'interfaccia utente di Flash. Il metodo `setStyleProperty` consente di impostare una proprietà per un componente senza dover creare un'istanza dell'oggetto `FStyleFormat`. L'uso del metodo `setStyleProperty` sovrascrive l'impostazione relativa a una proprietà specifica di un formato di stile assegnata al componente senza modificare le altre impostazioni delle proprietà. Per ulteriori informazioni, consultare le voci di `setStyleProperty` relative ai singoli componenti.

Quando si assegna il valore di un colore a una proprietà `FStyleFormat`, occorre specificare un colore RGB nel formato `0xRRGGBB`.

Per ulteriori informazioni sul formato di stile globale e sulla creazione di formati personalizzati, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Riepilogo dei metodi validi per l'oggetto `FStyleFormat`

Metodo	Descrizione
<code>FStyleFormat.addListener</code>	Registra un componente in un formato di stile.
<code>FStyleFormat.applyChanges</code>	Applica le modifiche apportate ai valori della proprietà in un formato di stile.
<code>FStyleFormat.removeListener</code>	Rimuove un componente con funzione di listener da un formato di stile.

Riepilogo delle proprietà valide per l'oggetto FStyleFormat

Nella tabella riportata di seguito è elencato il riepilogo delle proprietà per l'oggetto FStyleFormat.

Proprietà	Descrizione
<code>FStyleFormat.arrow</code>	Il colore della freccia usata nelle barre di scorrimento e nelle caselle di riepilogo a discesa.
<code>FStyleFormat.background</code>	Il colore dello sfondo di un componente.
<code>FStyleFormat.backgroundDisabled</code>	Il colore dello sfondo di un componente disattivato.
<code>FStyleFormat.check</code>	Il colore del segno di spunta in una casella di controllo selezionata.
<code>FStyleFormat.darkshadow</code>	Il colore del bordo interno o dell'ombreggiatura scura di un componente.
<code>FStyleFormat.face</code>	Il colore principale del componente.
<code>FStyleFormat.foregroundDisabled</code>	Il colore di primo piano di un componente disattivato.
<code>FStyleFormat.highlight</code>	Il colore del bordo interno o dell'ombreggiatura scura di un componente selezionato.
<code>FStyleFormat.highlight3D</code>	Il colore del bordo esterno o dell'ombreggiatura chiara di un componente selezionato.
<code>FStyleFormat.radioDot</code>	Il colore del puntino di un pulsante di scelta selezionato.
<code>FStyleFormat.scrollTrack</code>	Il colore della traccia in una barra di scorrimento.
<code>FStyleFormat.selection</code>	Il colore della barra di selezione che evidenzia la voce di un elenco in un componente.
<code>FStyleFormat.selectionDisabled</code>	Il colore della barra di selezione che evidenzia la voce di un elenco in un componente disattivato.
<code>FStyleFormat.selectionUnfocused</code>	Il colore della barra di selezione quando il componente non è attivato dalla tastiera.
<code>FStyleFormat.shadow</code>	Il colore del bordo esterno o dell'ombreggiatura chiara di un componente.
<code>FStyleFormat.textAlign</code>	L'allineamento (a destra, sinistra o al centro) del testo visualizzato in un componente o sopra di esso.
<code>FStyleFormat.textBold</code>	Specifica se il testo è in grassetto (<code>true</code>) o meno (<code>false</code>).
<code>FStyleFormat.textColor</code>	Il colore di testo predefinito in tutti i componenti assegnati al formato di stile.
<code>FStyleFormat.textDisabled</code>	Il colore del testo in un componente disattivato.
<code>FStyleFormat.textFont</code>	Il nome del tipo di carattere nel quale viene visualizzato il testo.
<code>FStyleFormat.textIndent</code>	Il rientro del testo dal margine sinistro al primo carattere di testo in pixel.
<code>FStyleFormat.textItalic</code>	Specifica se il testo è in corsivo (<code>true</code>) o meno (<code>false</code>).
<code>FStyleFormat.textLeftMargin</code>	Il margine sinistro del paragrafo per il testo in pixel.
<code>FStyleFormat.textRightMargin</code>	Il margine destro del paragrafo per il testo in pixel.
<code>FStyleFormat.textSelected</code>	Il colore della voce di elenco selezionata in un componente.
<code>FStyleFormat.textSize</code>	La dimensione del testo in punti.
<code>FStyleFormat.textUnderline</code>	Specifica se il testo è sottolineato (<code>true</code>) o meno (<code>false</code>).

Funzione di costruzione per l'oggetto FStyleFormat

Disponibilità

Flash Player 6.

Uso

```
new FStyleFormat()
```

Parametri

Nessuno.

Valori restituiti

Un'istanza dell'oggetto FStyleFormat.

Descrizione

Metodo; crea un nuovo oggetto FStyleFormat. È possibile creare nuovi oggetti FStyleFormat per definire le proprietà del testo e del colore per i formati di stile personalizzati usati con i componenti personalizzati o con i componenti dell'interfaccia utente di Flash. Tutti i componenti dell'interfaccia utente di Flash vengono assegnati per impostazione predefinita a un'istanza dell'oggetto FStyleFormat, `globalStyleFormat`. Non è necessario creare una nuova istanza dell'oggetto FStyleFormat per modificare le proprietà nel formato di stile globale perché l'istanza già esiste. È possibile inoltre usare il metodo `setStyleProperty` per modificare le proprietà di istanze specifiche di un componente senza usare la funzione di costruzione.

Per ulteriori informazioni, consultare la voce relativa al metodo `setStyleProperty` disponibile per ciascun componente: `FCheckBox.setStyleProperty`, `FComboBox.setStyleProperty`, ecc. Consultare anche la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

L'esempio seguente crea il nuovo formato di stile `formStyleFormat`.

```
formStyleFormat = new StyleFormat();
```

FStyleFormat.addListener

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.addListener(componente1 [, componente2, ...componenteN])
```

Parametri

componente1 ... *componenteN* Le istanze del componente da registrare in *myStyleFormat*.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra i componenti specificati in *myStyleFormat*. Usare questo metodo per registrare le istanze dei componenti dell'interfaccia utente di Flash o i componenti personalizzati in un formato di stile personalizzato. È possibile inoltre usare questo metodo con la sintassi riportata di seguito per registrare un componente personalizzato nel formato di stile globale usato da tutti i componenti dell'interfaccia utente di Flash per impostazione predefinita:

```
globalStyleFormat.addListener(componentePersonalizzato);
```

Esempio

Il codice seguente registra *formStyleFormat* con i componenti *myListBox*, *myComboBox* e *myScrollBar*.

```
formStyleFormat.addListener(myListBox, myComboBox, myScrollBar);
```

Vedere anche

FStyleFormat.applyChanges, *FStyleFormat.removeListener*

FStyleFormat.applyChanges

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.applyChanges([nomeProprietà1, ...nomeProprietàN])  
myStyleFormat.applyChanges()
```

Parametri

nomeProprietà1...nomeProprietàN Una serie di stringhe di testo che specificano le proprietà da aggiornare per tutti i componenti assegnati a *myStyleFormat*.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiorna l'istanza dell'oggetto nel formato di stile specificato e applica le modifiche a tutti i componenti assegnati al formato. È necessario richiamare questo metodo quando si aggiungono o rimuovono i listener oppure quando si modificano le proprietà. Durante l'aggiornamento delle proprietà, si consiglia di applicare il primo uso della sintassi in modo da aggiornare solo quelle proprietà per le quali si specifica un nuovo valore.

Uso 1: aggiorna solo le proprietà specificate nei parametri.

Uso 2: aggiorna tutte le informazioni nel formato di stile, ossia le proprietà e i componenti assegnati, sia che siano stati modificati o meno.

Esempio

Uso 1: l'esempio seguente aggiorna le proprietà *arrow* e *background*, ma non le proprietà *check* e *highlight* in *formStyleFormat*.

```
formStyleFormat.arrow = 0x00ffaa;  
formStyleFormat.background = 0xaabbcc;  
formStyleFormat.check = 0x000000;  
formStyleFormat.highlight = 0xffffffff;  
formStyleFormat.applyChanges("arrow", "background");
```

Usò 2: l'esempio seguente aggiorna tutte le proprietà in `formStyleFormat`: `arrow`, `background`, `check` e `highlight`.

```
formStyleFormat.arrow = 0x00ffaa;  
formStyleFormat.background = 0xaabbcc;  
formStyleFormat.check = 0x000000;  
formStyleFormat.highlight = 0xffffffff;  
formStyleFormat.applyChanges();
```

Vedere anche

`FStyleFormat.addListener`, `FStyleFormat.removeListener`

FStyleFormat.arrow

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.arrow
```

Descrizione

Proprietà; il valore del colore RGB per la proprietà `arrow` usato nelle barre di scorrimento e nelle caselle di riepilogo a discesa in componenti come barre di scorrimento, caselle di riepilogo e caselle combinate. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di `0x800080` alla proprietà `arrow` in `formStyleFormat`, producendo una freccia di colore viola.

```
formStyleFormat.arrow = 0x800080;
```

FStyleFormat.background

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.background
```

Descrizione

Proprietà; il valore del colore RGB per lo sfondo di un componente. Ad esempio, in un pulsante di scelta o una casella di controllo, lo sfondo è lo spazio racchiuso all'interno dell'area di selezione; in una casella di riepilogo o una casella combinata, invece, lo sfondo è l'area di visualizzazione. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di 0xFFE4E1 alla proprietà `background` per `formStyleFormat`, producendo uno sfondo di colore rosa pallido quando il componente viene attivato.

```
formStyleFormat.background = 0xFFE4E1;
```

Vedere anche

`FStyleFormat.face`

FStyleFormat.backgroundDisabled

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.backgroundDisabled
```

Descrizione

Proprietà; il valore del colore RGB per lo sfondo di un componente disattivato. Il colore di sfondo per gli elementi disattivati dell'interfaccia utente è generalmente grigio chiaro. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di 0x808080 alla proprietà `backgroundDisabled` per `formStyleFormat`, producendo uno sfondo di colore grigio quando il componente viene disattivato.

```
formStyleFormat.backgroundDisabled = 0x808080;
```

Vedere anche

`FStyleFormat.foregroundDisabled`

FStyleFormat.check

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.check
```

Descrizione

Proprietà; il valore del colore RGB per il segno di spunta in una casella di controllo selezionata. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di 0x228B22 alla proprietà `check` per `formStyleFormat`, producendo una freccia di colore verde scuro.

```
formStyleFormat.check = 0x228B22;
```

FStyleFormat.darkshadow

Disponibilità

Flash Player 6.

Uso

myStyleFormat.darkshadow

Descrizione

Proprietà; il valore del colore RGB per il bordo interno o l'ombreggiatura scura di un componente, ad esempio, il bordo interno di un pulsante di scelta o una casella di controllo non selezionati. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di `0x0000CD` alla proprietà `darkshadow` per `formStyleFormat`, producendo un bordo interno di colore blu intermedio.

```
formStyleFormat.darkshadow = 0x0000CD;
```

Vedere anche

`FStyleFormat.highlight`, `FStyleFormat.shadow`

FStyleFormat.face

Disponibilità

Flash Player 6.

Uso

myStyleFormat.face

Descrizione

Proprietà; il valore relativo al colore RGB per il colore principale di un componente, ad esempio, il grigio usato per il componente Pulsante o Barra di scorrimento. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x32CD32` alla proprietà `face` per `formStyleFormat`, producendo pulsanti e barre di scorrimento di colore verde limone.

```
formStyleFormat.face = 0x32CD32;
```


FStyleFormat.foregroundDisabled

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.foregroundDisabled
```

Descrizione

Proprietà; il valore relativo al colore RGB per il primo piano di un componente disattivato. Il colore di primo piano per gli elementi disattivati dell'interfaccia utente è generalmente grigio intermedio. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore 0x708090 alla proprietà `foregroundDisabled` per `formStyleFormat`, producendo un primo piano di colore grigio per i componenti disattivati.

```
formStyleFormat.foregroundDisabled = 0x708090;
```

Vedere anche

`FStyleFormat.backgroundDisabled`

FStyleFormat.highlight

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.highlight
```

Descrizione

Proprietà; il valore relativo al colore RGB per il bordo interno o l'ombreggiatura scura di un componente selezionato, ad esempio il bordo interno di un pulsante di scelta o una casella di controllo. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore 0xFF00FF alla proprietà `highlight` per `formStyleFormat`, producendo un bordo interno di colore giallo quando il componente viene selezionato.

```
formStyleFormat.highlight = 0xFF00FF;
```

Vedere anche

`FStyleFormat.darkshadow`

FStyleFormat.highlight3D

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.highlight3D
```

Descrizione

Proprietà; il valore relativo al colore RGB per il bordo esterno o l'ombreggiatura chiara di un componente selezionato, ad esempio il bordo esterno di un pulsante di scelta o una casella di controllo. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x40E0D0` alla proprietà `highlight3D` per `formStyleFormat`, producendo un bordo esterno di colore turchese quando il componente viene selezionato.

```
formStyleFormat.highlight3D = 0x40E0D0;
```

Vedere anche

`FStyleFormat.shadow`

FStyleFormat.radioDot

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.radioDot
```

Descrizione

Proprietà; il valore relativo al colore RGB per il puntino di selezione di un pulsante di scelta in un componente. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0xFF12AC` alla proprietà `radioDot` per `formStyleFormat`, producendo un puntino di selezione di colore rosa.

```
formStyleFormat.radioDot = 0xFF12AC;
```

FStyleFormat.removeListener

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.removeListener(componente)
```

Parametri

componente Il componente da rimuovere dal formato di stile.

Descrizione

Metodo; rimuove un componente assegnato al formato di stile.

- Se si rimuove un componente dell'interfaccia utente di Flash con funzione di listener dal formato di stile globale senza assegnarlo, ossia aggiungerlo come listener, a un formato di stile personalizzato, i clip filmato dell'elemento skin vengono visualizzati come se fossero stati originariamente realizzati dal creatore di contenuti del componente senza un valore di proprietà assegnato.
- Se si rimuove un componente dell'interfaccia utente di Flash con funzione di listener da un formato di stile personalizzato, il componente non usa più i valori di proprietà nel formato di stile personalizzato, ma usa i valori specificati per le proprietà nell'oggetto del formato di stile globale.
- Se si rimuove un componente personalizzato con funzione di listener da un formato di stile personalizzato senza aggiungerlo a un nuovo formato personalizzato, il componente usa i valori impostati per le proprietà nel formato di stile globale, ove possibile, altrimenti visualizza gli elementi skin senza un valore di proprietà.

Esempio

L'esempio seguente rimuove il componente `check1` da `globalStyleFormat`.

```
globalStyleFormat.removeListener(check1);
```

Vedere anche

`FStyleFormat.addListener`, `FStyleFormat.applyChanges`

FStyleFormat.scrollTrack

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.scrollTrack
```

Descrizione

Proprietà; il valore del colore RGB per la traccia di una barra di scorrimento. Il componente Barra di scorrimento viene utilizzato dai componenti Riquadro di scorrimento, Casella di riepilogo e Casella combinata. Se si modifica il valore della proprietà `scrollTrack` nel formato di stile globale, viene modificato il colore della traccia in tutti i componenti che usano le barre di scorrimento. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna un valore di `0xA0522D` alla proprietà `scrollTrack` in `formStyleFormat`, producendo una traccia di scorrimento di colore bordeaux.

```
formStyleFormat.scrollTrack = 0xA0522D;
```

FStyleFormat.selection

Disponibilità

Flash Player 6.

Uso

myStyleFormat.selection

Descrizione

Proprietà; il valore del colore RGB per la barra usata per evidenziare la voce selezionata nell'elenco di un componente. Questa proprietà opera in combinazione con la proprietà

`FStyleFormat.textSelected` per visualizzare le voci selezionate e coordinare i colori in modo da facilitare la lettura del testo. Ad esempio, il formato di stile globale assegna come valore il colore blu alla proprietà `selection` usata per visualizzare la barra di selezione nei componenti Casella di riepilogo e Casella combinata e assegna come valore il colore bianco alla proprietà `textSelected`; questa combinazione di colori facilita la visualizzazione del testo selezionato da parte dell'utente. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x87CEEB` alla proprietà `selection` in `formStyleFormat`, producendo una barra di selezione di colore azzurro.

```
formStyleFormat.selection = 0x87CEEB;
```

Vedere anche

`FStyleFormat.textSelected`

FStyleFormat.selectionDisabled

Disponibilità

Flash Player 6.

Uso

myStyleFormat.selectionDisabled

Descrizione

Proprietà; il valore relativo al colore RGB per la barra di selezione usata per evidenziare una voce di elenco in un componente disattivato. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x708090` alla proprietà `selectionDisabled` per `formStyleFormat`, producendo una barra di selezione di colore grigio ardesia quando il componente è disattivato.

```
formStyleFormat.selectionDisabled = 0x708090;
```

Vedere anche

`FStyleFormat.selection`

FStyleFormat.selectionUnfocused

Disponibilità

Flash Player 6.

Uso

`myStyleFormat.selectionUnfocused`

Descrizione

Proprietà; il valore relativo al colore RGB per la barra di selezione (evidenziazione) nell'elenco di un componente quando il componente non è attivato dalla tastiera. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0xaabbcc` alla proprietà `selectionUnfocused` per `formStyleFormat`.

```
formStyleFormat.selectionUnfocused = 0xaabbcc;
```

Vedere anche

`FStyleFormat.selection`

FStyleFormat.shadow

Disponibilità

Flash Player 6.

Uso

`myStyleFormat.shadow`

Descrizione

Proprietà; il valore relativo al colore RGB per il bordo esterno o l'ombreggiatura chiara di un componente, ad esempio il bordo esterno di un pulsante di scelta o una casella di controllo non selezionati. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x008080` alla proprietà `shadow` per `formStyleFormat`, producendo un bordo esterno di colore verde acqua per i componenti Pulsante di scelta o Casella di controllo quando non sono selezionati.

```
formStyleFormat.shadow = 0x008080;
```

Vedere anche

`FStyleFormat.check`

FStyleFormat.textAlign

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.textAlign
```

Descrizione

Proprietà; una stringa di testo che specifica l'allineamento a destra, a sinistra o al centro per il testo visualizzato in tutti i componenti assegnati al formato di stile. L'impostazione predefinita è `left`.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente allinea a destra il testo visualizzato nei componenti tramite `formStyleFormat`.

```
formStyleFormat.textAlign = "right";
```

Vedere anche

`FStyleFormat.textIndent`, `FStyleFormat.textLeftMargin`,
`FStyleFormat.textRightMargin`

FStyleFormat.textBold

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.textBold
```

Descrizione

Proprietà; un valore booleano che specifica se il testo visualizzato nei componenti usando il formato di stile selezionato è in grassetto (`true`) o meno (`false`). L'impostazione predefinita è `false`.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica la formattazione in grassetto del testo visualizzato nei componenti assegnati a `formStyleFormat`.

```
formStyleFormat.textBold = true;
```

Vedere anche

`FStyleFormat.textItalic`, `FStyleFormat.textUnderline`

FStyleFormat.textColor

Disponibilità

Flash Player 6.

Uso

`myStyleFormat.textColor`

Descrizione

Proprietà; il valore relativo al colore RGB per il colore di testo predefinito in tutti i componenti assegnati al formato di stile. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0x000000` alla proprietà `textColor` in `formStyleFormat`, producendo testo di colore nero.

```
formStyleFormat.textColor = 0x000000;
```

Vedere anche

`FStyleFormat.textDisabled`, `FStyleFormat.textSelected`

FStyleFormat.textDisabled

Disponibilità

Flash Player 6.

Uso

`myStyleFormat.textDisabled`

Descrizione

Proprietà; il valore relativo al colore RGB per il colore di testo predefinito usato per visualizzare il testo in tutti i componenti disattivati assegnati al formato di stile. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0xC0C0C0` alla proprietà `textDisabled` per `formStyleFormat`, producendo testo di colore argento quando il componente è disattivato.

```
formStyleFormat.textDisabled = 0xC0C0C0;
```

Vedere anche

`FStyleFormat.textAlign`, `FStyleFormat.textSelected`

FStyleFormat.textFont

Disponibilità

Flash Player 6.

Uso

`myStyleFormat.textFont`

Descrizione

Proprietà; una stringa di testo che specifica il tipo di carattere usato per visualizzare il testo in tutti i componenti assegnati al formato di stile.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `Arial` alla proprietà `textFont` per `formStyleFormat`.

```
formStyleFormat.textFont = "Arial";
```

FStyleFormat.textIndent

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.textIndent
```

Descrizione

Proprietà; un numero intero che specifica il rientro, in pixel, dal margine sinistro al primo carattere per tutto il testo visualizzato usando il formato di stile.

Esempio

Il codice seguente rientra tutto il testo visualizzato tramite `formStyleFormat` di 5 pixel.

```
formStyleFormat.textIndent = 5;
```

Vedere anche

`FStyleFormat.textAlign`, `FStyleFormat.textLeftMargin`

FStyleFormat.textItalic

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.textItalic
```

Descrizione

Proprietà; un valore booleano che specifica se tutto il testo visualizzato nei componenti usando il formato di stile selezionato è in corsivo (`true`) o meno (`false`). L'impostazione predefinita è `false`.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica la formattazione in corsivo per tutto il testo visualizzato nei componenti assegnati a `formStyleFormat`.

```
formStyleFormat.textItalic = true;
```

Vedere anche

`FStyleFormat.textBold`

FStyleFormat.textLeftMargin

Disponibilità

Flash Player 6.

Uso

myStyleFormat.textLeftMargin

Descrizione

Proprietà; un numero intero che specifica il margine sinistro del paragrafo, in pixel, per tutto il testo visualizzato nei componenti assegnati al formato di stile.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica un valore di 4 pixel per la proprietà `textLeftMargin` di `formStyleFormat`.

```
formStyleFormat.textLeftMargin = 4;
```

Vedere anche

`FStyleFormat.textRightMargin`

FStyleFormat.textRightMargin

Disponibilità

Flash Player 6.

Uso

myStyleFormat.textRightMargin

Proprietà; un numero intero che specifica il margine destro del paragrafo, in pixel, per tutto il testo visualizzato nei componenti assegnati al formato di stile.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica un valore di 4 pixel per la proprietà `textRightMargin` di `formStyleFormat`.

```
formStyleFormat.textRightMargin = 4;
```

Vedere anche

`FStyleFormat.textLeftMargin`

FStyleFormat.textSelected

Disponibilità

Flash Player 6.

Uso

myStyleFormat.textSelected

Descrizione

Proprietà; un valore relativo al colore RGB che specifica il colore del testo selezionato nei componenti assegnati al formato di stile. Questa proprietà opera in combinazione con la proprietà `FStyleFormat.selection` per visualizzare le voci di elenco selezionate e coordinare i colori in modo da facilitare la lettura del testo. Ad esempio, il formato di stile globale assegna come valore il colore blu alla proprietà `selection` usata per visualizzare la barra di selezione nei componenti Casella di riepilogo e Casella combinata e assegna come valore il colore bianco alla proprietà `textSelected`; questa combinazione di colori facilita la visualizzazione del testo selezionato da parte dell'utente. Il valore relativo al colore deve essere nel formato *0xRRGGBB*.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente assegna il valore `0xffffffff` alla proprietà `textSelected` per `formStyleFormat`, producendo testo di colore bianco quando il componente viene selezionato.

```
formStyleFormat.textSelected = 0xffffffff;
```

Vedere anche

`FStyleFormat.selection`, `FStyleFormat.textDisabled`

FStyleFormat.textSize

Disponibilità

Flash Player 6.

Uso

myStyleFormat.textSize

Descrizione

Proprietà; un numero intero che specifica la dimensione in punti del testo visualizzato nei componenti assegnati al formato di stile. L'impostazione predefinita per questa proprietà è 12 punti.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica una dimensione di 10 punti per il testo di tutti i componenti assegnati a `formStyleFormat`.

```
formStyleFormat.textSize = 10;
```

Vedere anche

`FStyleFormat.textFont`

FStyleFormat.textUnderline

Disponibilità

Flash Player 6.

Uso

```
myStyleFormat.textUnderline
```

Descrizione

Proprietà; specifica se il testo visualizzato nei componenti usando il formato di stile selezionato è sottolineato (`true`) o meno (`false`). L'impostazione predefinita è `false`.

È necessario usare `FStyleFormat.applyChanges` quando si aggiornano le proprietà con un nuovo valore.

Esempio

Il codice seguente specifica che tutto il testo visualizzato nei componenti assegnati a `formStyleFormat` è sottolineato.

```
formStyleFormat.textUnderline = true;
```

Vedere anche

`FStyleFormat.textBold`, `FStyleFormat.textItalic`

Function (oggetto)

L'oggetto `Function` è disponibile in Flash MX.

Riepilogo delle proprietà valide per l'oggetto Function

Metodo	Descrizione
<code>Function.prototype</code>	Si riferisce a un oggetto che è il prototipo di una classe.

Riepilogo dei metodi validi per l'oggetto Function

Metodo	Descrizione
<code>Function.apply</code>	Attiva il codice ActionScript per richiamare una funzione.
<code>Function.call</code>	Richiama la funzione rappresentata da un oggetto <code>Function</code> .

Function.apply

Disponibilità

Flash Player 6.

Uso

```
myFunction.apply(oggetto this, oggetto arguments)
```

Parametri

oggetto this L'oggetto al quale è stata applicata la funzione *myFunction*.

oggetto arguments Una matrice i cui elementi vengono passati alla funzione *myFunction* come parametri.

Valori restituiti

Tutti i valori specificati dalla funzione richiamata.

Descrizione

Metodo; specifica il valore dell'oggetto `this` da usare all'interno delle funzioni richiamate da ActionScript. Questo metodo specifica inoltre i parametri da passare a ogni funzione richiamata. Essendo un metodo dell'oggetto `Function`, `apply` è anche un metodo di ciascun oggetto `Function` in ActionScript.

I parametri vengono specificati come oggetto `Array`. Ciò risulta spesso utile quando non si conosce il numero dei parametri da passare finché lo script non viene effettivamente eseguito.

Esempio

Le seguenti invocazioni di funzione sono equivalenti:

```
Math.atan2(1, 0)
Math.atan2.apply(null, [1, 0])
```

È possibile creare un filmato Flash che contiene caselle di immissione che consentono all'utente di specificare il nome di una funzione da richiamare e nessuno o un determinato numero di parametri da passare alla funzione. Premendo un pulsante di chiamata, verrà usato il metodo `apply` per richiamare la funzione che specifica i parametri.

In questo esempio, l'utente specifica il nome di una funzione in una casella di immissione denominata `functionName`. Il numero dei parametri viene specificato in un campo di testo di input denominato `numParameters`. Un numero massimo di 10 parametri viene specificato nei campi di testo denominati `parametro1`, `parametro2` fino a `parametro10`.

```
on (release) {
    callTheFunction();
}
...
function callTheFunction()
{
    var theFunction = eval(functionName.text);
    var n = Number(numParameters);
    var parameters = [];
    for (var i = 0; i < n; i++) {
        parameters.push(eval("parametro" + i));
    }
    theFunction.apply(null, parametri);
}
```

Function.call

Disponibilità

Flash Player 6.

Uso

myFunction.call(oggetto this, parametro1, ..., parametroN)

Parametri

oggetto this Specifica il valore dell'oggetto `this` all'interno del corpo della funzione.

parametro1 Un parametro da passare alla funzione *myFunction*. È possibile specificare zero o più parametri.

parametroN

Valori restituiti

Nessuno.

Descrizione

Metodo; richiama la funzione rappresentata da un oggetto `Function`. Ogni funzione di `ActionScript` è rappresentata da un oggetto `Function`, pertanto tutte le funzioni supportano il metodo `call`.

In quasi tutti i casi, è possibile usare l'operatore di richiamo della funzione `()` invece del metodo `call`. L'operatore di richiamo della funzione crea un codice breve e leggibile. Il metodo `call` risulta utile soprattutto quando il parametro `this` dell'invocazione della funzione deve essere esplicitamente controllato. Di solito, se una funzione viene richiamata come metodo di un oggetto all'interno del corpo della funzione, il parametro `this` viene impostato su `myObject` come illustrato nell'esempio seguente:

```
myObject.myMethod(1, 2, 3);
```

In alcuni casi, è possibile impostare il parametro `this` in modo che faccia riferimento a un'altra posizione; ad esempio, se è necessario richiamare una funzione come metodo di un oggetto quando la funzione non è effettivamente memorizzata in questo modo.

```
myObject.myMethod.call(myOtherObject, 1, 2, 3);
```

È possibile passare il valore `null` per il parametro *oggetto this* per richiamare una funzione come funzione regolare e non come metodo di un oggetto. Ad esempio, le seguenti invocazioni di funzione sono equivalenti:

```
Math.sin(Math.PI / 4)
Math.sin.call(null, Math.PI / 4)
```

Esempio

In questo esempio, viene usato il metodo `call` in modo che la funzione si comporti come metodo di un altro oggetto, senza memorizzare tale funzione nell'oggetto stesso.

```
function MyObject() {
}
function MyMethod(obj) {
    trace("this == obj? " + (this == obj));
}
var obj = new MyObject();
MyMethod.call(obj, obj);
```

L'azione `trace` invia il codice seguente alla finestra `Output`:

```
this == obj? true
```

Function.prototype

Disponibilità

Flash Player 6.

Uso

myFunction.prototype

Descrizione

Proprietà; in una funzione di costruzione, la proprietà `prototype` si riferisce a un oggetto che è il prototipo della classe costruita. Ogni istanza della classe creata dalla funzione di costruzione eredita tutte le proprietà e i metodi dall'oggetto prototipo.

fscommand

Disponibilità

Flash Player 3.

Uso

```
fscommand("comando", "parametri")
```

Parametri

comando Stringa passata all'applicazione host per un uso indeterminato o comando passato alla versione autonoma di Flash Player.

parametri Stringa passata all'applicazione host per un uso indeterminato o valore passato a Flash Player.

Valori restituiti

Nessuno.

Descrizione

Azione; consente al filmato Flash di comunicare con Flash Player o con il programma in cui si trova Flash Player, come un browser Web. È possibile anche usare il comando `fscommand` per passare messaggi a Macromedia Director, Visual Basic o Visual C++ e ad altri programmi che supportano i controlli ActiveX.

Uso 1: per inviare un messaggio a Flash Player, è necessario usare i comandi e i parametri predefiniti. La tabella seguente contiene i valori che è possibile specificare per i parametri *command* e *parameters* dell'azione `fscommand` per controllare un filmato riprodotto nel lettore Flash autonomo (proiettori compresi):

Comando	Parametri	Scopo
<code>quit</code>	Nessuno	Chiude il proiettore.
<code>fullscreen</code>	<code>true</code> o <code>false</code>	Se si specifica <code>true</code> , Flash Player viene impostato in modalità a schermo intero. Se si specifica <code>false</code> , il lettore torna alla visualizzazione dei menu normale.
<code>allowscale</code>	<code>true</code> o <code>false</code>	Se si specifica <code>false</code> , il lettore viene impostato in modo tale che il filmato sia sempre visualizzato nelle dimensioni originali e non venga mai ridimensionato. Se si specifica <code>true</code> , il filmato viene ridimensionato per coincidere con le dimensioni del lettore.
<code>showmenu</code>	<code>true</code> o <code>false</code>	Se si specifica <code>true</code> , viene attivato l'intero gruppo delle voci di menu di scelta rapida. Se si specifica <code>false</code> , tutte le voci dei menu di scelta rapida, a eccezione di Informazioni su Flash Player, vengono disattivate.
<code>exec</code>	Percorso dell'applicazione	Esegue un'applicazione dal proiettore.
<code>trapallkeys</code>	<code>true</code> o <code>false</code>	Se si specifica <code>true</code> , tutti gli eventi di tasti, compresi i tasti di scelta rapida, vengono inviati al gestore <code>onClipEvent(keyDown/keyUp)</code> di Flash Player.

Uso 2: per usare `fscommand` per inviare un messaggio a un linguaggio di scripting quale JavaScript in un browser Web, è possibile passare due argomenti qualsiasi nei parametri *command* e *parameters*. Questi argomenti possono essere stringhe o espressioni e saranno usati in una funzione JavaScript che "cattura" o gestisce l'azione `fscommand`.

In un browser Web, l'azione `fscommand` richiama la funzione JavaScript `movieName_DoFSCommand` nella pagina HTML che contiene il filmato Flash, dove `nomefilmato` è il nome di Flash Player assegnato dall'attributo `NAME` del tag `EMBED` o dalla proprietà `ID` del tag `OBJECT`. Se a Flash Player viene assegnato il nome `myMovie`, la funzione JavaScript richiamata è `myMovie_DoFSCommand`.

Uso 3: l'azione `fscommand` può inviare messaggi a Macromedia Director che vengono interpretati da Lingo come stringhe, eventi o codice Lingo eseguibile. Se il messaggio è una stringa o un evento, è necessario scrivere il codice Lingo per riceverlo dall'azione `fscommand` ed eseguire un'azione in Director.

Uso 4: in Visual Basic, Visual C++ e altri programmi che supportano i controlli ActiveX, `fscommand` invia un evento VB con due stringhe che possono essere gestite nel linguaggio di programmazione dell'ambiente. Per ulteriori informazioni, usare le parole chiave `Flash method` per eseguire una ricerca nel Centro di assistenza Flash.

Esempio

Uso 1: nell'esempio seguente, l'azione `fscommand` imposta Flash Player in modo che ridimensioni il filmato e lo ingrandisca a schermo intero quando si rilascia il pulsante.

```
on(release){
    fscommand("fullscreen", true);
}
```

Uso 2: nell'esempio riportato di seguito, l'azione `fscommand` viene applicata a un pulsante in Flash per aprire una finestra con messaggio JavaScript in un pagina HTML. Il messaggio viene inviato a JavaScript come parametro `fscommand`.

È necessario aggiungere una funzione alla pagina HTML che contiene il filmato Flash. Questa funzione, `myMovie_DoFSCommand`, si trova nella pagina HTML e attende un'azione `fscommand` in Flash. Quando in Flash viene attivata l'azione `fscommand` (ad esempio, premendo il relativo pulsante), le stringhe comando e parametri vengono passate alla funzione `myMovie_DoFSCommand`. È possibile usare le stringhe passate nel codice JavaScript o VBScript nel modo desiderato. Nell'esempio seguente, la funzione contiene un'istruzione `if` condizionale che verifica se la stringa di comando è "messagebox". In tal caso, viene aperta una casella di avviso JavaScript (messagebox) che visualizza il contenuto della stringa parametri.

```
function myMovie_DoFSCommand(command, args) {
    if (command == "messagebox") {
        alert(args);
    }
}
```

Nel documento Flash aggiungere l'azione `fscommand` a un pulsante:

```
fscommand("messagebox", "Questa finestra con messaggio è stata richiamata da Flash.")
```

È inoltre possibile usare espressioni per l'azione `fscommand` e i parametri, come nell'esempio seguente:

```
fscommand("messagebox", "Ciao, " + nome + ", Benvenuto nel nostro sito Web !")
```

Per provare il filmato, scegliere File > Anteprima pubblicazione > HTML.

Nota: se si pubblica il filmato usando Flash con il modello `FSCCommand` nelle impostazioni di pubblicazione HTML, la funzione `myMovie_DoFSCommand` viene inserita automaticamente. Gli attributi `NAME` e `ID` del filmato saranno il nome del file. Ad esempio, per il file `myMovie fla`, gli attributi verrebbero impostati su `myMovie`.

function

Disponibilità

Flash Player 5.

Uso

```
function nomefunzione ([parametro0, parametro1,...parametroN]){  
    istruzione/i  
}  
function ([parametro0, parametro1,...parametroN]){  
    istruzione/i  
}
```

Parametri

nomefunzione Il nome della nuova funzione.

parametro Un identificatore che rappresenta un parametro da passare alla funzione. Si tratta di parametri opzionali.

istruzioni Una qualsiasi istruzione ActionScript definita per il corpo di funzione.

Valori restituiti

Nessuno.

Descrizione

Azione; una serie di istruzioni che vengono definite per eseguire una determinata attività. È possibile *dichiarare*, o definire, una funzione in una posizione, quindi richiamarla da altri script in un filmato. Quando si definisce una funzione, è possibile anche specificarne i parametri. I parametri sono dei segnaposto per i valori sui quali opererà la funzione. È possibile passare diversi parametri a una funzione ogni volta che viene richiamata. In tal modo, la stessa funzione può essere riutilizzata in varie situazioni.

L'azione `return` nelle *istruzioni* di una funzione determina la restituzione di un valore.

Uso 1: dichiara una funzione e specifica *nomefunzione*, *parametri* e *istruzione/i*. Quando si richiama una funzione, si invoca la dichiarazione della funzione. È consentito il riferimento anticipato a essa, ossia all'interno dello stesso elenco di azioni è possibile dichiarare una funzione dopo averla richiamata. La dichiarazione di una funzione sostituisce qualsiasi dichiarazione precedente della stessa. È possibile usare questa sintassi ogni qualvolta è consentita un'istruzione.

Uso 2: crea una funzione anonima e la restituisce. Questa sintassi si usa nelle espressioni e il suo uso è consigliato principalmente per l'inserimento di metodi negli oggetti.

Esempio

Uso 1: l'esempio seguente definisce la funzione `sqr` che accetta un parametro, quindi restituisce il valore `square(x*x)` del parametro. Se la funzione viene dichiarata e usata all'interno dello stesso script, la relativa dichiarazione può apparire dopo il richiamo della funzione.

```
y=sqr(3);  
  
function sqr(x) {  
    return x*x;  
}
```


Uso 2: l'esempio seguente definisce l'oggetto Circle:

```
function Circle(radius) {  
    this.radius = radius;  
}
```

La seguente istruzione definisce una funzione anonima che calcola l'area del cerchio, quindi la associa come metodo all'oggetto Circle:

```
Circle.prototype.area = function () {return Math.PI * this.radius *  
    this.radius}
```

ge (maggiore di o uguale a; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore >= (maggiore di o uguale a).

Uso

espressione1 ge *espressione2*

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); confronta la rappresentazione in formato stringa di *espressione1* con quella di *espressione2* e restituisce true se *espressione1* è maggiore di o uguale a *espressione2*; altrimenti, restituisce false.

Vedere anche

>= (maggiore di o uguale a)

getProperty

Disponibilità

Flash Player 4.

Uso

getProperty(*nomeistanza*, *proprietà*)

Parametri

nomeistanza Il nome dell'istanza del clip filmato per cui si recupera la proprietà.

proprietà Una proprietà del clip filmato.

Valori restituiti

Nessuno.

Descrizione

Funzione, restituisce il valore della *proprietà* specificata per l'*istanza* di clip filmato.

Esempio

L'esempio seguente recupera la coordinata dell'asse orizzontale (`_x`) per il clip filmato `myMovie` e la assegna alla variabile `myMovieX`:

```
myMovieX = getProperty(_root.myMovie, _x);
```

getTimer

Disponibilità

Flash Player 4.

Uso

```
getTimer()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Funzione; restituisce il numero di millesimi di secondo trascorsi dall'inizio della riproduzione del filmato.

getURL

Disponibilità

Flash 2. I metodi `GET` e `POST` sono disponibili solo per Flash Player 4 e versioni successive del lettore.

Uso

```
getURL(url [, finestra [, "variabili"]])
```

Parametri

url L'URL presso cui è reperibile il documento.

finestra Parametro opzionale che specifica la finestra o il frame HTML in cui caricare il documento. È possibile immettere il nome di una finestra specifica o selezionare uno dei seguenti nomi di destinazione riservati:

- `_self` indica il frame corrente nella finestra corrente.
- `_blank` indica una finestra nuova.
- `_parent` indica il frame principale rispetto a quello corrente.
- `_top` indica il frame di primo livello nella finestra corrente.

variabili Un metodo `GET` o `POST` per l'invio di variabili. Se non ci sono variabili, omettere questo parametro. Il metodo `GET` aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo `POST` invia le variabili in un'intestazione HTTP separata e si usa in caso di lunghe stringhe di variabili.

Valori restituiti

Nessuno.

Descrizione

Azione; carica un documento da un URL specifico in una finestra oppure passa le variabili a un'altra applicazione presso l'URL definito. Prima di provare questa azione, accertarsi che il percorso specificato per il file da caricare sia corretto. Per usare un URL assoluto, ad esempio *http://www.server.com*, si deve disporre di una connessione di rete.

Esempio

Questo esempio carica un URL nuovo in una finestra vuota del browser. L'azione `getUrl` usa la variabile `incomingAd` come parametro *url* in modo che sia possibile cambiare l'URL senza dover modificare il filmato Flash. I valori della variabile `incomingAd` vengono passati a Flash direttamente nel filmato tramite l'azione `loadVariables`.

```
on(release) {  
    getUrl(incomingAd, "_blank");  
}
```

Vedere anche

`loadVariables`, `XML.send`, `XML.sendAndLoad`, `XMLSocket.close`

getVersion

Disponibilità

Flash Player 5.

Uso

```
getVersion()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Funzione; restituisce una stringa contenente le informazioni relative alla versione di Flash Player e alla piattaforma in uso.

La funzione `getVersion` restituisce solo informazioni per Flash Player 5 o versioni successive.

Esempio

Il seguente è un esempio di stringa restituita dalla funzione `getVersion`.

```
WIN 5,0,17,0
```

Ciò significa che la piattaforma è Windows, il numero di versione principale di Flash Player è 5 e quello della versione secondaria è 17 (5.0r17).

_global

Disponibilità

Flash Player 6.

Uso

`_global.identificier`

Parametri

Nessuno.

Valori restituiti

Un riferimento all'oggetto globale in cui vengono memorizzate le classi ActionScript principali, come String, Object, Math e Array.

Descrizione

Identificatore; crea variabili globali, oggetti o classi. Ad esempio, è possibile creare una libreria che venga esposta come oggetto ActionScript globale, simile all'oggetto Math o Date. Diversamente dalle variabili e dalle funzioni locali o di linea temporale, le variabili e le funzioni globali sono visibili a tutte le linee temporali e le aree di validità nel filmato Flash, purché non vengano oscurate da identificatori con nomi uguali in aree di validità interne.

Esempio

L'esempio seguente crea una funzione di primo livello, `factorial`, disponibile per tutte le linee temporali e le aree di validità in un filmato Flash:

```
_global.factorial = function (n) {  
    if (n <= 1) {  
        return 1;  
    } else {  
        return n * factorial(n-1);  
    }  
}
```

Vedere anche

`var`, `set variable`

globalStyleFormat

Disponibilità

Flash Player 6.

Uso

`globalStyleFormat.proprietàStile`

Parametri

`proprietàStile` Una proprietà dell'oggetto FStyleFormat.

Valori restituiti

Nessuno.

Descrizione

Istanza oggetto; un'istanza dell'oggetto `FStyleFormat` che definisce le proprietà del formato di stile per i componenti dell'interfaccia utente di Flash. L'istanza `globalStyleFormat` è disponibile solo dopo che i componenti dell'interfaccia utente di Flash sono stati posizionati nello stage. È possibile impostare o modificare le proprietà del formato di stile per i componenti dell'interfaccia utente di Flash nell'istanza dell'oggetto `globalStyleFormat`. Per ulteriori informazioni, consultare la sezione "Personalizzazione dei colori e del testo del componente" del capitolo "Uso dei componenti" nella *Guida all'uso di Flash*.

Esempio

Il codice seguente imposta la proprietà `arrow` della proprietà `FStyleFormat` per l'istanza `globalStyleFormat`.

```
globalStyleFormat.arrow = 0x800080;
```

Vedere anche

`FStyleFormat` (oggetto)

gotoAndPlay

Disponibilità

Flash 2.

Uso

```
gotoAndPlay(scena, fotogramma)
```

Parametri

scena Il nome della scena a cui viene inviato l'indicatore di riproduzione.

fotogramma Il numero o l'etichetta del fotogramma a cui viene inviato l'indicatore di riproduzione.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al fotogramma specificato in una scena e inizia la riproduzione da tale fotogramma. Se non viene specificata una scena, l'indicatore di riproduzione passa al fotogramma specificato all'interno della scena corrente.

Esempio

Quando l'utente sceglie un pulsante a cui è assegnata un'azione `gotoAndPlay`, l'indicatore di riproduzione passa al fotogramma 16 e inizia la riproduzione.

```
on(release) {  
    gotoAndPlay(16);  
}
```

gotoAndStop

Disponibilità

Flash 2.

Uso

```
gotoAndStop(scena, fotogramma)
```

Parametri

scena Il nome della scena a cui viene inviato l'indicatore di riproduzione.

fotogramma Il numero o l'etichetta del fotogramma a cui viene inviato l'indicatore di riproduzione.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al fotogramma specificato in una scena e lo blocca in tale posizione. Se non viene specificata una scena, l'indicatore di riproduzione passa al fotogramma all'interno della scena corrente.

Esempio

Quando l'utente sceglie un pulsante a cui è assegnata un'azione `gotoAndStop`, l'indicatore di riproduzione passa al fotogramma 5 e interrompe la riproduzione del filmato.

```
on(release) {  
    gotoAndStop(5);  
}
```

gt (maggiore di; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore `>` (maggiore di).

Uso

```
espressione1 gt espressione2
```

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Descrizione

Operatore (confronto); confronta la rappresentazione in formato stringa di *espressione1* con quella di *espressione2* e restituisce `true` se *espressione1* è maggiore di *espressione2*; altrimenti, restituisce `false`.

Vedere anche

`>` (maggiore di)

_highquality

Disponibilità

Flash Player 4.

Uso

`_highquality`

Descrizione

Proprietà (globale); specifica il livello di antialiasing applicato al filmato corrente. Specificare 2 (Ottima) per applicare un livello di alta qualità con smussatura delle bitmap sempre attiva. Specificare 1 (Alta qualità) per applicare l'antialiasing; in tal modo le bitmap verranno smussate se il filmato non contiene animazioni. Specificare 0 (Bassa qualità) per impedire l'antialiasing.

Esempio

```
_highquality = 1;
```

Vedere anche

`_quality`, `toggleHighQuality`

if

Disponibilità

Flash Player 4.

Uso

```
if(condizione) {  
    istruzione/i;  
}
```

Parametri

condizione Espressione che restituisce `true` o `false`.

istruzione/i Le istruzioni da eseguire se o quando la condizione restituisce `true`.

Valori restituiti

Nessuno.

Descrizione

Azione; valuta una condizione per determinare l'azione successiva in un filmato. Se la condizione è `true`, Flash esegue le istruzioni che seguono tale condizione all'interno di parentesi graffe (`{}`). Se la condizione è `false`, Flash ignora le istruzioni all'interno delle parentesi graffe ed esegue le istruzioni che seguono le parentesi. Usare l'azione `if` per creare codice condizionale negli script.

Esempio

Nell'esempio sotto riportato, la condizione all'interno delle parentesi restituisce la variabile `name` per verificare se il suo valore letterale è "Erica". In caso affermativo, viene eseguita l'azione `play` indicata all'interno delle parentesi graffe.

```
if(name == "Erica"){  
    play();  
}
```

Esempio

Nell'esempio sotto riportato, l'azione `if` viene usata per valutare quando un oggetto trascinato nel filmato viene rilasciato dall'utente. Se l'oggetto è stato rilasciato meno di 300 millisecondi dopo il trascinamento, la condizione restituisce `true` e vengono eseguite le istruzioni contenute all'interno delle parentesi graffe. Tali istruzioni determinano le variabili che consentono di memorizzare la nuova posizione dell'oggetto nonché la forza e la velocità alle quali esso è stato rilasciato. Anche la variabile `timePressed` viene reimpostata. Se l'oggetto è stato rilasciato più di 300 millisecondi dopo il trascinamento, la condizione restituisce `false` e non viene eseguita alcuna istruzione.

```
if (getTimer()<timePressed+300) {
    // Se la condizione è true,
    // l'oggetto è stato rilasciato.
    // Qual è la nuova posizione di questo oggetto?
    xNewLoc = this._x;
    yNewLoc = this._y;
    // Con quanta forza è stato rilasciato?
    xTravel = xNewLoc-xLoc;
    yTravel = yNewLoc-yLoc;
    // Imposta la velocità dell'oggetto in base allo
    // spazio da esso percorso
    xInc = xTravel/2;
    yInc = yTravel/2;
    timePressed = 0;
}
```

Vedere anche

`else`

ifFrameLoaded

Disponibilità

Flash Player 3. L'azione `ifFrameLoaded` è diventata obsoleta in Flash 5 ed è consigliato l'uso dell'azione `MovieClip._framesloaded`.

Uso

```
ifFrameLoaded(scena, fotogramma) {
    istruzione;
}

ifFrameLoaded(fotogramma) {
    istruzione;
}
```

Parametri

scena La scena da caricare.

fotogramma Il numero o l'etichetta del fotogramma da caricare prima di eseguire l'istruzione successiva.

istruzione/i Le istruzioni da eseguire se la scena specificata o la scena e il fotogramma specificati vengono caricati.

Valori restituiti

Nessuno.

Descrizione

Azione; verifica se il contenuto di un determinato fotogramma è disponibile localmente. Usare `onFrameLoaded` per riprodurre una semplice animazione durante il caricamento del resto del filmato sul computer locale. La differenza tra l'uso di `onFramesLoaded` e l'uso di `onFrameLoaded` risiede nel fatto che `onFramesLoaded` consente di aggiungere anche le istruzioni `if` o `else`.

Vedere anche

`MovieClip.onFramesLoaded`

#include

Disponibilità

N/D

Uso

```
#include "nomefile.as";
```

Parametri

nomefile.as Il nome del file per lo script da aggiungere al pannello Azioni; *.as* è l'estensione di file consigliata.

Valori restituiti

Nessuno.

Descrizione

Azione; include il contenuto del file specificato nel parametro quando il filmato viene provato, pubblicato o esportato. L'azione `#include` viene invocata durante la prova, la pubblicazione o l'esportazione e viene verificata quando viene eseguito un controllo della sintassi.

#initclip

Disponibilità

Flash Player 6.

Uso

```
#initclip ordine
```

Parametri

ordine Numero intero che specifica l'ordine di esecuzione di blocchi di codice `#initclip`. Questo parametro è opzionale.

Descrizione

Azione; indica l'inizio di un blocco di azioni di inizializzazione di un componente. Quando vengono inizializzati più clip filmato contemporaneamente, è possibile usare il parametro *ordine* per specificare quale inizializzare per primo. Le azioni di inizializzazione di un componente vengono eseguite quando viene definito un simbolo di clip filmato. Se il clip filmato è un simbolo esportato, le azioni di inizializzazione di un componente vengono eseguite prima delle azioni relative al fotogramma 1 del file SWF. Altrimenti, esse vengono eseguite immediatamente prima delle azioni relative al fotogramma contenente la prima istanza del simbolo di clip filmato associato al componente.

Le azioni di inizializzazione di un componente vengono eseguite solo una volta durante la riproduzione di un filmato ed è consigliabile usarle per le procedure che richiedono una sola inizializzazione, quali la definizione e la registrazione di una classe.

Esempio

Il codice riportato di seguito viene assegnato al primo fotogramma di un filmato costituito dal componente Casella di controllo. Le azioni `#initclip` e `#endinitclip` definiscono il blocco di istruzioni tra esse racchiuso come azioni di inizializzazione del componente. Tali istruzioni registrano la classe e memorizzano i metodi in un oggetto prototipo.

```
#initclip
if (typeof(CheckBox) == "undefined") {
    // Definisce la funzione di costruzione per la classe CheckBox e la classe
    // stessa function CheckBox() {
    // Imposta le associazioni di dati
    this.watch ('value', function (id, oldval, newval) { ... });
    this.watch ('label', function (id, oldval, newval) { ... });
    }
    // Imposta la catena di prototipi CheckBox da ereditare da MovieClip
    CheckBox.prototype = new MovieClip();
    // Registra CheckBox come classe per il simbolo "Casella di controllo"
    Object.registerClass("Casella di controllo", CheckBox);
    // Imposta alcuni metodi
    CheckBox.prototype.enable = function () { ... };
    CheckBox.prototype.show = function () { ... };
    CheckBox.prototype.hide = function () { ... };
    // Imposta una funzione adatta a creare istanze
    // di caselle di controllo
    CheckBox.create = function (parentMovieClip, instanceName, depth) {
        parentMovieClip.attachMovie("Casella di controllo", instanceName, depth);
    };
}
#endinitclip
```

Nota: se si copia e incolla questo codice nel pannello Azioni, viene generato un errore durante la compilazione dello script poiché le funzioni non sono definite({...})

Vedere anche

`#endinitclip`

instanceof

Disponibilità

Flash Player 6.

Uso

oggetto instanceof classe

Parametri

oggetto Oggetto di ActionScript.

classe Riferimento a una funzione di costruzione di ActionScript, come String o Date.

Valori restituiti

Se *oggetto* è un'istanza di *classe*, `instanceof` restituisce `true`; in caso contrario, restituisce `false`.

Descrizione

Operatore; determina se un oggetto appartiene a una classe specifica. Verifica se *oggetto* è un'istanza di *classe*.

Un oggetto di `ActionScript` è definibile come istanza di una classe se l'oggetto prototipo della funzione di costruzione si trova nella catena di prototipi dell'oggetto in questione.

L'operatore `instanceof` non converte i tipi di base in oggetti wrapper. Ad esempio, il codice seguente restituisce `true`:

```
new String("Ciao") instanceof String
```

Mentre il codice seguente restituisce `false`:

```
"Ciao" instanceof String
```

Esempio

L'esempio seguente mostra l'algoritmo dell'operatore `instanceof` se codificato come funzione di `ActionScript`.

```
function instanceof (theObject, theClass){
    while ((theObject = theObject.__proto__) != null) {
        if (theObject == theClass.prototype) {
            return true;
        }
    }
    return false;
}
```

Vedere anche

`typeof`

int

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta in Flash 5 ed è consigliato l'uso del nuovo metodo `Math.round`.

Uso

```
int(valore)
```

Parametri

valore Numero da arrotondare in intero.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte un numero decimale nel numero intero più vicino.

Vedere anche

`Math.floor`

isFinite

Disponibilità

Flash Player 5.

Uso

```
isFinite(espressione)
```

Parametri

espressione Valore booleano, variabile o altra espressione da valutare.

Valori restituiti

Nessuno.

Descrizione

Funzione di primo livello; valuta l'*espressione* e restituisce `true` se si tratta di un numero finito, `false` se si tratta di un valore infinito o infinito negativo. La presenza di un valore infinito o infinito negativo indica una condizione di errore matematico quale una divisione per 0.

Esempio

I seguenti sono esempi di valori restituiti per `isFinite`:

```
isFinite(56)
// Restituisce true

isFinite(Number.POSITIVE_INFINITY)
// Restituisce false
```

isNaN

Disponibilità

Flash Player 5.

Uso

```
isNaN(espressione)
```

Parametri

espressione Valore booleano, variabile o altra espressione da valutare.

Valori restituiti

Nessuno.

Descrizione

Funzione di primo livello; valuta il parametro e restituisce `true` se il valore non è un numero (NaN) a indicare la presenza di errori matematici.

Esempio

Il seguente codice illustra i valori restituiti dalla funzione `isNaN`.

```
isNaN("Albero")
// Restituisce true

isNaN(56)
// Restituisce false

isNaN(Number.POSITIVE_INFINITY)
// Restituisce false
```

Key (oggetto)

L'oggetto `Key` è un oggetto di primo livello a cui si può accedere senza usare una funzione di costruzione. I metodi validi per l'oggetto `Key` consentono di creare un'interfaccia che l'utente può controllare tramite una tastiera standard. Le proprietà dell'oggetto `Key` sono valori costanti che rappresentano i tasti usati di solito per controllare i giochi. Per un elenco completo dei valori dei codici tasto, consultare l'Appendice C "Tasti della tastiera e valori dei codici tasto" nella *Guida all'uso di Flash*.

Esempio

Il seguente script usa l'oggetto `Key` per identificare i tasti su qualsiasi tastiera e consentire così all'utente di controllare un clip filmato.

```
onClipEvent (enterFrame) {
    if(Key.isDown(Key.RIGHT)) {
        this._x=_x+10;
    } else if (Key.isDown(Key.DOWN)) {
        this._y=_y+10;
    }
}
```

Riepilogo dei metodi validi per l'oggetto Key

Metodo	Descrizione
<code>Key.addListener</code>	Registra un oggetto affinché riceva una notifica quando si invocano i metodi <code>onKeyDown</code> e <code>onKeyUp</code> .
<code>Key.getAscii</code>	Restituisce il valore ASCII dell'ultimo tasto premuto.
<code>Key.getCode</code>	Restituisce il codice di tasto virtuale dell'ultimo tasto premuto.
<code>Key.isDown</code>	Restituisce <code>true</code> se viene premuto il tasto specificato nel parametro.
<code>Key.isToggled</code>	Restituisce <code>true</code> se <code>Bloc Num</code> o <code>Bloc Maiusc</code> sono attivati.
<code>Key.removeListener</code>	Rimuove un oggetto precedentemente registrato con <code>addListener</code> .

Riepilogo delle proprietà valide per l'oggetto Key

Tutte le proprietà valide per l'oggetto Key sono costanti.

Proprietà	Descrizione
Key.BACKSPACE	Costante associata al valore del codice tasto per il tasto Backspace (8).
Key.CAPSLock	Costante associata al valore del codice tasto per il tasto Bloc Maiusc (20).
Key.CONTROL	Costante associata al valore del codice tasto per il tasto Ctrl (17).
Key.DELETEKEY	Costante associata al valore del codice tasto per il tasto Canc (46).
Key.DOWN	Costante associata al valore del codice tasto per il tasto Freccia giù (40).
Key.END	Costante associata al valore del codice tasto per il tasto Fine (35).
Key.ENTER	Costante associata al valore del codice tasto per il tasto Invio (13).
Key.ESCAPE	Costante associata al valore del codice tasto per il tasto Esc (27).
Key.HOME	Costante associata al valore del codice tasto per il tasto Home (36).
Key.INSERT	Costante associata al valore del codice tasto per il tasto Ins (45).
Key.LEFT	Costante associata al valore del codice tasto per il tasto Freccia sinistra (37).
Key.PGDN	Costante associata al valore del codice tasto per il tasto Pagina giù (34).
Key.PGUP	Costante associata al valore del codice tasto per il tasto Pagina su (33).
Key.RIGHT	Costante associata al valore del codice tasto per il tasto Freccia destra (39).
Key.SHIFT	Costante associata al valore del codice tasto per il tasto Maiusc (16).
Key.SPACE	Costante associata al valore del codice tasto per la Barra spaziatrice (32).
Key.TAB	Costante associata al valore del codice tasto per il tasto Tab (9).
Key.UP	Costante associata al valore del codice tasto per il tasto Freccia su (38).

Riepilogo dei listener validi per l'oggetto Key

Metodo	Descrizione
Key.onKeyDown	Invia una notifica quando viene premuto un tasto.
Key.onKeyUp	Invia una notifica quando viene rilasciato un tasto.

Key.addListener

Disponibilità

Flash Player 6.

Uso

Key.addListener (*nuovoListener*)

Parametri

nuovoListener Oggetto con metodi `onKeyDown` e `onKeyUp`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto affinché riceva una notifica quando si invocano i metodi `onKeyDown` e `onKeyUp`. Quando viene premuto o rilasciato un tasto, indipendentemente dall'elemento attivo, per tutti gli oggetti "in ascolto" registrati con `addListener` viene invocato il relativo metodo `onKeyDown` o `onKeyUp`. Più oggetti contemporaneamente possono ricevere notifica dalla tastiera. Se il listener `nuovoListener` è già registrato, non sono necessarie modifiche.

Esempio

In questo esempio viene creato un nuovo oggetto listener e viene definita una funzione per `onKeyDown` e `onKeyUp`. L'ultima riga usa il metodo `addListener` per registrare il listener con l'oggetto `Key` affinché riceva una notifica dagli eventi associati ai tasti Freccia giù e Freccia su.

```
myListener = new Object();
myListener.onKeyDown = function () {
    trace ("È stato premuto un tasto.");
}
myListener.onKeyUp = function () {
    trace ("È stato rilasciato un tasto.");
}
Key.addListener(myListener);
```

Key.BACKSPACE

Disponibilità

Flash Player 5.

Uso

`Key.BACKSPACE`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Backspace (8).

Key.CAPSLOCK

Disponibilità

Flash Player 5.

Uso

`Key.CAPSLOCK`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Bloc Maiusc (20).

Key.CONTROL

Disponibilità

Flash Player 5.

Uso

`Key.CONTROL`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Ctrl (17).

Key.DELETEKEY

Disponibilità

Flash Player 5.

Uso

Key.DELETEKEY

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Canc (46).

Key.DOWN

Disponibilità

Flash Player 5.

Uso

Key.DOWN

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Freccia giù (40).

Key.END

Disponibilità

Flash Player 5.

Uso

Key.END

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Fine (35).

Key.ENTER

Disponibilità

Flash Player 5.

Uso

Key.ENTER

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Invio (13).

Key.ESCAPE

Disponibilità

Flash Player 5.

Uso

`Key.ESCAPE`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Esc (27).

Key.getAscii

Disponibilità

Flash Player 5.

Uso

```
Key.getAscii();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il codice ASCII dell'ultimo tasto premuto o rilasciato. I valori ASCII restituiti corrispondono ai valori per la tastiera inglese. Ad esempio, se si preme Maiusc+2, `Key.getAscii` restituisce @ sia su una tastiera giapponese che su una tastiera inglese.

Key.getCode

Disponibilità

Flash Player 5.

Uso

```
Key.getCode();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il valore del codice tasto dell'ultimo tasto premuto. Per impostare il valore del codice tasto restituito in modo che corrisponda al codice tasto virtuale di una tastiera standard, consultare le informazioni fornite nell'Appendice C "Tasti della tastiera e valori dei codici tasto" nella *Guida all'uso di Flash*.

Key.HOME

Disponibilità

Flash Player 5.

Uso

Key.HOME

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Home (36).

Key.INSERT

Disponibilità

Flash Player 5.

Uso

Key.INSERT

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Ins (45).

Key.isDown

Disponibilità

Flash Player 5.

Uso

```
Key.isDown(codice_tasto);
```

Parametri

codice_tasto Il valore del codice tasto assegnato a un tasto specifico oppure la proprietà dell'oggetto Key associata a un tasto specifico. Per un elenco completo di tutti i codici tasto associati a una tastiera standard, consultare l'Appendice C "Tasti della tastiera e valori dei codici tasto" nella *Guida all'uso di Flash*.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce *true* se il tasto specificato in *codice_tasto* viene premuto. In ambiente Macintosh, i valori del codice tasto per i tasti Bloc Maiusc e Bloc Num sono identici.

Key.isToggled

Disponibilità

Flash Player 5.

Uso

```
Key.isToggled(codice_tasto)
```

Parametri

codice_tasto Il codice tasto per Bloc Maiusc (20) o Bloc Num (144).

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce `true` se Bloc Maiusc o Bloc Num sono attivati/disattivati. In ambiente Macintosh, i valori del codice tasto per questi tasti sono identici.

Key.LEFT

Disponibilità

Flash Player 5.

Uso

`Key.LEFT`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Freccia sinistra (37).

Key.onKeyDown

Disponibilità

Flash Player 6.

Uso

`someListener.onKeyDown`

Descrizione

Listener; quando viene premuto un tasto viene inviata una notifica al listener. Per usare `onKeyDown` è necessario creare un oggetto listener. Sarà così possibile definire una funzione per `onKeyDown` e usare il metodo `addListener` per registrare il listener con l'oggetto `Key`, come nell'esempio seguente:

```
someListener = new Object();
someListener.onKeyDown = function () { ... };
Key.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Key.addListener`

Key.onKeyUp

Disponibilità

Flash Player 6.

Uso

```
someListener.onKeyUp
```

Descrizione

Listener; quando viene rilasciato un tasto viene inviata una notifica al listener. Per usare `onKeyUp` è necessario creare un oggetto listener. Sarà così possibile definire una funzione per `onKeyUp` e usare il metodo `addListener` per registrare il listener con l'oggetto `Key`, come nell'esempio seguente:

```
someListener = new Object();  
someListener.onKeyUp = function () { ... };  
Key.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Key.addListener`

Key.PGDN

Disponibilità

Flash Player 5.

Uso

```
Key.PGDN
```

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Pagina giù (34).

Key.PGUP

Disponibilità

Flash Player 5.

Uso

```
Key.PGUP
```

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Pagina su (33).

Key.removeListener

Disponibilità

Flash Player 6.

Uso

```
Key.removeListener (listener)
```

Parametri

`listener` Un oggetto.

Valori restituiti

Se il *listener* è stato rimosso correttamente, il metodo restituisce *true*. Se il *listener* non è stato rimosso correttamente poiché, ad esempio, non era incluso nell'elenco dei listener dell'oggetto *Key*, il metodo restituisce *false*.

Descrizione

Metodo; rimuove un oggetto precedentemente registrato con il metodo `addListener`.

Key.RIGHT

Disponibilità

Flash Player 5.

Uso

`Key.RIGHT`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Freccia destra (39).

Key.SHIFT

Disponibilità

Flash Player 5.

Uso

`Key.SHIFT`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Maiusc (16).

Key.SPACE

Disponibilità

Flash Player 5.

Uso

`Key.SPACE`

Descrizione

Proprietà; costante associata al valore del codice tasto per la Barra spaziatrice (32).

Key.TAB

Disponibilità

Flash Player 5.

Uso

`Key.TAB`

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Tab (9).

Key.UP

Disponibilità

Flash Player 5.

Uso

Key.UP

Descrizione

Proprietà; costante associata al valore del codice tasto per il tasto Freccia su (38).

le (minore di o uguale a; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore `<=` (minore o uguale a).

Uso

espressione1 le *espressione2*

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); confronta *espressione1* ed *espressione2*, quindi restituisce `true` se *espressione1* è minore o uguale a *espressione2*; altrimenti restituisce `false`.

Vedere anche

`<=` (minore o uguale a)

length

Disponibilità

Flash Player 4. Questa funzione e tutte le relative funzioni stringa sono diventate obsolete in Flash 5. È consigliabile usare i metodi e la proprietà `length` dell'oggetto `String` per eseguire le stesse operazioni.

Uso

`length(espressione)`

`length(variabile)`

Parametri

espressione Una stringa qualsiasi.

variabile Il nome di una variabile.

Valori restituiti

Nessuno.

Descrizione

Funzione stringa; restituisce la lunghezza della stringa specificata o del nome della variabile.

Esempio

L'esempio seguente restituisce la lunghezza della stringa "Ciao".

```
length("Ciao")
```

Il risultato è 4.

Vedere anche

" " (delimitatore di stringa), `String.length`

_level

Disponibilità

Flash Player 4.

Uso

```
_levelN
```

Descrizione

Proprietà; riferimento alla linea temporale del filmato principale di `_levelN`. Per caricare i filmati in Flash Player, è necessario usare l'azione `loadMovieNum` prima di identificarli come target mediante la proprietà `_level`. È possibile usare `_levelN` anche per identificare come target un filmato caricato al livello assegnato da *N*.

Il filmato iniziale avviato in un'istanza di Flash Player viene caricato automaticamente in `_level0`. Il filmato in `_level0` determina la velocità dei fotogrammi, il colore di sfondo e le dimensioni dei fotogrammi di tutti gli altri filmati caricati. I filmati verranno quindi impilati nei livelli numerati progressivamente al di sopra del filmato caricato in `_level0`.

È necessario assegnare un livello a ciascun filmato caricato in Flash Player mediante l'azione `loadMovieNum`. I livelli possono essere assegnati in qualsiasi ordine. Se viene assegnato un livello che contiene già un file SWF (compreso `_level0`) il filmato corrispondente a quel livello viene scaricato e sostituito dal nuovo filmato.

Esempio

L'esempio seguente indica come arrestare l'indicatore di riproduzione nella linea temporale principale del filmato in `_level9`.

```
_level9.stop();
```

L'esempio seguente illustra come inviare al fotogramma 5 l'indicatore di riproduzione nella linea temporale principale del filmato in `_level4`. Il filmato in `_level4` deve essere stato precedentemente caricato mediante un'azione `loadMovieNum`.

```
_level4.gotoAndStop(5);
```

Vedere anche

`loadMovie`, `MovieClip.swapDepths`

loadMovie

Disponibilità

Flash Player 3.

Uso

```
loadMovie("url",livello/target[, variabili])
```

Parametri

url URL assoluto o relativo per il file SWF o JPEG da caricare. Il percorso relativo deve riferirsi al file SWF al livello 0. Tale URL deve trovarsi nello stesso sottodominio dell'URL che contiene il filmato. Per essere usati in Flash Player o verificati in modalità di prova nell'applicazione di creazione di contenuti Flash, tutti i file SWF devono essere memorizzati nella stessa cartella e i nomi dei file non possono includere specifiche della cartella o dell'unità disco.

target Percorso per identificare un clip filmato target. Il clip filmato target verrà sostituito dal filmato o dall'immagine caricati. È necessario specificare un clip filmato *target* oppure il *livello* di un filmato target; non è possibile specificare entrambi.

livello Numero intero che specifica il livello in cui il filmato verrà caricato in Flash Player. Quando si carica un filmato o un'immagine in un livello, in Modalità normale l'azione `loadMovie` nel pannello Azioni diventa `loadMovieNum`; in Modalità esperto, è necessario specificare l'azione `loadMovieNum` o selezionarla dalla casella degli strumenti Azioni.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio di variabili. Il parametro deve essere costituito dalla stringa GET o POST. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo GET aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo POST invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Azione; carica un file SWF o JPEG in Flash Player durante la riproduzione del filmato originale. L'azione `loadMovie` consente di visualizzare più filmati contemporaneamente e di passare da un filmato all'altro senza dover caricare un altro documento HTML. Senza l'azione `loadMovie`, viene visualizzato un solo filmato (file SWF), quindi Flash Player si chiude.

Quando si usa l'azione `loadMovie` è necessario specificare un livello in Flash Player o un clip filmato *target* in cui caricare il filmato. Se si specifica il livello, l'azione diventa `loadMovieNum`. Se invece il filmato viene caricato in un clip filmato *target*, è possibile usare il percorso *target* di quest'ultimo per identificare il filmato caricato.

Un'immagine o un filmato caricato in un percorso *target* eredita le proprietà di posizione, rotazione e scala del clip filmato identificato. L'angolo superiore sinistro dell'immagine o del filmato caricato si allinea al punto di registrazione del clip filmato identificato. In alternativa, se il percorso *target* è una linea temporale `_root`, l'angolo superiore sinistro dell'immagine o del filmato si allinea all'angolo superiore sinistro dello stage.

Usare l'azione `unloadMovie` per rimuovere i filmati caricati con l'azione `loadMovie`.

Esempio

Nell'esempio seguente, l'istruzione `loadMovie` è associata a un pulsante di navigazione denominato `Products`. Sullo stage è presente un clip filmato invisibile con il nome di istanza `dropZone`. L'azione `loadMovie` usa questo clip filmato come parametro `target` per caricare i prodotti presenti nel file SWF nella posizione corretta sullo stage.

```
on(release) {  
    loadMovie("products.swf",_root.dropZone);  
}
```

Nell'esempio seguente, un'immagine JPEG viene caricata dalla stessa directory del file SWF che richiama l'azione `loadMovie`:

```
loadMovie("image45.jpeg", "ourMovieClip");
```

Vedere anche

`loadMovieNum`, `unloadMovie`, `unloadMovieNum`, `_level`

loadMovieNum

Disponibilità

Flash Player 4. I file Flash 4 aperti in Flash 5 verranno convertiti per garantire l'uso della sintassi corretta.

Uso

```
loadMovieNum("url",livello[, variabili])
```

Parametri

url URL assoluto o relativo per il file SWF o JPEG da caricare. Il percorso relativo deve riferirsi al file SWF al livello 0. Tale URL deve trovarsi nello stesso sottodominio dell'URL che contiene il filmato. Per essere usati nella versione autonoma di Flash Player o verificati in modalità di prova filmato nell'applicazione di creazione Flash, tutti i file SWF devono essere memorizzati nella stessa cartella e i nomi dei file non possono includere specifiche della cartella o dell'unità disco.

livello Numero intero che specifica il livello in cui il filmato verrà caricato in Flash Player.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio di variabili. Il parametro deve essere costituito dalla stringa `GET` o `POST`. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo `GET` aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo `POST` invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Azione; carica un file SWF o JPEG in un livello di Flash Player durante la riproduzione del filmato originale. Quando si carica un filmato in un livello anziché in un percorso `target`, in Modalità normale l'azione `loadMovie` nel pannello Azioni diventa `loadMovieNum`; in Modalità esperto, è necessario specificare l'azione `loadMovieNum` o selezionarla dalla casella degli strumenti Azioni. In genere, viene visualizzato un solo filmato (file SWF), quindi Flash Player si chiude. L'azione `loadMovieNum` consente di visualizzare più filmati contemporaneamente o di passare da un filmato all'altro senza dover caricare un altro documento HTML.

Flash Player ordina i livelli a partire dal livello 0. I livelli possono essere paragonati a fogli di acetato, poiché sono completamente trasparenti tranne che nelle aree contenenti gli oggetti presenti su ogni livello. Quando si usa l'azione `loadMovie`, è necessario specificare un livello in Flash Player nel quale caricare il filmato. Una volta caricato il filmato in un livello, per identificarlo è possibile usare la sintassi `_levelN`, dove *N* è il numero del livello.

Quando si carica un filmato, è possibile specificare qualsiasi numero di livello e caricare filmati in un livello sul quale sia già stato caricato un file SWF. In questo caso, il nuovo filmato sostituirà il file SWF esistente. Se si carica un filmato nel livello 0, ogni livello in Flash Player viene scaricato e il livello 0 viene sostituito dal nuovo file. Il filmato nel livello 0 determina la velocità dei fotogrammi, il colore di sfondo e le dimensioni dei fotogrammi di tutti gli altri filmati caricati.

L'azione `loadMovieNum` consente anche di caricare file JPEG in un filmato durante la riproduzione. Durante il caricamento del file, l'angolo superiore sinistro dell'immagine si allinea all'angolo superiore sinistro dello stage sia nel caso di immagini che di file SWF. In entrambi i casi, il file caricato eredita le proprietà di rotazione e scala e i contenuti originali vengono sovrascritti.

Usare l'azione `unloadMovieNum` per rimuovere i filmati o le immagini caricati mediante l'azione `loadMovieNum`.

Esempio

Nell'esempio seguente, l'immagine JPEG "image45.jpg" viene caricata nel livello 2 di Flash Player.

```
loadMovieNum("http://www.blag.com/image45.jpg", 2); //
```

Vedere anche

`loadMovie`, `unloadMovie`, `unloadMovieNum`, `_level`

loadVariables

Disponibilità

Flash Player 4.

Uso

```
loadVariables("url", livello/target[, variabili])
```

Parametri

url URL assoluto o relativo in cui si trovano le variabili. Se si accede al filmato usando un browser Web, l'host dell'URL deve trovarsi nello stesso sottodominio del filmato stesso.

livello Numero intero che specifica il livello in cui vengono caricate le variabili in Flash Player. Quando si caricano delle variabili in un livello, nel pannello Azioni in Modalità normale l'azione diventa `loadMovieNum`; in Modalità esperto, è necessario specificare l'azione `loadMovieNum` o selezionarla dalla casella degli strumenti Azioni.

target Il percorso target di un clip filmato in cui caricare le variabili. È necessario specificare un clip filmato *target* oppure *livello* (il livello) in Flash Player; non è possibile specificare entrambi.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio di variabili. Il parametro deve essere costituito dalla stringa `GET` o `POST`. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo `GET` aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo `POST` invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Azione; legge i dati da un file esterno, quale un file di testo o un testo creato tramite script CGI, ASP (Active Server Pages), PHP (Personal Home Page) o script Perl e imposta i valori delle variabili in un livello di Flash Player o in un clip filmato target. Questa azione consente anche di aggiornare le variabili nel filmato attivo assegnando ad esse nuovi valori.

Il testo nell'URL specificato deve essere in formato MIME standard *application/x-www-form-urlencoded* (un formato standard usato dagli script CGI). Il filmato e le variabili da caricare devono risiedere nello stesso sottodominio. Non vi sono limitazioni sul numero di variabili che è possibile specificare. Ad esempio, questa frase definisce diverse variabili:

```
company=Macromedia&address=600+Townsend&city=San+Francisco&zip=94103
```

Il primo filmato aperto in un'istanza di Flash Player viene caricato nel livello più basso (identificato nel codice come `_level0`). Quando si usa l'azione `loadMovie` o `loadMovieNum` per caricare in sequenza più filmati in Flash Player, è necessario assegnare un numero di livello in Flash Player o un clip filmato target per ogni filmato caricato. Quando si usa l'azione `loadVariables`, è necessario specificare un livello in Flash Player o un clip filmato target in cui caricare le variabili.

Esempio

Nell'esempio seguente, le informazioni vengono caricate da un file di testo nei campi di testo del clip filmato `varTarget` sulla linea temporale principale. I nomi delle variabili dei campi di testo devono corrispondere ai nomi delle variabili nel file `data.txt`.

```
on(release) {  
    loadVariables("data.txt", "_root.varTarget");  
}
```

Vedere anche

`loadVariablesNum`, `loadMovie`, `loadMovieNum`, `getURL`, `MovieClip.loadMovie`, `MovieClip.loadVariables`

loadVariablesNum

Disponibilità

Flash Player 4. I file Flash 4 aperti in Flash 5 verranno convertiti per garantire l'uso della sintassi corretta.

Uso

```
loadVariables (url [,livello [, variabili]]);
```

Parametri

url URL assoluto o relativo in cui si trovano le variabili. Se si accede al filmato usando un browser Web, l'host dell'URL deve trovarsi nello stesso sottodominio del filmato stesso.

livello Numero intero che specifica il livello in cui vengono caricate le variabili in Flash Player.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio di variabili. Il parametro deve essere costituito dalla stringa GET o POST. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo GET aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo POST invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Azione; legge i dati da un file esterno, quale un file di testo o un testo creato tramite script CGI, ASP (Active Server Pages), PHP (Personal Home Page) o script Perl e imposta i valori delle variabili in un livello di Flash Player. Questa azione consente anche di aggiornare le variabili nel filmato attivo assegnando ad esse nuovi valori. Quando si caricano delle variabili in un livello, nel pannello Azioni in Modalità normale l'azione diventa `loadMovieNum`; in Modalità esperto, è necessario specificare l'azione `loadMovieNum` o selezionarla dalla casella degli strumenti Azioni.

Il testo nell'URL specificato deve essere in formato MIME standard *application/x-www-form-urlencoded* (un formato standard usato dagli script CGI). Il filmato e le variabili da caricare devono risiedere nello stesso sottodominio. Non vi sono limitazioni sul numero di variabili che è possibile specificare. Ad esempio, questa frase definisce diverse variabili:

```
company=Macromedia&address=600+Townsend&city=San+Francisco&zip=94103
```

Il primo filmato aperto in un'istanza di Flash Player viene caricato nel livello più basso (identificato nel codice come `_level0`). Quando si usa l'azione `loadMovie` o `loadMovieNum` per caricare in sequenza più filmati in Flash Player, è necessario assegnare un numero di livello in Flash Player o un clip filmato target per ogni filmato caricato. Quando si usa l'azione `loadVariablesNum`, è necessario specificare un livello in Flash Player nel quale caricare le variabili.

Esempio

Nell'esempio seguente, le informazioni vengono caricate da un file di testo nei campi di testo della linea temporale principale del filmato al livello 0 in Flash Player. I nomi delle variabili dei campi di testo devono corrispondere ai nomi delle variabili nel file `data.txt`.

```
on(release) {  
    loadVariablesNum("data.txt", 0);  
}
```

Vedere anche

`getUrl`, `loadMovie`, `loadMovieNum`, `loadVariables`, `MovieClip.loadMovie`, `MovieClip.loadVariables`

LoadVars (oggetto)

L'oggetto `LoadVars` rappresenta un'alternativa all'azione `loadVariables` per trasferire le variabili da un filmato Flash a un server.

È possibile usare l'oggetto `LoadVars` per ottenere informazioni sugli errori e sulle azioni in corso, e per scaricare dati in streaming. L'oggetto `LoadVars` funziona in modo analogo all'oggetto XML, in quanto usa i metodi `load`, `send` e `sendAndLoad` per comunicare con un server. La differenza principale tra l'oggetto `LoadVars` e l'oggetto XML è che `LoadVars` trasferisce le coppie nome/valore di ActionScript anziché trasferire una struttura ad albero DOM XML memorizzata nell'oggetto XML.

All'oggetto `LoadVars` si applicano le stesse restrizioni di sicurezza valide per l'oggetto `XML`.

Prima di richiamare i metodi dell'oggetto `LoadVars`, è necessario crearne un'istanza usando la funzione di costruzione `new LoadVars()`.

L'oggetto `LoadVars` è supportato da Flash Player 6 e versioni successive.

Riepilogo dei metodi validi per l'oggetto `LoadVars`

Metodo	Descrizione
<code>LoadVars.load</code>	Scarica variabili da un URL specificato.
<code>LoadVars.getBytesTotal</code>	Restituisce il numero di byte caricati da un metodo <code>load</code> o <code>sendAndLoad</code> .
<code>LoadVars.getBytesTotal</code>	Restituisce il numero totale di byte che verranno scaricati mediante un metodo <code>load</code> o <code>sendAndLoad</code> .
<code>LoadVars.send</code>	Invia le variabili da un oggetto <code>LoadVars</code> a un URL.
<code>LoadVars.sendAndLoad</code>	Invia le variabili da un oggetto <code>LoadVars</code> a un URL e scarica la risposta del server in un oggetto target.
<code>LoadVars.toString</code>	Restituisce una stringa con codifica URL contenente tutte le variabili enumerabili nell'oggetto <code>LoadVars</code> .

Riepilogo delle proprietà valide per l'oggetto `LoadVars`

Tutte le proprietà valide per l'oggetto `Key` sono costanti.

Proprietà	Descrizione
<code>LoadVars.contentType</code>	Indica il tipo MIME dei dati.
<code>LoadVars.load</code>	Valore booleano che indica se un'operazione <code>load</code> o <code>sendAndLoad</code> è stata completata.

Riepilogo degli eventi validi per l'oggetto `LoadVars`

Metodo	Descrizione
<code>LoadVars.onLoad</code>	Invocato quando un'operazione <code>load</code> o <code>sendAndLoad</code> è stata completata.

Funzione di costruzione per l'oggetto `LoadVars`

Disponibilità

Flash Player 6.

Uso

```
new LoadVars()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un'istanza dell'oggetto `LoadVars`. È quindi possibile usare i metodi dell'oggetto `LoadVars` per inviare e caricare dati.

Esempio

Nell'esempio seguente, viene creata un'istanza dell'oggetto `LoadVars` denominata `myLoadVars`:

```
myLoadVars = new LoadVars();
```

LoadVars.contentType

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.contentType
```

Descrizione

Proprietà; il tipo MIME inviato al server quando si richiama il metodo `LoadVars.send` o `LoadVars.sendAndLoad`. Il tipo predefinito è `application/x-www-urlform encoded`.

Vedere anche

`LoadVars.send`, `LoadVars.sendAndLoad`

LoadVars.getBytesLoaded

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.getBytesLoaded()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero di byte scaricati mediante un metodo `load` o `sendAndLoad`. Il metodo `getBytesLoaded` restituisce `undefined` se non è in corso alcuna operazione `load` o se l'operazione `load` non è stata ancora avviata.

LoadVars.getBytesTotal

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.getBytesTotal()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce il numero totale di byte che vengono scaricati mediante un'operazione `load` o `sendAndLoad`. Il metodo `getBytesTotal` restituisce `undefined` se non è in corso alcuna operazione `load` o se l'operazione `load` non è stata ancora avviata. Questo metodo restituisce `undefined` anche quando il numero totale di byte non può essere determinato, ad esempio nel caso in cui lo scaricamento è cominciato ma il server non ha trasmesso un HTTP in formato `Content-Length`.

LoadVars.load

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.load(url)
```

Parametri

url L'URL da cui scaricare le variabili.

Valori restituiti

Una stringa.

Descrizione

Metodo; scarica variabili da un URL specificato, analizza i dati della variabile e inserisce le variabili risultanti in `loadVarsObject`. Le proprietà contenute in `loadVarsObject` con nomi uguali a quelli delle variabili scaricate vengono sovrascritte. Le proprietà contenute in `loadVarsObject` con nomi diversi da quelli delle variabili scaricate non vengono eliminate. Tale azione è asincrona.

Il contenuto dei dati scaricati deve essere del tipo MIME *application/x-www-urlform-encoded*. Si tratta dello stesso formato usato da `loadVariables`.

Questo metodo è simile al metodo `XML.load` dell'oggetto `XML`.

LoadVars.loaded

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.loaded
```

Descrizione

Proprietà; `undefined` per impostazione predefinita. Quando viene avviata un'operazione `load` o `sendAndLoad`, la proprietà `loaded` viene impostata su `false`. Quando l'operazione `load` o `sendAndLoad` viene completata, la proprietà `loaded` viene impostata su `true`. Se l'operazione `load` non è ancora stata completata o si è verificato un errore, la proprietà `loaded` rimane impostata su `false`.

La proprietà `LoadVars.loaded` è simile alla proprietà `XML.loaded` dell'oggetto `XML`.

LoadVars.onLoad

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.onLoad(successo)
```

Parametri

successo Il parametro successo indica se l'operazione load è stata completata correttamente (true) o meno (false).

Valori restituiti

Un valore booleano.

Descrizione

Gestore di evento; invocato quando un'operazione load o sendAndLoad viene completata. Se l'operazione è stata eseguita correttamente, a *loadVarsObject* vengono assegnate le variabili scaricate mediante l'operazione load o sendAndLoad; tali variabili sono disponibili quando onLoad viene invocato.

Questo metodo è undefined per impostazione predefinita; è possibile definirlo assegnandogli una funzione di callback.

Questo metodo è simile al metodo XML.onLoad dell'oggetto XML.

LoadVars.send

Disponibilità

Flash Player 6.

Uso

```
loadVarsObject.send(url [, target, metodo] )
```

Parametri

loadVarsObject L'oggetto LoadVars da cui caricare le variabili.

url L'URL in cui caricare le variabili.

target Il frame del browser in cui verranno visualizzate tutte le risposte.

metodo Il metodo "GET" o "POST" del protocollo HTTP.

Valori restituiti

Una stringa.

Descrizione

Metodo; invia le variabili presenti nell'oggetto *myLoadVars* all'URL specificato. Tutte le variabili enumerabili nell'oggetto *myLoadVars* sono concatenate in una stringa nel formato *application/x-www-urlform-encoded* per impostazione predefinita; la stringa viene inviata all'URL mediante il metodo HTTP POST. Si tratta dello stesso formato usato dall'azione loadVariables. Il contenuto di tipo MIME inviato nelle intestazioni delle richieste HTTP è il valore di *myLoadVars.contentType* o il tipo predefinito *application/x-www-urlform-encoded*. Si usa il metodo "POST" a meno che non sia specificato "GET".

Se il parametro *target* è specificato, la risposta del server viene visualizzata nel frame del browser denominato target. Se il parametro *target* è omissso, la risposta del server viene rimossa.

Questo metodo è simile al metodo XML.send dell'oggetto XML.

LoadVars.sendAndLoad

Disponibilità

Flash Player 6.

Uso

```
myLoadVars.sendAndLoad(url, oggettoTarget[, metodo])
```

Parametri

loadVarsObject L'oggetto LoadVars da cui caricare le variabili.

url L'URL in cui caricare le variabili.

oggettoTarget L'oggetto LoadVars in cui vengono inserite le variabili scaricate.

metodo Il metodo "GET" o "POST" del protocollo HTTP.

Valori restituiti

Una stringa.

Descrizione

Metodo; invia le variabili presenti nell'oggetto *myLoadVars* all'URL specificato. La risposta del server viene scaricata, analizzata come dati della variabile e le variabili risultanti vengono inserite nell'oggetto *oggettoTarget*.

Le variabili vengono inviate con una procedura simile a quella di `LoadVars.send`. Inoltre, esse vengono scaricate in *oggettoTarget* con una procedura simile a quella di `LoadVars.load`.

Questo metodo è simile al metodo `XML.sendAndLoad` dell'oggetto XML.

LoadVars.toString

Disponibilità

Flash Player 6.

Uso

```
loadVarsObject.toString()
```

Parametri

Nessuno.

Valori restituiti

Una stringa.

Descrizione

Metodo; restituisce una stringa contenente tutte le variabili enumerabili nell'oggetto LoadVars, nel contenuto MIME con codifica *application/x-www-urlform-encoded*.

Esempio

```
var myVars = new LoadVars();  
myVars.name = "Gary";  
myVars.age = 26;  
trace (myVars.toString());  
risulterà come  
name=Gary&age=26
```

lt (minore di; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto in Flash 5 ed è consigliato l'uso del nuovo operatore < (minore di).

Uso

espressione1 < *espressione2*

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Descrizione

Operatore (confronto); confronta *espressione1* ed *espressione2*, quindi restituisce *true* se *espressione1* è minore di *espressione2*; altrimenti restituisce *false*.

Vedere anche

< (minore di)

Math (oggetto)

L'oggetto Math è un oggetto di primo livello a cui si può accedere senza usare una funzione di costruzione.

I metodi e le proprietà di questo oggetto consentono di accedere e gestire le costanti e le funzioni matematiche. Tutte le proprietà e i metodi dell'oggetto Math sono statici e per richiamarli è necessario usare la sintassi `Math.metodo(parametro)` o `Math.costante`. In ActionScript, le costanti sono definite come numeri in virgola mobile e doppia precisione secondo le specifiche IEEE-754.

Diversi metodi dell'oggetto Math richiedono il valore degli angoli in radianti come parametro. La seguente equazione consente di calcolare il valore in radianti o di passare semplicemente l'equazione (immettendo un valore corrispondente in gradi) come parametro.

Per calcolare il valore in radianti, usare la formula seguente:

```
radianti = Math.PI/180 * gradi
```

Nell'esempio seguente, l'equazione viene passata come parametro in modo da calcolare il seno di un angolo di 45 gradi:

```
Math.SIN(Math.PI/180 * 45) equivale a Math.SIN(.7854)
```

L'oggetto Math è totalmente supportato in ambiente Flash Player 5. In Flash Player 4, i metodi dell'oggetto Math funzionano ugualmente, ma essi sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Riepilogo dei metodi validi per l'oggetto Math

Metodo	Descrizione
Math.abs	Calcola il valore assoluto.
Math.acos	Calcola l'arco coseno.
Math.asin	Calcola l'arco seno.
Math.atan	Calcola l'arcotangente.
Math.atan2	Calcola l'angolo dall'asse x al punto.
Math.ceil	Arrotonda un numero per eccesso all'intero più vicino.
Math.cos	Calcola il coseno.
Math.exp	Calcola un valore esponenziale.
Math.floor	Arrotonda un numero per difetto all'intero più vicino.
Math.log	Calcola il logaritmo naturale.
Math.max	Restituisce il numero maggiore tra due interi.
Math.min	Restituisce il numero minore tra due interi.
Math.pow	Calcola x elevato alla y .
Math.random	Restituisce un numero pseudo-casuale compreso tra 0,0 e 1,2.
Math.round	Arrotonda al numero intero più vicino.
Math.sin	Calcola il seno.
Math.sqrt	Calcola la radice quadrata.
Math.tan	Calcola la tangente.

Riepilogo delle proprietà valide per l'oggetto Math

Tutte le proprietà valide per l'oggetto Math sono costanti.

Proprietà	Descrizione
Math.E	Costante di Eulero, base dei logaritmi naturali (circa 2,718).
Math.LN2	Il logaritmo naturale di 2 (circa 0,693).
Math.LOG2E	Logaritmo di e in base 2 (circa 1,442).
Math.LN10	Logaritmo naturale di 10 (circa 2,302).
Math.LOG10E	Logaritmo di e in base 10 (circa 0,434).
Math.PI	Il rapporto tra la circonferenza di un cerchio e il suo diametro (circa 3,14159).
Math.SQRT1_2	Il reciproco della radice quadrata di 1/2 (circa 0,707).
Math.SQRT2	La radice quadrata di 2 (circa 1,414).

Math.abs

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.abs(x);
```

Parametri

x Un numero.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce il valore assoluto del numero specificato dal parametro x .

Math.acos

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.acos(x);
```

Parametri

x Un numero compreso tra $-1,0$ e $1,0$.

Valori restituiti

Nessuno.

Descrizione

Metodo; calcola e restituisce l'arcocoseno del numero specificato dal parametro x in radianti.

Math.asin

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.asin(x);
```

Parametri

x Un numero compreso tra $-1,0$ e $1,0$.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce l'arcoseno del numero specificato nel parametro x in radianti.

Math.atan

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

`Math.atan(x)`

Parametri

x Un numero.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce l'arcotangente del numero specificato nel parametro x . Restituisce un valore compreso tra il valore di π greco negativo diviso per 2 e π greco positivo diviso per 2.

Math.atan2

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

`Math.atan2(y, x);`

Parametri

x Numero che specifica la coordinata x del punto.

y Numero che specifica la coordinata y del punto.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce l'arcotangente delle coordinate y/x in radianti. Il valore restituito rappresenta l'angolo opposto all'angolo retto di un triangolo rettangolo, dove x è la lunghezza del lato adiacente e y la lunghezza del lato opposto.

Math.ceil

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.ceil(x);
```

Parametri

x Numero o espressione.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce il valore arrotondato per eccesso del numero o dell'espressione specificata. Il valore arrotondato per eccesso di un numero corrisponde all'intero più vicino, maggiore o uguale al numero stesso.

Math.cos

Uso

```
Math.cos(x);
```

Parametri

x Un angolo misurato in radianti.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce il coseno (un valore compreso tra $-1,0$ e $1,0$) dell'angolo specificato dal parametro x . L'angolo x deve essere specificato in radianti. Per calcolare un radiante, usare l'espressione fornita nell'introduzione all'oggetto Math.

Math.E

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.E
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; costituisce la base per i logaritmi naturali, espressa come e . Il valore di e è uguale a circa 2,71828.

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Math.exp

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.exp(x);
```

Parametri

x Esponente; numero o espressione.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce il valore della base del logaritmo naturale (e) elevato alla potenza specificata nel parametro x . La costante `Math.E` fornisce il valore di e .

Math.floor

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.floor(x);
```

Parametri

x Numero o espressione.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce il valore arrotondato per difetto del numero o dell'espressione specificati nel parametro x . Il valore arrotondato in difetto è il numero intero più vicino, minore o uguale al numero o all'espressione specificata.

Esempio

Il codice seguente restituisce il valore 12.

```
Math.floor(12.5);
```

Math.log

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.log(x);
```

Parametri

x Numero o espressione con un valore maggiore di 0.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce il logaritmo naturale del parametro *x*.

Math.LOG2E

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.LOG2E
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il logaritmo in base 2 della costante *e* (Math.E), espresso come $\log_2 e$, con un valore uguale a circa 1,442695040888963387.

Math.LOG10E

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.LOG10E
```


Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il logaritmo in base 10 della costante e (Math.E), espresso come $\log_{10}e$, con un valore uguale a circa 0,43429448190325181667.

Math.LN2

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

Math.LN2

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il logaritmo naturale di 2, espresso come $\log_e 2$, con un valore uguale a circa 0,69314718055994528623.

Math.LN10

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

Math.LN10

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il logaritmo naturale di 10, espresso come $\log_e 10$, con un valore uguale a circa 2,3025850929940459011.

Math.max

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.max(x , y);
```

Parametri

x Numero o espressione.

y Numero o espressione.

Valori restituiti

Un numero.

Descrizione

Metodo; valuta *x* e *y*, quindi restituisce il valore maggiore.

Math.min

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.min(x , y);
```

Parametri

x Numero o espressione.

y Numero o espressione.

Valori restituiti

Nessuno.

Descrizione

Metodo; valuta *x* e *y*, quindi restituisce il valore minore.

Math.PI

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.PI
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il rapporto tra la circonferenza di un cerchio e il suo diametro, espresso come pi greco, con un valore pari a circa 3,14159265358979.

Math.pow

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.pow(x , y);
```

Parametri

x Numero da elevare alla potenza.

y Numero che specifica la potenza a cui elevare il valore *x*.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce *x* elevato alla *y*, x^y .

Math.random

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.random();
```

Parametri

Nessuno.

Valori restituiti

Un numero.

Descrizione

Metodo; restituisce *n*, dove $0 \leq n < 1$.

Vedere anche

random

Math.round

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.round(x);
```

Parametri

x Un numero.

Valori restituiti

Un numero.

Descrizione

Metodo; arrotonda per eccesso o per difetto il valore del parametro x al numero intero più vicino e restituisce il valore corrispondente.

Math.sin

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.sin(x);
```

Parametri

x Un angolo misurato in radianti.

Valori restituiti

Nessuno.

Descrizione

Metodo; calcola e restituisce il seno dell'angolo specificato, in radianti. Per calcolare un radiante, usare l'espressione fornita nell'introduzione all'oggetto Math.

Math.sqrt

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

```
Math.sqrt(x);
```

Parametri

x Numero o espressione con un valore maggiore o uguale 0.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce la radice quadrata del numero specificato.

Math.SQRT1_2

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

Math.SQRT1_2

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante matematica; rappresenta il reciproco della radice quadrata di un mezzo (1/2), con un valore uguale a circa 0,707106781186.

Math.SQRT2

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

Math.SQRT2

Parametri

Nessuno.

Descrizione

Costante matematica; restituisce la radice quadrata di 2, con un valore uguale a circa 1,414213562373.

Math.tan

Disponibilità

Flash Player 5. In Flash Player 4, i metodi e le proprietà dell'oggetto Math sono emulati per approssimazione e la loro accuratezza può risultare inferiore rispetto alle funzioni matematiche non emulate supportate da Flash Player 5.

Uso

Math.tan(x);

Parametri

x Un angolo misurato in radianti.

Valori restituiti

Un numero.

Descrizione

Metodo; calcola e restituisce la tangente dell'angolo specificato. Per calcolare un radiante, usare le informazioni fornite nell'introduzione a Math (oggetto).

maxscroll

Disponibilità

Flash Player 4.

Uso

```
variable_name.maxscroll
```

Descrizione

Proprietà (sola lettura); proprietà obsoleta che indica il numero della prima riga visibile nel campo di testo quando anche l'ultima riga del campo è visibile. La proprietà `maxscroll` opera insieme alla proprietà `scroll` per controllare la visualizzazione delle informazioni in un campo di testo. È possibile recuperare il valore questa proprietà, ma non modificarla.

Vedere anche

`TextField.maxscroll`, `TextField.scroll`

mbchr

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta ed è consigliato l'uso del metodo `String.fromCharCode`.

Uso

```
mbchr(numero)
```

Parametri

numero Il numero da convertire in un carattere a più byte.

Valori restituiti

Una stringa.

Descrizione

Funzione stringa; converte un numero in codice ASCII in un carattere a più byte.

Vedere anche

`String.fromCharCode`

mblength

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta ed è consigliato l'uso di String (oggetto).

Uso

```
mblength(stringa);
```

Parametri

stringa Una stringa.

Valori restituiti

Un numero.

Descrizione

Funzione stringa; restituisce la lunghezza della stringa di caratteri a più byte.

mbord

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta in Flash 5 ed è consigliato l'uso del nuovo metodo `String.charCodeAt`.

Uso

```
mbord(carattere);
```

Parametri

carattere Il carattere da convertire in un numero a più byte.

Valori restituiti

Un numero.

Descrizione

Funzione stringa; converte il carattere specificato in un numero a più byte.

Vedere anche

`String.fromCharCode`

mbsubstring

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta in Flash 5 e si consiglia l'uso del nuovo metodo `String.substr`.

Uso

```
mbsubstring(valore, indice, numero)
```

Parametri

valore La stringa a più byte dalla quale estrarre una nuova stringa a più byte.

indice Il numero del primo carattere da estrarre.

numero Il numero di caratteri da includere nella stringa estratta, escluso il carattere specificato nell'argomento `indice`.

Valori restituiti

Una stringa.

Descrizione

Funzione stringa; estrae una nuova stringa di caratteri a più byte da una stringa di caratteri a più byte.

Vedere anche

`String.substr`

method

Disponibilità

Flash Player 6.

Uso

```
oggetto.metodo = function ([parametri]) {  
    ...corpo della funzione...  
};
```

Parametri

oggetto Identificatore di un oggetto.

metodo Identificatore di un metodo.

parametri Parametri da passare alla funzione. Opzionale.

Valori restituiti

Nessuno.

Descrizione

Azione (solo in modalità normale); consente di definire i metodi per gli oggetti usando il pannello Azioni in modalità normale. Per ulteriori informazioni sulla definizione dei metodi per gli oggetti, consultare la *Guida all'uso di Flash*.

Mouse (oggetto)

L'oggetto Mouse è un oggetto di primo livello a cui si può accedere senza usare una funzione di costruzione. È possibile usare i metodi dell'oggetto Mouse per nascondere e mostrare il puntatore in un filmato. Per impostazione predefinita il puntatore è visibile, ma è possibile nascondere e usare un puntatore personalizzato creato come clip filmato.

Riepilogo dei metodi validi per l'oggetto Mouse

Metodo	Descrizione
<code>Mouse.addListener</code>	Registra un oggetto affinché riceva un messaggio di notifica degli eventi <code>onMouseDown</code> , <code>onMouseMove</code> e <code>onMouseUp</code> .
<code>Mouse.hide</code>	Nasconde il puntatore nel filmato.
<code>Mouse.removeListener</code>	Rimuove un oggetto precedentemente registrato con il metodo <code>addListener</code> .
<code>Mouse.show</code>	Visualizza il puntatore nel filmato.

Riepilogo dei metodi validi per l'oggetto listener Mouse

Metodo	Descrizione
<code>MovieClip.onMouseDown</code>	Il listener riceve notifica quando il pulsante del mouse viene premuto.
<code>MovieClip.onMouseMove</code>	Il listener riceve notifica quando il mouse viene spostato.
<code>MovieClip.onMouseUp</code>	Il listener riceve notifica quando il pulsante viene rilasciato.

Mouse.addListener

Disponibilità

Flash Player 6.

Uso

`Mouse.addListener` (*nuovoListener*)

Parametri

nuovoListener Un oggetto.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto affinché riceva notifica dai gestori di callback `onMouseDown`, `onMouseMove` e `onMouseUp`.

Il nuovo parametro *nuovoListener* dovrebbe contenere un oggetto con metodi definiti per gli eventi `onMouseDown`, `onMouseMove` e `onMouseUp`.

Quando il pulsante del mouse viene premuto o rilasciato, o quando il mouse stesso viene spostato, indipendentemente dall'elemento attivo, per tutti gli oggetti registrati con il metodo `addListener` viene invocato il corrispondente metodo `onMouseDown`, `onMouseMove` oppure `onMouseUp`. Più oggetti contemporaneamente possono ricevere notifica dalla tastiera. Se il listener *nuovoListener* è già registrato, non sono necessarie modifiche.

Mouse.hide

Disponibilità

Flash Player 5.

Uso

```
Mouse.hide();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; nasconde il puntatore in un filmato. Per impostazione predefinita il puntatore è visibile.

Esempio

Il codice seguente, associato a un clip filmato nella linea temporale principale, nasconde il puntatore standard e imposta le posizioni *x* e *y* dell'istanza di clip filmato `customCursor` sulle posizioni *x* e *y* del mouse nella linea temporale principale.

```
onClipEvent(enterFrame) {  
    Mouse.hide();  
    customCursorMC._x = _root._xmouse;  
    customCursorMC._y = _root._ymouse;  
}
```

Vedere anche

`Mouse.show`, `MovieClip._xmouse`, `MovieClip._ymouse`

Mouse.onMouseDown

Disponibilità

Flash Player 6.

Uso

```
someListener.onMouseDown
```

Descrizione

Listener; quando viene premuto il tasto del mouse, viene inviata una notifica al listener. Per usare il listener `onMouseDown` è necessario creare un oggetto listener. Sarà così possibile definire una funzione per `onMouseDown` e usare il metodo `addListener` per registrare il listener con l'oggetto `Mouse`, come nell'esempio seguente:

```
someListener = new Object();  
someListener.onMouseDown = function () { ... };  
Mouse.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Mouse.addListener`

Mouse.onMouseMove

Disponibilità

Flash Player 6.

Uso

someListener.onMouseMove

Descrizione

Listener; quando viene spostato il mouse, viene inviata una notifica al listener. Per usare il listener `onMouseMove` è necessario creare un oggetto listener. Sarà così possibile definire una funzione per `onMouseMove` e usare il metodo `addListener` per registrare il listener con l'oggetto `Mouse`, come nell'esempio seguente:

```
someListener = new Object();
someListener.onMouseDown = function () { ... };
Mouse.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Mouse.addListener`

Mouse.onMouseUp

Disponibilità

Flash Player 6.

Uso

someListener.onMouseUp

Descrizione

Listener; quando il pulsante del mouse viene rilasciato, viene inviata una notifica al listener. Per usare il listener `onMouseUp` è necessario creare un oggetto listener. Sarà così possibile definire una funzione per `onMouseUp` e usare il metodo `addListener` per registrare il listener con l'oggetto `Mouse`, come nell'esempio seguente:

```
someListener = new Object();
someListener.onMouseUp = function () { ... };
Mouse.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Mouse.addListener`

Mouse.removeListener

Disponibilità

Flash Player 6.

Uso

`Mouse.removeListener (listener)`

Parametri

`listener` Un oggetto.

Valori restituiti

Se l'oggetto *listener* viene rimosso correttamente, il metodo restituisce `true`; se, viceversa, il *listener* non è stato rimosso correttamente poiché, ad esempio, non figurava nell'elenco dei listener dell'oggetto `Mouse`, il metodo restituisce `false`.

Descrizione

Metodo; rimuove un oggetto precedentemente registrato con il metodo `addListener`.

Mouse.show

Disponibilità

Flash Player 5.

Uso

`Mouse.show()`;

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; visualizza il puntatore in un filmato. Per impostazione predefinita il puntatore è visibile.

Vedere anche

`Mouse.show`, `MovieClip._xmouse`, `MovieClip._ymouse`

MovieClip (oggetto)

I metodi dell'oggetto `MovieClip` forniscono le stesse funzionalità delle azioni standard applicabili a clip filmato. Vi sono inoltre metodi aggiuntivi a cui non corrispondono azioni analoghe nella casella degli strumenti Azioni del pannello Azioni.

Non è necessario usare una funzione di costruzione per richiamare i metodi dell'oggetto `MovieClip`, bensì è possibile fare riferimento alle istanze di clip filmato per nome, usando la sintassi seguente:

```
myMovieClip.play();  
myMovieClip.gotoAndPlay(3);
```

Riepilogo dei metodi validi per l'oggetto MovieClip

Metodo	Descrizione
<code>MovieClip.attachMovie</code>	Associa un filmato nella libreria.
<code>MovieClip.createEmptyMovieClip</code>	Crea un clip filmato vuoto.
<code>MovieClip.createTextField</code>	Crea un campo di testo vuoto.
<code>MovieClip.duplicateMovieClip</code>	Duplica il clip filmato specificato.
<code>MovieClip.getBounds</code>	Restituisce i valori minimo e massimo delle coordinate x e y di un filmato rispetto a uno spazio di coordinate specificato.
<code>MovieClip.getBytesLoaded</code>	Restituisce il numero di byte caricati per il clip filmato specificato.
<code>MovieClip.getBytesTotal</code>	Restituisce la dimensione del clip filmato in byte.
<code>MovieClip.getDepth</code>	Restituisce la profondità di un clip filmato.
<code>MovieClip.getURL</code>	Recupera un documento da un URL.
<code>MovieClip.globalToLocal</code>	Converte le coordinate di un punto da coordinate rispetto allo stage in coordinate locali rispetto al clip filmato specificato.
<code>MovieClip.gotoAndPlay</code>	Invia l'indicatore di riproduzione a un fotogramma specifico nel clip filmato e riproduce il filmato.
<code>MovieClip.gotoAndStop</code>	Invia l'indicatore di riproduzione a un fotogramma specifico nel clip filmato e interrompe la riproduzione del filmato.
<code>MovieClip.hitTest</code>	Restituisce <code>true</code> se il riquadro di limitazione del clip filmato specificato interseca il riquadro di limitazione del clip filmato target.
<code>MovieClip.loadMovie</code>	Carica il filmato specificato nel clip filmato.
<code>MovieClip.loadVariables</code>	Carica le variabili da un URL o da un'altra posizione, nel clip filmato.
<code>MovieClip.localToGlobal</code>	Converte le coordinate di un punto da coordinate locali rispetto al clip filmato in coordinate globali rispetto allo stage.
<code>MovieClip.nextFrame</code>	Invia l'indicatore di riproduzione al fotogramma successivo del clip filmato.
<code>MovieClip.play</code>	Riproduce il clip filmato specificato.
<code>MovieClip.prevFrame</code>	Invia l'indicatore di riproduzione al fotogramma precedente del clip filmato.
<code>MovieClip.removeMovieClip</code>	Rimuove il clip filmato dalla linea temporale, se esso è stato creato tramite un'azione o un metodo <code>duplicateMovieClip</code> o il metodo <code>attachMovie</code> .
<code>MovieClip.setMask</code>	Specifica un clip filmato come maschera per un altro clip filmato.
<code>MovieClip.startDrag</code>	Rende un clip filmato mobile e ne avvia il trascinamento.
<code>MovieClip.stop</code>	Interrompe il filmato attualmente in riproduzione.
<code>MovieClip.stopDrag</code>	Interrompe l'azione di trascinamento in corso di qualunque clip filmato.
<code>MovieClip.swapDepths</code>	Scambia il livello di profondità di due filmati.
<code>MovieClip.unloadMovie</code>	Rimuove un filmato caricato tramite l'azione <code>loadMovie</code> .

Riepilogo dei metodi Disegno validi per l'oggetto MovieClip

Metodo	Descrizione
<code>MovieClip.beginFill</code>	Inizia a disegnare un riempimento sullo stage.
<code>MovieClip.beginGradientFill</code>	Inizia a disegnare un riempimento sfumato sullo stage.
<code>MovieClip.clear</code>	Rimuove tutti i comandi per il disegno associati a un'istanza clip filmato.
<code>MovieClip.curveTo</code>	Disegna una curva usando l'ultimo stile linea adottato.
<code>MovieClip.endFill</code>	Completa il riempimento specificato tramite <code>beginFill</code> o <code>beginGradientFill</code> .
<code>MovieClip.lineTo</code>	Definisce il tratto delle linee create con i metodi <code>lineTo</code> e <code>curveTo</code> .
<code>MovieClip.lineTo</code>	Disegna una linea usando lo stile linea corrente.
<code>MovieClip.moveTo</code>	Sposta la posizione corrente del disegno alle coordinate specificate.

Riepilogo delle proprietà valide per l'oggetto MovieClip

Proprietà	Descrizione
<code>MovieClip._alpha</code>	Il valore di trasparenza dell'istanza di un clip filmato.
<code>MovieClip._currentframe</code>	Il numero di fotogramma in cui è posizionato l'indicatore di riproduzione.
<code>MovieClip._droptarget</code>	Il percorso assoluto, con sintassi della barra inclinata, dell'istanza del clip filmato su cui è stato rilasciato un clip filmato trascinabile.
<code>MovieClip.enabled</code>	Indica se un pulsante di clip filmato è attivato.
<code>MovieClip.focusEnabled</code>	Consente a un clip filmato di essere attivato.
<code>MovieClip._focusrect</code>	Indica se un clip filmato attivato è delimitato da un rettangolo giallo.
<code>MovieClip._framesloaded</code>	Il numero di fotogrammi caricati da un filmato in streaming.
<code>MovieClip._height</code>	L'altezza di un'istanza di clip filmato espressa in pixel.
<code>MovieClip.hitArea</code>	Individua un altro clip filmato come area di interazione per un pulsante di clip filmato.
<code>MovieClip._highquality</code>	Imposta la qualità di rendering di un filmato.
<code>MovieClip._name</code>	Il nome di un'istanza di clip filmato.
<code>MovieClip._parent</code>	Riferimento al clip filmato che racchiude il clip filmato.
<code>MovieClip._rotation</code>	Il grado di rotazione di un'istanza di clip filmato.
<code>MovieClip._soundbuftime</code>	Il numero di secondi prima dell'inizio dello streaming di un suono.
<code>MovieClip.tabChildren</code>	Indica se gli oggetti secondari di un clip filmato sono inclusi nell'ordinamento di tabulazione automatico.
<code>MovieClip.tabEnabled</code>	Indica se un clip filmato è incluso nell'ordinamento di tabulazione automatico.
<code>MovieClip.tabIndex</code>	Indica l'ordine di tabulazione di un oggetto.
<code>MovieClip._target</code>	Il percorso target di un'istanza di clip filmato.
<code>MovieClip._totalframes</code>	Il numero totale di fotogrammi in un'istanza di clip filmato.
<code>MovieClip.trackAsMenu</code>	Indica se altri pulsanti possono ricevere eventi associati al rilascio del pulsante del mouse.
<code>MovieClip._url</code>	L'URL del file SWF dal quale è stato scaricato il clip filmato.

Proprietà	Descrizione
<code>MovieClip.useHandCursor</code>	Determina se lo strumento Mano viene visualizzato quando l'utente posiziona il puntatore sul pulsante di un clip filmato.
<code>MovieClip._visible</code>	Valore booleano che determina se un'istanza di clip filmato è nascosta o visibile.
<code>MovieClip._width</code>	La larghezza di un'istanza di clip filmato espressa in pixel.
<code>MovieClip._x</code>	La coordinata x di un'istanza di clip filmato.
<code>MovieClip._xmouse</code>	La coordinata x del cursore all'interno di un'istanza di clip filmato.
<code>MovieClip._xscale</code>	Valore che specifica la percentuale di modifica in scala orizzontale di un clip filmato.
<code>MovieClip._y</code>	La coordinata y di un'istanza di clip filmato.
<code>MovieClip._ymouse</code>	La coordinata y del cursore all'interno di un'istanza di clip filmato.
<code>MovieClip._yscale</code>	Valore che specifica la percentuale di modifica in scala verticale di un clip filmato.

Riepilogo dei gestori di eventi validi per l'oggetto `MovieClip`

Proprietà	Descrizione
<code>MovieClip.onData</code>	Invocato quando tutti i dati vengono caricati in un clip filmato.
<code>MovieClip.onDragOut</code>	Invocato quando si seleziona il pulsante con il mouse e si trascina il puntatore fuori dall'area del pulsante.
<code>MovieClip.onDragOver</code>	Invocato quando il pulsante del mouse viene premuto all'esterno dell'area del pulsante e quindi il puntatore viene nuovamente trascinato sopra il pulsante.
<code>MovieClip.onEnterFrame</code>	Invocato costantemente con frequenza pari a quella dei fotogrammi del filmato. Le azioni associate all'evento di clip filmato <code>enterFrame</code> vengono eseguite prima di qualsiasi azione associata ai fotogrammi interessati.
<code>MovieClip.onKeyDown</code>	Invocato quando viene premuto un tasto. L'uso dei metodi <code>Key.getCode</code> e <code>Key.getAscii</code> consente di recuperare informazioni sull'ultimo tasto premuto.
<code>MovieClip.onKeyUp</code>	Invocato quando viene rilasciato un tasto.
<code>MovieClip.onKillFocus</code>	Invocato quando un pulsante viene disattivato.
<code>MovieClip.onLoad</code>	Invocato quando il clip filmato diventa un'istanza e appare nella linea temporale.
<code>MovieClip.onMouseDown</code>	Invocato quando il pulsante sinistro del mouse viene premuto.
<code>MovieClip.onMouseMove</code>	Invocato a ogni spostamento del mouse.
<code>MovieClip.onMouseUp</code>	Invocato quando il pulsante sinistro del mouse viene rilasciato.
<code>MovieClip.onPress</code>	Invocato quando il mouse viene premuto mentre il puntatore si trova sopra un pulsante.
<code>MovieClip.onRelease</code>	Invocato quando il mouse viene rilasciato mentre il puntatore si trova sopra un pulsante.
<code>MovieClip.onReleaseOutside</code>	Invocato quando il pulsante del mouse viene rilasciato mentre il puntatore si trova al di fuori dell'area del pulsante, dopo la pressione del pulsante quando il puntatore è all'interno di essa.
<code>MovieClip.onRollOut</code>	Invocato quando il puntatore del mouse viene spostato fuori dall'area del pulsante.
<code>MovieClip.onRollOver</code>	Invocato quando il puntatore del mouse scorre sopra il pulsante.

Proprietà	Descrizione
<code>MovieClip.onSetFocus</code>	Invocato quando un pulsante è l'elemento attivo e un tasto viene rilasciato.
<code>MovieClip.onUnload</code>	Invocato nel primo fotogramma dopo la rimozione del clip filmato dalla linea temporale. Le azioni associate all'evento clip filmato Unload vengono eseguite prima di qualsiasi azione associata al fotogramma interessato.

MovieClip._alpha

Disponibilità

Flash Player 4.

Uso

`myMovieClip._alpha`

Descrizione

Proprietà; imposta o recupera la trasparenza alfa (*valore*) del clip filmato specificato tramite *MovieClip*. I valori validi vanno da 0 (completamente trasparente) a 100 (completamente opaco). In un clip filmato in cui la proprietà `_alpha` è impostata su 0, gli oggetti sono attivi anche se sono invisibili. Ad esempio, è comunque possibile premere un pulsante in un clip filmato in cui la proprietà `_alpha` è impostata su 0.

Esempio

L'istruzione seguente imposta la proprietà `_alpha` del clip filmato `star` su 30% quando il pulsante viene premuto.

```
on(release) {  
    star._alpha = 30;  
}
```

MovieClip.attachMovie

Disponibilità

Flash Player 5.

Uso

`myMovieClip.attachMovie(nomeId, nuovoNome, profondità [, initObject])`

Parametri

nomeId Il nome di concatenamento del simbolo di clip filmato nella libreria da associare a un clip filmato sullo stage. Corrisponde al nome immesso nel campo Identificatore nella finestra di dialogo Proprietà di concatenamento del simbolo.

nuovoNome Nome univoco dell'istanza del clip filmato da associare al clip filmato.

profondità Numero intero che specifica il livello di profondità in cui posizionare il filmato.

initObject Oggetto contenente proprietà da destinare al nuovo clip filmato associato. Questo parametro consente ai clip filmato creati dinamicamente di ricevere parametri a cui essere associati. Se *initObject* non è un oggetto, esso viene ignorato. Tutte le proprietà di *initObject* vengono copiate nella nuova istanza. Le proprietà specificate tramite *initObject* sono disponibili per la funzione di costruzione. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; richiede un simbolo alla libreria e lo associa al filmato sullo stage specificato da *MovieClip*. Usare l'azione o metodo `removeMovieClip` o `unloadMovie` per rimuovere un filmato associato tramite il metodo `attachMovie`.

Esempio

Nell'esempio seguente, il simbolo con il "cerchietto" identificatore di concatenamento viene associato all'istanza del clip filmato che si trova sullo stage.

```
on (release) {
    thing.attachMovie( "cerchietto", "cerchietto1", 2 );
}
```

Vedere anche

`MovieClip.removeMovieClip`, `MovieClip.unloadMovie`, `Object.registerClass`, `removeMovieClip`

MovieClip.beginFill

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.beginFill ([rgb[, alfa]])
```

Parametro

rgb Valore esadecimale del colore (ad esempio, rosso corrisponde a 0xFF0000, blu a 0x0000FF e così via). Se questo valore non viene fornito o è di tipo "undefined", il riempimento non viene creato.

alfa Numero intero compreso tra 0 e 100 che specifica il valore alfa del riempimento. Se questo valore non viene fornito, viene usato 100 (uniforme). Se il valore è inferiore a 0, Flash userà 0. Se il valore è superiore a 100, Flash userà 100.

Valori restituiti

Nessuno.

Descrizione

Metodo; identifica la posizione iniziale del tracciato di un nuovo disegno. Se esiste già un tracciato aperto, ma la posizione corrente del disegno non corrisponde a quella specificata nel metodo `moveTo` a cui è associato un riempimento, essa viene chiusa con una linea e quindi riempita. Questa procedura è simile a quella eseguita quando viene richiamato il metodo `endFill`.

Vedere anche

`MovieClip.beginGradientFill`, `MovieClip.endFill`

MovieClip.beginGradientFill

Disponibilità

Flash Player 6.

Uso

myMovieClip.beginGradientFill (*tipoRiempimento*, *colori*, *alfa*, *rapporti*, *matrice*)

Parametro

tipoRiempimento La stringa "lineare" o la stringa "radiale".

colori Matrice di valori RGB esadecimali dei colori da usare nella sfumatura (ad esempio, rosso corrisponde a 0xFF0000, blu a 0x0000FF e così via).

alfa Matrice di valori alfa per i corrispondenti colori nella matrice *colori*; i valori validi sono compresi tra 0 e 100. Se il valore è inferiore a 0, Flash userà 0. Se il valore è superiore a 100, Flash userà 100.

rapporti Matrice per la distribuzione dei colori; i valori validi sono compresi tra 0 e 255. Questo valore definisce la percentuale di larghezza a cui il colore viene rapportato al 100%.

matrice Matrice di trasformazione; un oggetto contenente una delle due serie di proprietà riportate di seguito.

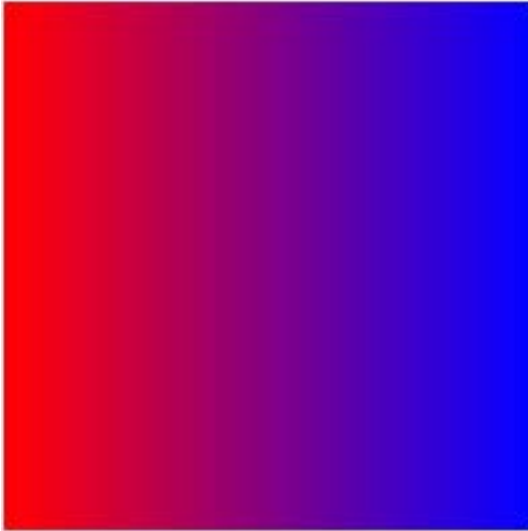
- *a, b, c, d, e, f, g, h, i* da usare per descrivere una matrice 3 x 3 della forma seguente:

```
a b c
d e f
g h i
```

Nell'esempio seguente, viene usato un metodo `beginGradientFill` con un parametro *matrice* che è un oggetto contenente queste proprietà.

```
_root.createEmptyMovieClip( "sfumatura", 1 );
    with ( _root.grad ) {
        {
            colors = [ 0xFF0000, 0x0000FF ];
            alphas = [ 100, 100 ];
            ratios = [ 0, 0xFF ];
            matrix = { a:200, b:0, c:0, d:0, e:200, f:0, g:200, h:200, i:1
        };
        beginGradientFill( "lineare", colori, alfa, rapporti, matrice
    );
        moveto(100,100);
        lineto(100,300);
        lineto(300,300);
        lineto(300,100);
        lineto(100,100);
        endFill();
    }
}
```

Se una proprietà *matrixType* non esiste, i restanti parametri sono tutti necessari; se uno di essi è mancante, la funzione risulterà errata. Questa matrice scala, trasla, ruota e inclina la sfumatura definita dai valori (-1,-1) e (1,1).



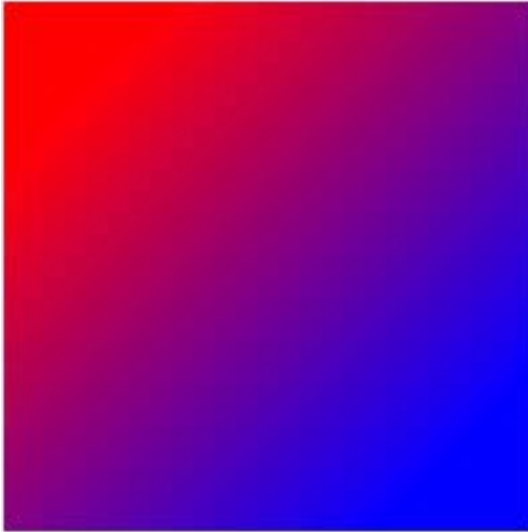
- *matrixType*, *x*, *y*, *w*, *h*, *r*.

La proprietà *matrixType* corrisponde alla stringa "casella", *x* alla posizione sull'asse orizzontale relativa al punto di registrazione del clip filmato principale per l'angolo superiore sinistro della sfumatura, *y* alla posizione sull'asse verticale del clip filmato principale relativo allo stesso punto di registrazione, *w* è la larghezza dell'area interessata dalla sfumatura, mentre *h* ne specifica l'altezza ed *r* definisce la sua rotazione in radianti.

Nell'esempio seguente, viene usato un metodo `beginGradientFill` con un parametro *matrice* che è un oggetto contenente queste proprietà.

```
_root.createEmptyMovieClip( "sfumatura", 1 );
    with ( _root.grad ) {
        {
            colors = [ 0xFF0000, 0x0000FF ];
            alphas = [ 100, 100 ];
            ratios = [ 0, 0xFF ];
            matrix = { matrixType:"casella", x:100, y:100, w:200, h:200,
r:(45/180)*Math.PI };
            beginGradientFill( "lineare", colori, alfa, rapporti, matrice
);
            moveto(100,100);
            lineto(100,300);
            lineto(300,300);
            lineto(300,100);
            lineto(100,100);
            endFill();
        }
    }
```

Se esiste una proprietà *matrixType*, essa deve essere uguale a "casella" e i restanti parametri sono tutti necessari. La funzione risulta errata se non vengono soddisfatte tutte queste condizioni.



Valori restituiti

Nessuno.

Descrizione

Metodo; identifica la posizione iniziale del tracciato di un nuovo disegno. Se il primo parametro è *undefined*, o se nessun parametro viene passato, il tracciato risulta privo di riempimento. Se esiste già un tracciato aperto, ma la posizione corrente del disegno non corrisponde a quella specificata nel metodo *moveTo* a cui è associato un riempimento, essa viene chiusa con una linea e quindi riempita. Questa procedura è simile a quella eseguita quando viene richiamato il metodo *endFill*.

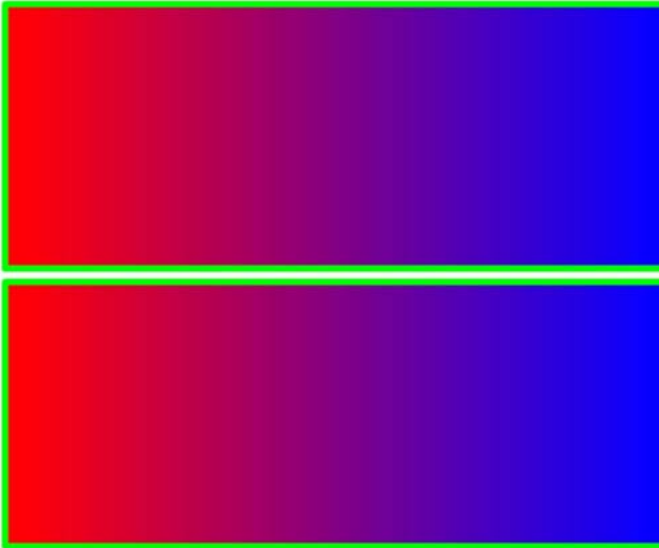
Tale metodo non funziona se si verifica una delle seguenti condizioni:

- Il numero di voci presenti nei parametri *colori*, *alfa* e *rapporti* differisce.
- Il parametro *tipoRiempimento* non è "lineare" né "radiale".
- Uno qualsiasi dei campi del parametro *matrice* nell'oggetto manca o non è valido.

Esempio

Il codice riportato di seguito usa entrambi i metodi per disegnare due rettangoli sovrapposti con un riempimento sfumato rosso-blu e un tratto verde uniforme dello spessore di 5 punti.

```
_root.createEmptyMovieClip("goober",1);
with ( _root.goober ) {
{
  colors = [ 0xFF0000, 0x0000FF ];
  alphas = [ 100, 100 ];
  ratios = [ 0, 0xFF ];
 LineStyle( 5, 0x00ff00 );
  matrix = { a:500,b:0,c:0,d:0,e:200,f:0,g:350,h:200,i:1};
  beginGradientFill( "lineare", colori, alfa, rapporti, matrice );
  moveto(100,100);
  lineto(100,300);
  lineto(600,300);
  lineto(600,100);
  lineto(100,100);
  endFill();
  matrice = { matrixType:"casella", x:100, y:310, w:500, h:200, r:(0/
180)*Math.PI };
  beginGradientFill( "lineare", colori, alfa, rapporti, matrice );
  moveto(100,310);
  lineto(100,510);
  lineto(600,510);
  lineto(600,310);
  lineto(100,310);
  endFill();
}
}
```



Vedere anche

[MovieClip.beginFill](#), [MovieClip.endFill](#), [MovieClip.lineStyle](#), [MovieClip.lineTo](#), [MovieClip.moveTo](#)

MovieClip.clear

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.clear()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove tutti i comandi per il disegno associati a un'istanza clip filmato. Forme e linee tracciate con gli strumenti di disegno di Flash rimangono invariati. Richiamando il metodo `clear` si rimuove lo stile linea corrente.

Vedere anche

`MovieClip.lineStyle`

MovieClip.createEmptyMovieClip

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.createEmptyMovieClip (nomeIstanza, profondità)
```

Parametro

nomeIstanza Stringa che identifica il nome istanza di un nuovo clip filmato.

profondità Numero intero che specifica la profondità del nuovo clip filmato.

Valori restituiti

Nessuno.

Descrizione

Metodo; crea un clip filmato vuoto come filmato secondario di un clip filmato esistente. Questo metodo è simile al metodo `attachMovie` ma non è necessario specificare un nome di concatenamento esterno per il nuovo clip filmato. Il punto di registrazione di un clip filmato vuoto appena creato è l'angolo superiore sinistro. Tale metodo non funziona se manca uno qualsiasi dei parametri.

Vedere anche

`MovieClip.attachMovie`

MovieClip.createTextField

Disponibilità

Flash Player 6.

Uso

myMovieClip.createTextField (*nomeIstanza*, *profondità*, *x*, *y*, *larghezza*, *altezza*)

Parametri

nomeIstanza Stringa che identifica il nome dell'istanza di un nuovo campo di testo.

profondità Numero intero positivo che specifica la profondità del nuovo campo di testo.

x Numero intero che specifica la coordinata x del nuovo campo di testo.

y Numero intero che specifica la coordinata y del nuovo campo di testo.

larghezza Numero intero positivo che specifica la larghezza del nuovo campo di testo.

altezza Numero intero positivo che specifica la altezza del nuovo campo di testo.

Valori restituiti

Nessuno.

Descrizione

Metodo; crea un nuovo campo di testo vuoto secondario rispetto al clip filmato specificato dal parametro *MovieClip*. È possibile usare il metodo `createTextField` per creare campi di testo durante la riproduzione di un filmato. Il campo di testo è posizionato a (*x*, *y*) e ha dimensioni *larghezza* per *altezza*. I parametri *x* e *y* sono relativi al clip filmato che contiene il campo secondario; questi parametri corrispondono alle proprietà `_x` e `_y` del campo di testo. I parametri *larghezza* e *altezza* corrispondono alle proprietà `_width` e `_height` dei campi di testo.

Le proprietà predefinite di un campo di testo sono le seguenti:

```
type = "dynamic",
border = false,
background = false,
password = false,
multiline = false,
html = false,
embedFonts = false,
variable = null,
maxChars = null+0
```

Un campo di testo creato tramite `createTextField` riceve il seguente oggetto `TextFormat`

predefinito:

```
font = "Times New Roman"  
size = 12  
textColor = 0x000000  
bold = false  
italic = false,  
underline = false  
url = ""  
target = ""  
align = "a sinistra"  
leftMargin = 0  
rightMargin = 0  
indent = 0  
leading = 0  
bullet = false  
tabStops = [] (matrice vuota)
```

Esempio

Nell'esempio seguente, viene creato un campo di testo con larghezza uguale a 300, altezza 100, una coordinata *x* pari a 100 e una coordinata *y* pari a 100, senza bordo, con il testo in rosso e sottolineato.

```
_root.createTextField("testoutente",1,100,100,300,100);  
mytext.multiline = true;  
mytext.wordWrap = true;  
mytext.border = false;  
  
myformat = new TextFormat();  
myformat.color = 0xff0000;  
myformat.bullet = false;  
myformat.underline = true;  
  
mytext.text = "questo è il primo testo di verifica dell'oggetto campo";  
mytext.setTextFormat(formatopersonale);
```

Vedere anche

`TextFormat` (oggetto)

MovieClip._currentframe

Disponibilità

Flash Player 4.

Uso

`myMovieClip._currentframe`

Descrizione

Proprietà (sola lettura); restituisce il numero del fotogramma in cui si trova attualmente l'indicatore di riproduzione nella linea temporale specificata da *MovieClip*.

Esempio

Nell'esempio seguente, viene usata la proprietà `_currentframe` per impostare l'avanzamento dell'indicatore di riproduzione del clip filmato `actionClip` di cinque fotogrammi rispetto alla posizione corrente.

```
actionClip.gotoAndStop(_currentframe + 5);
```


MovieClip.curveTo

Disponibilità

Flash Player 6.

Uso

myMovieClip.curveTo (controlloX, controlloY, ancoraggioX, ancoraggioY)

Parametri

controlloX Numero intero che specifica la posizione sull'asse orizzontale rispetto al punto di registrazione del clip filmato principale del punto di controllo.

controlloY Numero intero che specifica la posizione sull'asse verticale rispetto al punto di registrazione del clip filmato principale del punto di controllo.

ancoraggioX Numero intero che specifica la posizione sull'asse orizzontale rispetto al punto di registrazione del clip filmato principale del punto di ancoraggio successivo.

ancoraggioY Numero intero che specifica la posizione sull'asse verticale rispetto al punto di registrazione del clip filmato principale del punto di ancoraggio successivo.

Valori restituiti

Nessuno.

Descrizione

Metodi; disegna una curva dalla posizione corrente sino a (*ancoraggioX*, *ancoraggioY*) adottando lo stile linea corrente e usando il punto di controllo specificato da (*controlloX*, *controlloY*). La posizione corrente viene quindi impostata su (*ancoraggioX*, *ancoraggioY*). Se nel clip filmato in cui si sta disegnando sono presenti contenuti creati con gli strumenti di disegno di Flash, le chiamate a *curveTo* vengono tracciate al di sotto di tali contenuti. Se si richiama il metodo *curveTo* prima di *moveTo*, la posizione iniziale corrente viene automaticamente impostata sui valori predefiniti (0, 0). Se manca uno qualsiasi dei parametri, questo metodo non funziona e la posizione corrente non viene modificata.

Esempio

Nell'esempio seguente, viene tracciato un cerchio con una linea blu sottile e uniforme a cui viene applicato un riempimento rosso uniforme.

```
_root.createEmptyMovieClip( "cerchio", 1 );
con ( _root.circle )
{
    lineStyle( 0, 0x0000FF, 100 );
    beginFill( 0xFF0000 );
    moveTo( 500, 500 );
    curveTo( 600, 500, 600, 400 );
    curveTo( 600, 300, 500, 300 );
    curveTo( 400, 300, 400, 400 );
    curveTo( 400, 500, 500, 500 );
    endFill();
}
```

Vedere anche

MovieClip.beginFill, MovieClip.createEmptyMovieClip, MovieClip.endFill, MovieClip.lineStyle, MovieClip.lineTo, MovieClip.moveTo

MovieClip._droptarget

Disponibilità

Flash Player 4.

Uso

myMovieClip._droptarget

Descrizione

Proprietà (sola lettura); restituisce il percorso assoluto usando la sintassi della barra inclinata dell'istanza di clip filmato su cui è stato rilasciato *MovieClip*. La proprietà `_droptarget` restituisce sempre un percorso che inizia con una barra inclinata (/). Per confrontare la proprietà `_droptarget` di un'istanza con un riferimento, usare la funzione `eval` che consente di convertire il valore restituito dalla notazione che usa la barra inclinata in un riferimento che usa la sintassi del punto.

Esempio

L'esempio seguente valuta la proprietà `_droptarget` dell'istanza di clip filmato `garbage`, quindi usa la funzione `eval` per convertire il valore dalla notazione che usa la barra rovesciata in un riferimento che usa la sintassi del punto. Il riferimento `garbage` viene quindi confrontato con il riferimento all'istanza di clip filmato `trash`. Se i due riferimenti coincidono, la visibilità di `garbage` viene impostata su `false`. In caso contrario, l'istanza di `garbage` viene reimpostata sulla posizione originale.

```
if (eval(garbage._droptarget) == _root.trash) {  
    garbage._visible = false;  
} else {  
    garbage._x = x_pos;  
    garbage._y = y_pos;  
}
```

Le variabili `x_pos` e `y_pos` vengono impostate sul fotogramma 1 del filmato tramite lo script seguente:

```
x_pos = garbage._x;  
y_pos = garbage._y;
```

Vedere anche

`startDrag`

MovieClip.duplicateMovieClip

Disponibilità

Flash Player 5.

Uso

myMovieClip.duplicateMovieClip(nuovonome, profondità [,initObject])

Parametri

nuovonome Identificatore univoco per il clip filmato duplicato.

profondità Numero univoco che specifica il livello di profondità in cui posizionare il filmato specificato.

`initObject` Oggetto contenente proprietà da destinare al clip filmato duplicato. Questo parametro consente ai clip filmato creati dinamicamente di ricevere parametri a cui essere associati. Se `initObject` non è un oggetto, esso viene ignorato. Tutte le proprietà di `initObject` vengono copiate nella nuova istanza. Le proprietà specificate tramite `initObject` sono disponibili per la funzione di costruzione. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; crea un'istanza del clip filmato specificato durante la riproduzione del filmato. La riproduzione dei clip filmato duplicati viene avviata sempre a partire dal fotogramma 1, a prescindere dal fotogramma su cui si trovava il clip filmato originale quando è stato richiamato il metodo `duplicateMovieClip`. Le variabili nel clip filmato principale non vengono copiate nel clip filmato duplicato. I clip filmato creati mediante il metodo `duplicateMovieClip` non vengono duplicati se si richiama `duplicateMovieMethod` sui rispettivi clip filmato principali. L'eliminazione del clip filmato principale comporta l'eliminazione del clip filmato duplicato. I clip filmato aggiunti con il metodo `duplicateMovieClip` possono essere eliminati tramite l'azione o metodo `removeMovieClip`.

Vedere anche

`duplicateMovieClip`, `MovieClip.removeMovieClip`, `removeMovieClip`

MovieClip.enabled

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.enabled
```

Descrizione

Proprietà; valore booleano che indica se il clip filmato di un pulsante è attivato. Il valore predefinito di `enabled` è `true`. Se `enabled` viene impostato su `false`, i metodi di callback del clip filmato del pulsante e gli eventi associati alle azioni `on` non vengono più invocati e i fotogrammi `Sopra`, `Basso` e `Alto` vengono disattivati. La proprietà `enabled` non influisce sulla linea temporale del pulsante di clip filmato; se è in corso la riproduzione di un clip filmato, la riproduzione non viene interrotta. Il clip filmato continua a ricevere gli eventi di clip filmato, quali `mouseDown`, `mouseUp`, `keyDown` e `keyUp`.

La proprietà `enabled` regola esclusivamente le proprietà che definiscono il clip filmato di un pulsante. È possibile modificare la proprietà `enabled` in qualsiasi momento; il clip filmato del pulsante modificato viene attivato o disattivato immediatamente. La proprietà `enabled` è verificabile controllando un oggetto prototipo. Se questa proprietà è impostata su `false`, l'oggetto non è incluso nell'ordinamento di tabulazione automatico.

MovieClip.endFill

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.endFill()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; applica un riempimento alle linee e alle curve aggiunte dall'ultima volta in cui si è richiamato il metodo `beginFill` o `beginGradientFill`. Flash usa il tipo di riempimento specificato l'ultima volta che si è richiamato `beginFill` o `beginGradientFill`. Se la posizione corrente del disegno non corrisponde a quella specificata in precedenza mediante un metodo `moveTo` e il tipo di riempimento è stato definito, il tracciato viene chiuso con una linea e quindi riempito.

MovieClip.focusEnabled

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.focusEnabled
```

Descrizione

Proprietà; se il valore è `undefined` o `false`, un clip filmato può essere l'elemento attivo solo se è il clip filmato di un pulsante. Se invece il valore della proprietà `focusEnabled` è `true`, un clip filmato può essere l'elemento attivo anche se non è il clip filmato di un pulsante.

MovieClip._focusrect

Disponibilità

Flash Player 6.

Uso

```
myMovieClip._focusrect
```

Descrizione

Proprietà; valore booleano che specifica se un clip filmato attivato da tastiera è delimitato da un rettangolo giallo. Questa proprietà ha la priorità sulla proprietà globale `_focusrect`.

MovieClip._framesloaded

Disponibilità

Flash Player 4.

Uso

myMovieClip._framesloaded

Descrizione

Proprietà (sola lettura); il numero di fotogrammi che sono stati caricati da un filmato in streaming. Questa proprietà consente di determinare se il contenuto di un fotogramma specifico e di tutti i fotogrammi che lo precedono sono stati caricati e sono disponibili localmente nel browser. È una proprietà utile per monitorare il processo di scaricamento di filmati di grandi dimensioni. È possibile, ad esempio, visualizzare un messaggio che informa gli utenti sull'avanzamento del processo di caricamento del filmato fino al raggiungimento di un fotogramma specifico.

Esempio

Nell'esempio seguente, viene usata la proprietà `_framesloaded` per avviare un filmato quando tutti i fotogrammi sono stati caricati. Se non tutti i fotogrammi sono stati caricati, il valore della proprietà `_xscale` dell'istanza di clip filmato `loader` viene proporzionalmente incrementato per creare una barra di avanzamento.

```
if (_framesloaded >= _totalframes) {
    gotoAndPlay ("Scena 1", "avvio");
} else {
    _root.loader._xscale = (_framesloaded/_totalframes)*100;
}
```

MovieClip.getBounds

Disponibilità

Flash Player 5.

Uso

myMovieClip.getBounds(spazioCoordinateTarget)

Parametri

spazioCoordinateTarget Il percorso target della linea temporale le cui coordinate verranno usate come punto di riferimento.

Valori restituiti

Un oggetto con le proprietà `xMin`, `xMax`, `yMin` e `yMax`.

Descrizione

Metodo; restituisce i valori minimo e massimo delle coordinate *x* e *y* dell'oggetto `MovieClip` dell'istanza specificata da *MovieClip* per il parametro *spazioCoordinateTarget*.

Nota: usare i metodi `localToGlobal` e `globalToLocal` dell'oggetto `MovieClip` per convertire le coordinate locali del clip filmato in coordinate rispetto allo stage o viceversa, rispettivamente.

Esempio

Nell'esempio seguente, all'oggetto restituito dal metodo `getBounds` viene assegnato l'identificatore `clipBounds`. Sarà quindi possibile accedere ai valori di ciascuna proprietà e usarli in uno script. In questo script, un'altra istanza di clip filmato (`clip2`) viene collocata accanto a `clip`.

```
clipBounds = clip.getBounds(_root);  
clip2._x = clipBounds.xMax;
```

Vedere anche

`MovieClip.globalToLocal`, `MovieClip.localToGlobal`

MovieClip.getBytesLoaded

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.getBytesLoaded()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che indica il numero dei byte caricati.

Descrizione

Metodo; restituisce il numero di byte caricati (di cui si è effettuato lo streaming) per l'oggetto `MovieClip` specificato. È possibile confrontare il valore del metodo `getBytesLoaded` con quello del metodo `getBytesTotal` per determinare la percentuale di clip filmato caricata.

Vedere anche

`MovieClip.getBytesTotal`

MovieClip.getBytesTotal

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.getBytesTotal()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che indica la dimensione complessiva, in byte, dell'oggetto `MovieClip` specificato.

Descrizione

Metodo; restituisce la dimensione, in byte, dell'oggetto `MovieClip` specificato. Nel caso di clip filmato esterni (il filmato principale o un clip filmato che è stato caricato in un percorso target o in un livello), il valore restituito corrisponde alla dimensione del file SWF.

Vedere anche

`MovieClip.getBytesLoaded`

MovieClip.getDepth

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.getDepth
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce la profondità di un'istanza di clip filmato.

MovieClip.getURL

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.getURL(URL [,finestra, variabili]);
```

Parametri

URL L'URL presso cui è reperibile il documento.

finestra Parametro opzionale che specifica il nome, il frame o l'espressione che indica la finestra o il frame HTML in cui caricare il documento. È altresì possibile usare uno dei seguenti nomi target riservati: `_self` indica il frame corrente nella finestra corrente, `_blank` indica una finestra nuova, `_parent` indica il frame principale rispetto a quello corrente, `_top` indica il frame di primo livello nella finestra corrente.

variabili Parametro opzionale che specifica un metodo per l'invio delle variabili associate al filmato da caricare. Se non ci sono variabili, omettere questo parametro; in caso contrario, specificare se si intende caricare le variabili con il metodo GET o POST. GET aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. POST invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Metodo; carica un documento dall'URL specificato nella finestra indicata. Il metodo `getURL` consente inoltre di passare variabili a un'altra applicazione definita presso l'URL tramite il metodo GET o POST.

Vedere anche

`getURL`

MovieClip.globalToLocal

Disponibilità

Flash Player 5.

Uso

myMovieClip.globalToLocal(punto)

Parametri

punto Il nome o l'identificatore di un oggetto creato tramite l'oggetto generico Object che specifica le coordinate *x* e *y* come proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; converte le coordinate dell'oggetto *point* da coordinate rispetto allo stage (globali) a coordinate rispetto al clip filmato (locali).

Esempio

L'esempio seguente converte le coordinate globali *x* e *y* dell'oggetto *point* in coordinate locali del clip filmato.

```
onClipEvent(mouseMove) {
    point = new object();
    point.x = _root._xmouse;
    point.y = _root._ymouse;
    globalToLocal(punto);
    trace(_root._xmouse + " " + _root._ymouse);
    trace(point.x + " " + point.y);
    updateAfterEvent;
}
```

Vedere anche

`MovieClip.getBounds`, `MovieClip.localToGlobal`

MovieClip.gotoAndPlay

Disponibilità

Flash Player 5.

Uso

myMovieClip.gotoAndPlay(fotogramma)

Parametri

fotogramma Il numero di fotogramma a cui viene inviato l'indicatore di riproduzione.

Valori restituiti

Nessuno.

Descrizione

Metodo; avvia la riproduzione del filmato a partire dal fotogramma specificato.

Vedere anche

`gotoAndPlay`

MovieClip.gotoAndStop

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.gotoAndStop(fotogramma)
```

Parametri

fotogramma Il numero di fotogramma a cui viene inviato l'indicatore di riproduzione.

Valori restituiti

Nessuno.

Descrizione

Metodo; sposta l'indicatore di riproduzione al fotogramma specificato di un clip filmato e lo arresta in quella posizione.

Vedere anche

`gotoAndStop`

MovieClip._height

Disponibilità

Flash Player 4.

Uso

`myMovieClip._height`

Descrizione

Proprietà; imposta e recupera l'altezza di un clip filmato in pixel.

Esempio

L'esempio di codice seguente imposta l'altezza e la larghezza di un clip filmato quando l'utente fa clic con il mouse.

```
onClipEvent(mouseDown) {  
    _width=200;  
    _height=200;  
}
```

MovieClip._highquality

Disponibilità

Flash Player 6.

Uso

`myMovieClip._highquality`

Descrizione

Proprietà (globale); specifica il livello di antialiasing applicato al filmato corrente. Specificare 2 (Ottima) per applicare un livello di alta qualità con smussatura delle bitmap sempre attiva. Specificare 1 (Alta qualità) per applicare l'antialiasing; in tal modo le bitmap verranno smussate se il filmato non contiene animazioni. Specificare 0 (Bassa qualità) per impedire l'antialiasing. Questa proprietà può sovrascrivere la proprietà globale `_highquality`.

Esempio

```
myMovieClip._highquality = 1;
```

Vedere anche

`_quality`, `toggleHighQuality`

MovieClip.hitArea

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.hitArea
```

Valori restituiti

Un riferimento a un clip filmato.

Descrizione

Proprietà; specifica un altro clip filmato come area attiva per il clip filmato di un pulsante. Se la proprietà `hitArea` non esiste o ha valore `null` o `undefined`, il clip filmato del pulsante stesso viene usato come area attiva. Il valore della proprietà `hitArea` può essere un riferimento a un oggetto clip filmato.

È possibile modificare la proprietà `hitArea` in qualsiasi momento; il clip filmato del pulsante modificato assume immediatamente il comportamento di area attiva. Non è necessario che il clip filmato designato come area attiva sia visibile poiché l'attivazione della sua forma grafica viene comunque verificata. La proprietà `hitArea` è deducibile da un oggetto prototipo.

MovieClip.hitTest

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.hitTest(x, y, indicatoreForma);
```

```
myMovieClip.hitTest(target)
```

Parametri

x La coordinata *x* dell'area attiva sullo stage.

y La coordinata *y* dell'area attiva sullo stage.

Le coordinate *x* e *y* sono definite nello spazio di coordinate globali.

target Il percorso *target* dell'area attiva che potrebbe intersecare o sovrapporsi all'istanza specificata in *MovieClip*. Il *target* rappresenta di solito un pulsante o un campo per l'immissione di testo.

indicatoreForma Valore booleano che specifica se valutare l'intera forma dell'istanza specificata (*true*) oppure solo il riquadro di limitazione (*false*). Questo parametro può essere specificato solo se l'area attiva è identificata dalle coordinate *x* e *y*.

Valori restituiti

Nessuno.

Descrizione

Metodo; valuta l'istanza specificata in *MovieClip* per verificare se si sovrappone o interseca con l'area attiva identificata dall'argomento *target* o dalle coordinate *x* e *y*.

Nel primo caso confronta le coordinate *x* e *y* e la forma o il riquadro di limitazione dell'istanza specificata, a seconda dell'impostazione di *indicatoreForma*. Se *indicatoreForma* è impostato su *true*, viene valutata solo l'area correntemente occupata dall'istanza sullo stage, e se le coordinate *x* e *y* si sovrappongono in un punto qualsiasi, viene restituito il valore *true*. Ciò consente di determinare se il clip filmato si trova all'interno di un'area o di un punto attivo specificato.

Nel secondo caso valuta i riquadri di limitazione dell'istanza specificata e del *target*, quindi restituisce *true* se si sovrappongono o intersecano in un punto qualsiasi.

Esempio

Nell'esempio seguente, viene usato il metodo `hitTest` con le proprietà `x_mouse` e `y_mouse` per determinare se il mouse si trova sul riquadro di limitazione del target.

```
if (hitTest( _root._xmouse, _root._ymouse, false));
```

Nell'esempio seguente, viene usato il metodo `hitTest` per determinare se il clip filmato `ball` interseca o si sovrappone al clip filmato `square`.

```
if(_root.ball, hittest(_root.square)){  
    trace("ball interseca square");  
}
```

Vedere anche

`MovieClip.getBounds`, `MovieClip.globalToLocal`, `MovieClip.localToGlobal`

MovieClip.lineStyle

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.lineStyle ([spessore[, rgb[, alfa]])
```

Parametri

spessore Numero intero che indica lo spessore della linea in punti; i valori validi sono compresi tra 0 e 255. Se il numero non viene specificato o se il parametro risulta *undefined*, non viene tracciata alcuna linea. Se il valore passato è inferiore a 0, Flash userà 0. Il valore 0 indica lo spessore minimo; 255 quello massimo. Se il valore passato è superiore a 255, l'interprete di Flash userà 255.

rgb Valore esadecimale del colore della linea (ad esempio, rosso corrisponde a 0xFF0000, blu a 0x0000FF e così via). Se non viene indicato alcun valore, Flash userà 0x000000 (nero).

alfa Numero intero che indica il valore alfa (trasparenza) del colore della linea; i valori validi sono compresi tra 0 e 100. Se non viene indicato alcun valore, Flash userà 100 (nero). Se il valore è inferiore a 0, Flash userà 0; se il valore è superiore a 100, l'interprete di Flash userà 100.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica lo stile di una linea usato da Flash per le successive chiamate dei metodi `lineTo` e `curveTo` fino a quando non si richiama `lineStyle` specificando parametri differenti. È possibile richiamare il metodo `lineStyle` mentre si disegna un tracciato per specificare stili diversi da applicare alle varie linee all'interno del tracciato.

Nota: richiamando `clear` si reimposta il metodo `lineStyle` su `undefined`.

Esempio

Il codice riportato di seguito disegna un triangolo con una linea magenta uniforme e spessore pari a 5 punti senza applicare alcun riempimento.

```
_root.createEmptyMovieClip( "triangolo", 1 );
with ( _root.triangle )
{
    lineStyle( 5, 0xff00ff, 100 );
    moveTo( 200, 200 );
    lineTo( 300, 300 );
    lineTo( 100, 300 );
    lineTo( 200, 200 );
}
```

Vedere anche

`MovieClip.beginFill`, `MovieClip.beginGradientFill`, `MovieClip.clear`,
`MovieClip.curveTo`, `MovieClip.lineTo`, `MovieClip.moveTo`,

MovieClip.lineTo

Disponibilità

Flash Player 6.

Uso

myMovieClip.lineTo (x, y)

Parametri

x Numero intero che indica la posizione sull'asse orizzontale rispetto al punto di registrazione del clip filmato principale.

y Numero intero che indica la posizione sull'asse verticale rispetto al punto di registrazione del clip filmato principale.

Valori restituiti

Nessuno.

Descrizione

Metodo; disegna una linea dalla posizione corrente sino a (x, y) adottando lo stile linea corrente; la posizione corrente viene impostata su (x, y). Se nel clip filmato in cui si sta disegnando sono presenti contenuti creati con gli strumenti di disegno di Flash, le chiamate di `lineTo` vengono tracciate al di sotto di tali contenuti. Se si richiama il metodo `lineTo` prima di `moveTo`, la posizione iniziale corrente viene automaticamente impostata sui valori predefiniti (0, 0). Se manca uno qualsiasi dei parametri, questo metodo non funziona e la posizione corrente non viene modificata.

Esempio

Nell'esempio seguente, viene disegnato un triangolo senza linee di contorno con un riempimento blu parzialmente trasparente.

```
_root.createEmptyMovieClip ("triangolo", 1 );
with (_root.triangolo){
    beginFill (0x0000FF, 50);
    lineStyle (5, 0xFF00FF, 100);
    moveTo (200, 200);
   .lineTo (300, 300);
   .lineTo (100, 300);
   .lineTo (200, 200);
    endFill();
}
```

Vedere anche

MovieClip.beginFill, MovieClip.createEmptyMovieClip, MovieClip.endFill, MovieClip.lineStyle, MovieClip.moveTo

MovieClip.loadMovie

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.loadMovie(url [,variabili]);
```

Parametri

url URL assoluto o relativo per il file SWF o JPEG da caricare. Il percorso relativo deve riferirsi al file SWF a `_level0`. Tale URL deve trovarsi nello stesso sottodominio dell'URL che contiene il filmato. Per essere usati nella versione autonoma di Flash Player o verificati in modalità di prova filmato nell'applicazione di creazione Flash, tutti i file SWF devono essere memorizzati nella stessa cartella e i nomi dei file non possono includere specifiche della cartella o dell'unità disco.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio o il caricamento di variabili. Il parametro deve essere costituito dalla stringa GET o POST. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo GET aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo POST invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Metodo; carica file SWF o JPEG in un clip filmato in Flash Player durante la riproduzione del filmato originale. Senza il `loadMovie`, viene visualizzato un solo filmato (file SWF), quindi Flash Player si chiude. Il metodo `loadMovie` consente di visualizzare più filmati contemporaneamente o di passare da un filmato all'altro senza dover caricare un altro documento HTML.

Un'immagine o un filmato caricato in un clip filmato ne eredita le proprietà di posizione, rotazione e scala. È possibile usare il percorso `target` del clip filmato per identificare il filmato caricato.

Usare il metodo `unloadMovie` per rimuovere i filmati o le immagini caricati con il metodo `loadMovie`. Usare il metodo `loadVariables` per mantenere il filmato attivo e aggiornare le variabili con i nuovi valori.

Vedere anche

`loadMovie`, `loadMovieNum`, `MovieClip.loadVariables`, `MovieClip.unloadMovie`, `unloadMovie`, `unloadMovieNum`

MovieClip.loadVariables

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.loadVariables("url", variabili);
```

Parametri

url URL assoluto o relativo per il file esterno che contiene le variabili da caricare. L'host dell'URL deve trovarsi nello stesso sottodominio del clip filmato.

variabili Parametro opzionale che specifica un metodo HTTP per l'invio di variabili. Il parametro deve essere costituito dalla stringa `GET` o `POST`. Se non ci sono variabili da inviare, omettere questo parametro. Il metodo `GET` aggiunge le variabili alla fine dell'URL e si usa in caso di un numero limitato di variabili. Il metodo `POST` invia le variabili in un'intestazione HTTP separata e si usa in caso di stringhe di variabili di notevole lunghezza.

Valori restituiti

Nessuno.

Descrizione

Metodo; legge i dati da un file esterno e imposta i valori delle variabili in *MovieClip*. Il file esterno può essere un file di testo creato tramite script CGI, Active Server Pages (ASP) o Personal Home Page (PHP) e può contenere un numero illimitato di variabili.

Questo metodo consente anche di aggiornare le variabili nel clip filmato attivo assegnando ad esse nuovi valori.

È necessario che il testo dell'URL sia in formato MIME standard: *application/x-www-form-urlencoded* (script in formato CGI).

Vedere anche

`loadMovie`, `loadVariables`, `loadVariablesNum`, `MovieClip.unloadMovie`

MovieClip.localToGlobal

Disponibilità

Flash Player 5.

Uso

myMovieClip.localToGlobal(punto)

Parametri

punto Il nome o l'identificatore di un oggetto creato tramite l'oggetto Object che specifica le coordinate *x* e *y* come proprietà.

Valori restituiti

Nessuno.

Descrizione

Metodo; converte le coordinate dell'oggetto *point* da coordinate rispetto al clip filmato (locali) in coordinate rispetto allo stage (globali).

Esempio

Nell'esempio seguente, le coordinate *x* e *y* dell'oggetto *point* vengono convertite da coordinate del clip filmato (locali) in coordinate dello stage (globali). Le coordinate locali *x* e *y* vengono specificate tramite le proprietà *_xmouse* e *_ymouse* in modo da recuperare le coordinate *x* e *y* della posizione del mouse.

```
onClipEvent(mouseMove) {
    point = new object();
    point.x = _root._xmouse;
    point.y = _root._ymouse;
    _root.out3 = point.x + " === " + point.y;
    _root.out = _root._xmouse + " === " + _root._ymouse;
    localToGlobal(punto);
    _root.out3 = point.x + " === " + point.y;
    updateAfterEvent;
}
```

Vedere anche

[MovieClip.globalToLocal](#)

MovieClip.moveTo

Disponibilità

Flash Player 6.

Uso

myMovieClip.moveTo(x, y)

Parametri

x Numero intero che indica la posizione sull'asse orizzontale rispetto al punto di registrazione del clip filmato principale.

y Numero intero che indica la posizione sull'asse verticale rispetto al punto di registrazione del clip filmato principale.

Valori restituiti

Nessuno.

Descrizione

Metodo; sposta la posizione corrente del disegno a (x, y). Se manca uno qualsiasi dei parametri, questo metodo non funziona e la posizione corrente non viene modificata.

Esempio

Nell'esempio seguente, viene disegnato un triangolo con una linea magenta uniforme di spessore pari a 5 punti senza applicare alcun riempimento. La prima linea crea un clip filmato vuoto con cui disegnare. All'interno dell'istruzione `with` viene definito un tipo di linea, mentre la posizione iniziale del disegno è indicata dal metodo `moveTo`.

```
_root.createEmptyMovieClip( "triangolo", 1 );
with ( _root.triangle )
{
 LineStyle( 5, 0xff00ff, 100 );
  moveTo( 200, 200 );
  lineTo( 300, 300 );
  lineTo( 100, 300 );
  lineTo( 200, 200 );
}
```

Vedere anche

`MovieClip.createEmptyMovieClip`, `MovieClip.LineStyle`, `MovieClip.lineTo`

MovieClip._name

Disponibilità

Flash Player 4.

Uso

`myMovieClip._name`

Descrizione

Proprietà; restituisce il nome dell'istanza del clip filmato specificato mediante *MovieClip*.

MovieClip.nextFrame

Disponibilità

Flash Player 5.

Uso

`myMovieClip.nextFrame()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; invia l'indicatore di riproduzione al fotogramma successivo e lo arresta in tale posizione.

Vedere anche

`nextFrame`

MovieClip.onData

Disponibilità

Flash Player 6.

Uso

myMovieClip.onData

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un clip filmato riceve dati in seguito al richiamo di `loadVariables` o `loadMovie`.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onData` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onData");  
};
```

MovieClip.onDragOut

Disponibilità

Flash Player 6.

Uso

myMovieClip.onDragOver

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore viene premuto, trascinato all'esterno e quindi nuovamente sul clip filmato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onDragOut` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onDragOut");  
};
```

Vedere anche

`MovieClip.onDragOver`

MovieClip.onDragOver

Disponibilità

Flash Player 6.

Uso

myMovieClip.onDragOver

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore viene premuto, trascinato all'esterno e quindi nuovamente sul clip filmato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onDragOver` che invia un'azione `trace` alla finestra `Output`.

```
myMovieClip.onData = function () {  
    trace ("richiamato onDragOver");  
};
```

Vedere anche

`MovieClip.onDragOut`

MovieClip.onEnterFrame

Disponibilità

Flash Player 6.

Uso

myMovieClip.onEnterFrame

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato costantemente con frequenza pari a quella dei fotogrammi del filmato. Le azioni associate all'evento di clip filmato `enterFrame` vengono eseguite prima di qualsiasi azione associata ai fotogrammi interessati.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onEnterFrame` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onEnterFrame");  
};
```

MovieClip.onKeyDown

Disponibilità

Flash Player 6.

Uso

myMovieClip.onKeyDown

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un clip filmato è l'elemento attivo e un tasto viene premuto. L'evento `onKeyDown` viene invocato senza parametri. È possibile usare i metodi `Key.getAscii` e `Key.getCode` per determinare quale tasto è stato premuto.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onDragDown` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onKeyDown");  
};
```

Vedere anche

`MovieClip.onKeyUp`

MovieClip.onKeyUp

Disponibilità

Flash Player 6.

Uso

myMovieClip.onKeyUp

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Evento; invocato quando viene rilasciato un tasto. L'evento `onKeyUp` viene invocato senza parametri. È possibile usare i metodi `Key.getAscii` e `Key.getCode` per determinare quale tasto è stato premuto.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onKeyPress` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onKeyUp");  
};
```

MovieClip.onKillFocus

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onKillFocus = function (nuovoAttivo) {  
    istruzioni;  
};
```

Parametri

nuovoAttivo L'oggetto che viene attivato tramite tastiera.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; evento invocato quando un clip filmato non è più attivato dalla tastiera. Il metodo `onKillFocus` riceve il parametro *nuovoObiettivo*, che corrisponde a un oggetto rappresentante il nuovo oggetto attivato. Se nessun oggetto viene attivato, il valore di *nuovoObiettivo* è `null`.

MovieClip.onLoad

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onLoad
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il clip filmato diventa un'istanza e viene visualizzato nella linea temporale.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onLoad` che invia un'azione trace alla finestra Output.

```
myMovieClip.onData = function () {  
    trace ("richiamato onLoad");  
};
```

MovieClip.onMouseDown

Disponibilità

Flash Player 6.

Uso

myMovieClip.onMouseDown

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il pulsante sinistro del mouse viene premuto.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onMouseDown` che invia un'azione trace alla finestra Output.

```
someListener.onMouseDown = function () { ... };  
    trace ("richiamato onMouseDown");  
}
```

MovieClip.onMouseMove

Disponibilità

Flash Player 6.

Uso

myMovieClip.onMouseMove

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il mouse viene spostato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onMouseMove` che invia un'azione `trace` alla finestra `Output`.

```
someListener.onMouseDown = function () { ... };
    trace ("richiamato onMouseMove");
};
```

MovieClip.onMouseUp

Disponibilità

Flash Player 6.

Uso

myMovieClip.onMouseUp

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il pulsante sinistro del mouse viene rilasciato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onMouseUp` che invia un'azione `trace` alla finestra `Output`.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onMouseUp");
};
```

MovieClip.onPress

Disponibilità

Flash Player 6.

Uso

myMovieClip.onPress

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando si fa clic su un clip filmato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onPress` che invia un'azione trace alla finestra `Output`.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onPress");
};
```

MovieClip.onRelease

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onRelease
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il pulsante di clip filmato viene rilasciato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onPress` che invia un'azione trace alla finestra `Output`.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onRelease");
};
```

MovieClip.onReleaseOutside

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onReleaseOutside
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il pulsante del mouse viene rilasciato mentre il puntatore si trova all'esterno del clip filmato dopo essere stato premuto all'interno di esso.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onReleaseOutside` che invia un'azione trace alla finestra Output.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onReleaseOutside");
};
```

MovieClip.onRollOut

Disponibilità

Flash Player 6.

Uso

myMovieClip.onRollOut

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore viene spostato all'esterno dell'area di un clip filmato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onRollOut` che invia un'azione trace alla finestra Output.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onRollOut");
};
```

MovieClip.onRollOver

Disponibilità

Flash Player 6.

Uso

myMovieClip.onRollOver

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il puntatore scorre sopra l'area di un clip filmato.

È necessario definire una funzione che venga eseguita quando l'evento viene invocato.

Esempio

Nell'esempio seguente viene definita una funzione per il metodo `onRollOver` che invia un'azione trace alla finestra `Output`.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onRollOver");
};
```

MovieClip.onSetFocus

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onSetFocus = function(vecchioAttivo){
    istruzioni;
};
```

Parametri

vecchioAttivo L'oggetto da disattivare.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un clip filmato viene attivato tramite tastiera. Il parametro *vecchioAttivo* si riferisce all'oggetto attivato precedentemente da tastiera. Ad esempio, se si preme il tasto `Tab` per trasferire lo stato attivo da un clip filmato a un campo di testo, *vecchioAttivo* contiene l'istanza del clip filmato.

Se precedentemente non è stato attivato alcun oggetto, *vecchioAttivo* contiene un valore `null`.

MovieClip.onUnload

Disponibilità

Flash Player 6.

Uso

```
myMovieClip.onUnload
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento: invocato nel primo fotogramma dopo la rimozione del clip filmato dalla linea temporale. Le azioni associate all'evento `onUnload` vengono eseguite prima di qualsiasi azione associata al fotogramma interessato. È necessario definire una funzione che venga eseguita quando tale evento viene invocato.

Esempio

Nell'esempio seguente, viene definita una funzione per il metodo `onUnload` che invia un'azione trace alla finestra `Output`.

```
someListener.onMouseUp = function () { ... };
    trace ("richiamato onUnload");
};
```

MovieClip._parent

Disponibilità

Flash Player 6.

Uso

```
myMovieClip._parent.property
_parent._parent.property
```

Descrizione

Proprietà; specifica o restituisce un riferimento al clip filmato contenente l'oggetto o il clip filmato corrente. L'oggetto corrente è quello contenente il codice ActionScript che fa riferimento a `_parent`. Usare la proprietà `_parent` per specificare il percorso relativo a clip filmato oppure a oggetti che si trovano sopra l'oggetto o il clip filmato attuale.

Vedere anche

`_root`, `targetPath`

MovieClip.play

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.play();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; sposta l'indicatore di riproduzione lungo la linea temporale del clip filmato.

Vedere anche

`play`

MovieClip.prevFrame

Disponibilità

Flash Player 5.

Uso

myMovieClip.prevFrame()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; invia l'indicatore di riproduzione al fotogramma precedente e lo arresta in tale posizione.

Vedere anche

prevFrame

MovieClip.removeMovieClip

Disponibilità

Flash Player 5.

Uso

myMovieClip.removeMovieClip()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove un'istanza di clip filmato creata con l'azione `duplicateMovieclip` oppure con i metodi `duplicateMovieClip` o `attachMovie` dell'oggetto `MovieClip`.

Vedere anche

`MovieClip.attachMovie`, `MovieClip.attachMovie`, `removeMovieClip`, `MovieClip.attachMovie`

MovieClip._rotation

Disponibilità

Flash Player 4.

Uso

myMovieClip._rotation

Descrizione

Proprietà; specifica la rotazione del clip filmato in gradi.

MovieClip.setMask

Disponibilità

Flash Player 6.

Uso

myMovieClip.setMask (mascheraMovieClip)

Parametri

myMovieClip Il nome dell'istanza di clip filmato da mascherare.

mascheraMovieClip Il nome dell'istanza di clip filmato da usare come maschera.

Valori restituiti

Nessuno.

Descrizione

Metodo; usa il clip filmato indicato nel parametro *mascheraMovieClip* come maschera per rivelare il clip filmato specificato nel parametro *myMovieClip*.

Il metodo `setMask` consente di usare come maschere clip filmato multifotogramma con contenuto multilivello complesso. È possibile aprire e chiudere le maschere in fase di esecuzione, ma non è possibile usare la stessa maschera per mascherature multiple (consentite invece se si usano i livelli maschera). Se in un clip filmato mascherato sono presenti dei caratteri dispositivo, essi vengono visualizzati ma non mascherati. Non è possibile usare un clip filmato come maschera di se stesso, ad esempio `mc.setMask(mc)`.

Se si crea un livello maschera che contiene un clip filmato e successivamente si applica il metodo `setMask`, il metodo `setMask` acquisisce priorità e non è reversibile. Ad esempio, è possibile avere un clip filmato in un livello maschera denominato `UIMask` che maschera un altro livello contenente un clip filmato denominato `IMaskee`. Se, durante la riproduzione del filmato, si richiama `UIMask.setMask(UIMaskee)`, da quel momento in poi `UIMask` viene mascherato da `UIMaskee`.

Per cancellare una maschera creata con `ActionScript`, passare il valore `null` al metodo `setMask`. Il codice riportato di seguito cancella la maschera senza modificare il livello maschera sulla linea temporale.

```
UIMask.setMask(null)
```

Esempio

Nel codice riportato di seguito, viene usato il clip filmato `circleMask` per mascherare il clip filmato `theMaskee`.

```
theMaskee.setMask(mascheraCircolare);
```

MovieClip._soundbuftime

Disponibilità

Flash Player 6.

Uso

myMovieClip._soundbuftime

Descrizione

Proprietà (globale); numero intero che specifica in secondi il tempo di prebuffer di un suono prima che abbia inizio lo streaming.

MovieClip.startDrag

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.startDrag([bloccato [sinistra , superiore , destra, inferiore]])
```

Parametri

bloccato Un valore booleano che specifica se il clip filmato mobile è bloccato al centro rispetto alla posizione del mouse (*true*) oppure ancorato al punto del clip filmato selezionato inizialmente dall'utente (*false*). Questo parametro è opzionale.

sinistra, superiore, destra, inferiore Valori relativi alle coordinate del filmato principale del clip filmato che definiscono un rettangolo di delimitazione per quest'ultimo. Si tratta di parametri opzionali.

Valori restituiti

Nessuno.

Descrizione

Metodo; consente il trascinamento da parte dell'utente del clip filmato specificato. Il filmato è mobile fino a quando viene interrotto esplicitamente richiamando il metodo `stopDrag` o fino a quando viene reso mobile un altro clip filmato. È possibile trascinare un solo clip filmato alla volta.

Vedere anche

`MovieClip._droptarget`, `MovieClip.startDrag`, `MovieClip.stopDrag`

MovieClip.stop

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.stop()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; interrompe il clip filmato attualmente in riproduzione.

Vedere anche

`stop`

MovieClip.stopDrag

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.stopDrag()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; termina un metodo `startDrag`. Un filmato reso trascinabile con il metodo `startDrag` rimane tale finché non viene aggiunto un metodo `stopDrag` o finché un altro clip filmato non viene reso trascinabile. È possibile trascinare un solo clip filmato alla volta.

Vedere anche

`MovieClip._droptarget`, `MovieClip.startDrag`, `stopDrag`

MovieClip.swapDepths

Disponibilità

Flash Player 5.

Uso

```
myMovieClip.swapDepths(profondità)
```

```
myMovieClip.swapDepths(target)
```

Parametri

target L'istanza di clip filmato la cui profondità viene sostituita con quella dell'istanza specificata in *MovieClip*. Le due istanze devono avere lo stesso clip filmato principale.

profondità Numero che specifica il livello di profondità in cui posizionare il filmato *MovieClip*.

Valori restituiti

Nessuno.

Descrizione

Metodo; scambia la posizione di impilamento od ordine *z* (livello di profondità) dell'istanza specificata (*MovieClip*) con quella del filmato specificato nel parametro *target* o con il filmato che occupa attualmente il livello di profondità specificato nel relativo parametro *profondità*. I due filmati devono avere lo stesso clip filmato principale. Lo scambio del livello di profondità tra i clip filmato implica lo spostamento di un filmato in primo o in secondo piano rispetto all'altro. Eventuali interpolazioni del filmato in corso quando si richiama questo metodo verranno interrotte.

Vedere anche

`_level`

MovieClip.tabChildren

Disponibilità

Flash Player 6.

Uso

myMovieClip.tabChildren

Descrizione

Proprietà; `undefined` per impostazione predefinita. Se `tabChildren` risulta `undefined` o `true`, gli elementi secondari di un clip filmato vengono inclusi nell'ordinamento di tabulazione automatico. Se invece il valore di `tabChildren` è `false`, gli elementi secondari non vengono inclusi nell'ordinamento.

Esempio

Un widget d'interfaccia utente costituito da una casella di riepilogo e realizzato come clip filmato contiene diversi elementi, ciascuno dei quali può essere selezionato e attivato come pulsante. In ogni caso, soltanto la casella di riepilogo può essere usata come posizione di tabulazione. Gli elementi contenuti nella casella di riepilogo dovrebbero essere esclusi dall'ordinamento di tabulazione. A tal fine, la proprietà `tabChildren` della casella di riepilogo dovrebbe essere impostata su `false`.

Nel caso venga usata la proprietà `tabIndex`, la proprietà `tabChildren` influirà unicamente sull'ordinamento di tabulazione automatico.

Vedere anche

`Button.tabIndex`, `TextField.tabIndex`

MovieClip.tabEnabled

Disponibilità

Flash Player 6.

Uso

MovieClip.tabEnabled

Descrizione

Proprietà; è possibile impostarla su un'istanza degli oggetti `MovieClip`, `Button` o `TextField`. Risulta `undefined` per impostazione predefinita.

Se la proprietà `tabEnabled` risulta `undefined` o `true`, l'oggetto a cui essa è associata viene incluso nell'ordinamento di tabulazione automatico. Anche nel caso in cui la proprietà `tabIndex` è impostata su un dato valore, l'oggetto viene incluso nell'ordinamento personalizzato. Se `tabEnabled` è `false`, l'oggetto non viene incluso nell'ordinamento di tabulazione automatico. Nel caso di un clip filmato, se `tabEnabled` è `false`, gli elementi secondari del clip filmato possono essere ancora inclusi nell'ordinamento di tabulazione automatico, a meno che anche la proprietà `tabChildren` non sia impostata su `false`.

Vedere anche

`MovieClip.tabChildren`, `MovieClip.tabIndex`

MovieClip.tabIndex

Disponibilità

Flash Player 6.

Uso

myMovieClip.tabIndex

Descrizione

Proprietà; consente di personalizzare l'ordinamento di tabulazione degli oggetti in un filmato. La proprietà `tabIndex` risulta `undefined` per impostazione predefinita. È possibile attribuire `tabIndex` a un'istanza associata a un pulsante, a un clip filmato o a un campo di testo.

Se un oggetto in un filmato Flash contiene una proprietà `tabIndex`, l'ordinamento di tabulazione automatico viene disattivato e l'ordinamento di tabulazione viene calcolato in base alle proprietà `tabIndex` degli oggetti contenuti nel filmato. L'ordinamento di tabulazione personalizzato riguarda solo gli oggetti che hanno proprietà `tabIndex`.

La proprietà `tabIndex` deve essere un numero intero positivo. Gli oggetti vengono ordinati in base alle relative proprietà `tabIndex` in ordine ascendente. Un oggetto il cui valore `tabIndex` equivale a 1 precederà un oggetto con valore `tabIndex` pari a 2. L'ordinamento di tabulazione personalizzato non tiene conto delle relazioni gerarchiche tra gli oggetti di un filmato Flash. Tutti gli oggetti di un filmato Flash con proprietà `tabIndex` vengono posizionati nell'ordine di tabulazione. Non usare lo stesso valore `tabIndex` per più oggetti.

MovieClip._target

Disponibilità

Flash Player 4.

Uso

myMovieClip._target

Descrizione

Proprietà (sola lettura); restituisce il percorso target dell'istanza di clip filmato specificata nel parametro *ClipFilmato*.

MovieClip._totalframes

Disponibilità

Flash Player 4.

Uso

myMovieClip._totalframes

Descrizione

Proprietà (sola lettura); restituisce il numero totale di fotogrammi dell'istanza di clip filmato specificata nel parametro *ClipFilmato*.

MovieClip.trackAsMenu

Disponibilità

Flash Player 6.

Uso

myMovieClip.trackAsMenu

Descrizione

Proprietà; valore booleano che indica se altri pulsanti o clip filmato possono o meno ricevere eventi associati al rilascio del pulsante del mouse. Questa proprietà consente di creare dei menu. È possibile attribuire la proprietà `trackAsMenu` a qualsiasi pulsante e oggetto di clip filmato. Se la proprietà `trackAsMenu` non esiste, il comportamento predefinito è `false`.

È possibile modificare la proprietà `trackAsMenu` in qualsiasi momento; il pulsante di clip filmato modificato assume immediatamente il nuovo comportamento.

Vedere anche

`Button.trackAsMenu`

MovieClip.unloadMovie

Disponibilità

Flash Player 5.

Uso

myMovieClip.unloadMovie()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove un clip filmato caricato con i metodi `loadMovie` o `attachMovie` dell'oggetto `MovieClip`.

Vedere anche

`MovieClip.attachMovie`, `MovieClip.loadMovie`, `unloadMovie`, `unloadMovieNum`

MovieClip._url

Disponibilità

Flash Player 4.

Uso

myMovieClip._url

Descrizione

Proprietà (sola lettura); recupera l'URL dal file SWF dal quale è stato scaricato il clip filmato.

MovieClip.useHandCursor

Disponibilità

Flash Player 6.

Uso

myMovieClip.useHandCursor

Descrizione

Proprietà; valore booleano che indica se il cursore Mano viene visualizzato quando il puntatore viene fatto scorrere sul clip filmato di un pulsante. Il valore predefinito di `useHandCursor` è `true`. Se `useHandCursor` è impostato su `true`, il cursore Mano standard usato per i pulsanti viene visualizzato quando il puntatore scorre sul clip filmato di un pulsante. Se invece `useHandCursor` risulta `false`, viene usato il cursore freccia.

È possibile modificare la proprietà `useHandCursor` in qualsiasi momento; il pulsante di clip filmato modificato assume immediatamente il comportamento del nuovo cursore. La proprietà `useHandCursor` è deducibile da un oggetto prototipo.

MovieClip._visible

Disponibilità

Flash Player 4.

Uso

myMovieClip._visible

Descrizione

Proprietà; valore booleano che indica se il filmato specificato mediante il parametro *ClipFilmato* è visibile. I clip filmato non visibili (proprietà `_visible` impostata su `false`) sono disattivati. Ad esempio, un pulsante in un clip filmato con la proprietà `_visible` impostata su `false` non può essere premuto.

MovieClip._width

Disponibilità

Flash Player 4 come proprietà di sola lettura.

Uso

myMovieClip._width

Descrizione

Proprietà; imposta e recupera la larghezza di un clip filmato in pixel.

Esempio

L'esempio di codice seguente imposta le proprietà di altezza e larghezza di un clip filmato quando l'utente fa clic con il mouse.

```
onClipEvent(mouseDown) {  
    _width=200;  
    _height=200;  
}
```

Vedere anche

`MovieClip._height`

MovieClip._x

Disponibilità

Flash Player 3.

Uso

`myMovieClip._x`

Descrizione

Proprietà; numero intero che imposta la coordinata x del filmato rispetto alle coordinate locali del clip filmato principale. Se un clip filmato è nella linea temporale principale, allora il sistema di coordinate considera l'angolo superiore sinistro dello stage come il punto (0, 0). Se il clip filmato è all'interno di un altro clip filmato a cui sono state applicate trasformazioni, il clip si trova nel sistema di coordinate locali del clip che lo contiene. Ad esempio, se un clip filmato è ruotato di 90° in senso antiorario, il clip filmato secondario eredita un sistema di coordinate ruotato di 90° in senso antiorario. Le coordinate del clip filmato si riferiscono alla posizione del punto di registrazione.

Vedere anche

`MovieClip._xscale`, `MovieClip._y`, `MovieClip._yscale`

MovieClip._xmouse

Disponibilità

Flash Player 5.

Uso

`myMovieClip._xmouse`

Descrizione

Proprietà (sola lettura); restituisce la coordinata x della posizione del mouse.

Vedere anche

`Mouse` (oggetto), `MovieClip._ymouse`

MovieClip._xscale

Disponibilità

Flash Player 4.

Uso

`myMovieClip._xscale`

Descrizione

Proprietà; determina la scala orizzontale (*percentuale*) del clip filmato applicata dal punto di registrazione del clip filmato. Il punto di registrazione predefinito è (0,0).

Il ridimensionamento del sistema di coordinate locale influisce sull'impostazione delle proprietà `_x` e `_y` che sono definite in pixel interi. Se, ad esempio, il clip filmato principale è ridimensionato al 50%, l'impostazione della proprietà `_x` sposta un oggetto nel clip filmato di un numero di pixel pari alla metà rispetto al numero corrispondente per il filmato al 100%.

Vedere anche

`MovieClip._x`, `MovieClip._y`, `MovieClip._yscale`

MovieClip._y

Disponibilità

Flash Player 3.

Uso

myMovieClip._y

Descrizione

Proprietà; imposta la coordinata *y* del filmato rispetto alle coordinate locali del clip filmato principale. Se un clip filmato è nella linea temporale principale, allora il sistema di coordinate considera l'angolo superiore sinistro dello stage come il punto (0, 0). Se il clip filmato è all'interno di un altro clip filmato a cui sono state applicate trasformazioni, il clip si trova nel sistema di coordinate locali del clip che lo contiene. Ad esempio, se un clip filmato è ruotato di 90° in senso antiorario, il clip filmato secondario eredita un sistema di coordinate ruotato di 90° in senso antiorario. Le coordinate del clip filmato si riferiscono alla posizione del punto di registrazione.

Vedere anche

MovieClip._x, MovieClip._xscale, MovieClip._yscale

MovieClip._ymouse

Disponibilità

Flash Player 5.

Uso

myMovieClip._ymouse

Descrizione

Proprietà (sola lettura); restituisce la coordinata *y* della posizione del mouse.

Vedere anche

Mouse (oggetto), MovieClip._xmouse

MovieClip._yscale

Disponibilità

Flash Player 4.

Uso

myMovieClip._yscale

Descrizione

Proprietà; determina la scala verticale (*percentuale*) del clip filmato applicata dal punto di registrazione del clip filmato. Il punto di registrazione predefinito è (0,0).

Il ridimensionamento del sistema di coordinate locale influisce sull'impostazione delle proprietà *_x* e *_y* che sono definite in pixel interi. Se, ad esempio, il clip filmato principale è ridimensionato al 50%, l'impostazione della proprietà *_x* sposta un oggetto nel clip filmato di un numero di pixel pari alla metà rispetto al numero corrispondente per il filmato al 100%.

Vedere anche

MovieClip._x, MovieClip._xscale, MovieClip._y

NaN

Disponibilità

Flash Player 5.

Uso

NaN

Descrizione

Variabile; una variabile predefinita con il valore IEEE-754 per NaN (non un numero).

ne (non uguale; specifico per stringhe)

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto ed è consigliato l'uso dell'operatore `!=(diseguaglianza)`.

Uso

espressione1 `ne` *espressione2*

Parametri

espressione1, *espressione2* Numeri, stringhe o variabili.

Valori restituiti

Nessuno.

Descrizione

Operatore (confronto); confronta *espressione1* ed *espressione2*, quindi restituisce `true` se *espressione1* è diversa da *espressione2*; altrimenti restituisce `false`.

Vedere anche

`!=(diseguaglianza)`

new

Disponibilità

Flash Player 5.

Uso

`new` *funzione di costruzione*()

Parametri

funzione di costruzione Una funzione seguita da un qualsiasi parametro opzionale tra parentesi. La funzione corrisponde di solito al nome del tipo di oggetto (ad esempio Array, Math, Number, Object) da creare.

Valori restituiti

Nessuno.

Descrizione

Operatore; crea un nuovo oggetto, inizialmente anonimo, e richiama la funzione identificata mediante il parametro *funzione di costruzione*. L'operatore `new` passa alla funzione ogni parametro opzionale tra parentesi e il nuovo oggetto creato, a cui fa riferimento usando la parola chiave `this`. La funzione di costruzione può così usare `this` per impostare le variabili dell'oggetto.

La proprietà `prototype` della funzione di costruzione viene copiata nella proprietà `__proto__` del nuovo oggetto. Di conseguenza, il nuovo oggetto supporta tutti i metodi e le proprietà specificate nell'oggetto prototipo della funzione di costruzione.

Esempio

Nell'esempio seguente, viene creata la funzione `Book` e quindi usato l'operatore `new` per creare gli oggetti `book1` e `book2`.

```
function Book(name, price){
    this.name = name;
    this.price = price;
}

book1 = new Book("La congiura degli innocenti", 19.95);
book2 = new Book("Opera fluttuante", 10.95);
```

Esempio

Nell'esempio seguente, viene creata un'istanza dell'oggetto `Array` con 18 elementi:

```
golfCourse = new Array(18);
```

Vedere anche

`[]` (accesso matrice), `{}` (operatore di inizializzazione degli oggetti)

La sezione relativa al metodo di costruzione all'interno di ciascuna voce oggetto.

newline

Disponibilità

Flash Player 4.

Uso

```
newline
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Costante; inserisce un carattere di ritorno a capo (`\n`) aggiungendo una riga vuota nel codice ActionScript. La costante `newline` consente di creare spazio per le informazioni recuperate da una funzione o da un'azione nel codice.

nextFrame

Disponibilità

Flash 2.

Uso

nextFrame()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al successivo fotogramma e lo blocca in tale posizione.

Esempio

Quando l'utente preme il pulsante, l'indicatore di riproduzione si sposta al fotogramma successivo e si arresta.

```
on (release) {  
    nextFrame();  
}
```

nextScene

Disponibilità

Flash 2.

Uso

nextScene()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al fotogramma 1 della scena successiva e lo blocca in tale posizione.

Esempio

Quando l'utente rilascia il pulsante, l'indicatore di riproduzione si sposta al fotogramma 1 della scena successiva.

```
on(release) {  
    nextScene();  
}
```

Vedere anche

prevScene

not

Disponibilità

Flash Player 4. Questo operatore è diventato obsoleto ed è consigliato l'uso dell'operatore ! (NOT logico).

Uso

not espressione

Parametri

espressione Qualsiasi variabile o altra espressione convertibile in un valore booleano.

Descrizione

Operatore; esegue un'operazione di NOT logico in ambiente Flash Player 4.

Vedere anche

! (NOT logico)

null

Disponibilità

Flash Player 5.

Uso

null

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Parola chiave; valore speciale che può essere assegnato a variabili o restituito da una funzione se non è stato fornito alcun dato. Il valore *null* consente di rappresentare valori mancanti o valori senza un tipo di dati definito.

Esempio

In un contesto numerico, *null* viene valutato come 0. I test di uguaglianza possono essere effettuati con *null*. In questa istruzione, il nodo di un albero binario non possiede nodi secondari a sinistra; il campo per l'elemento a sinistra può quindi essere impostato su *null*.

```
if (tree.left == null) {  
    tree.left = new TreeNode();  
}
```

Number (funzione)

Disponibilità

Flash Player 4.

Uso

`Number(espressione)`

Parametri

espressione Espressione da convertire in un numero.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte, il parametro *espressione* in un numero e restituisce un valore nel modo seguente.

Se *espressione* è un numero, il valore restituito è *espressione*.

Se *espressione* è un valore booleano, il valore restituito è 1 se *espressione* risulta `true` e 0 se *espressione* risulta `false`.

Se *espressione* è una stringa, la funzione tenta di analizzare *espressione* come se fosse un numero decimale seguito da un valore esponente opzionale, ad esempio `1,57505e-3`.

Se *espressione* è `undefined`, il valore restituito è 0.

Questa funzione viene usata per convertire i file Flash 4 contenenti operatori obsoleti quando vengono importati in ambienti di creazione di Flash 5. Per ulteriori informazioni, vedere l'operatore `&`.

Vedere anche

`Number` (oggetto)

Number (oggetto)

L'oggetto `Number` è un semplice oggetto wrapper per il tipo di dati numerico, ossia consente di manipolare i valori numerici di base con i metodi e le proprietà associati all'oggetto `Number`. Questo oggetto è identico all'oggetto `Number` di JavaScript. In Flash MX, l'oggetto `Number` è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

È necessario usare una funzione di costruzione per richiamare i metodi dell'oggetto `Number`, ma non per richiamare le sue proprietà. Gli esempi seguenti descrivono la sintassi per richiamare i metodi e le proprietà dell'oggetto `Number`.

Nell'esempio seguente, viene richiamato il metodo `toString` dell'oggetto `Number` che restituisce la stringa `"1234"`.

```
myNumber = new Number(1234);  
myNumber.toString();
```

Nell'esempio seguente, viene richiamata la proprietà `MIN_VALUE` (definita anche come costante) dell'oggetto `Number`.

```
smallest = Number.MIN_VALUE
```

Riepilogo dei metodi validi per l'oggetto Number

Metodo	Descrizione
<code>Number.toString</code>	Restituisce una rappresentazione in formato stringa di un oggetto Number.
<code>Number.valueOf</code>	Restituisce il valore di base di un oggetto Number.

Riepilogo delle proprietà valide per l'oggetto Number

Proprietà	Descrizione
<code>Number.MAX_VALUE</code>	Costante equivalente al numero rappresentabile più elevato (doppia precisione IEEE-754). Questo numero è circa 1,7976931348623158e+308.
<code>Number.MIN_VALUE</code>	Costante equivalente al numero rappresentabile più basso (doppia precisione IEEE-754). Questo numero è circa 5e-324.
<code>Number.NaN</code>	Costante equivalente al valore per NaN (Not a Number).
<code>Number.NEGATIVE_INFINITY</code>	Costante equivalente al valore infinito negativo.
<code>Number.POSITIVE_INFINITY</code>	Costante equivalente al valore infinito positivo. Questo valore è lo stesso della variabile globale Infinity.

Funzione di costruzione per l'oggetto Number

Disponibilità

Flash Player 5.

Uso

```
myNumber = new Number(valore)
```

Parametri

valore Il valore numerico dell'oggetto Number che viene creato o un valore da convertire in un numero.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un nuovo oggetto Number. È necessario usare la funzione di costruzione Number quando vengono usati i metodi `toString` e `valueOf` dell'oggetto Number. Non usare una funzione di costruzione quando si usano le proprietà dell'oggetto Number. La funzione di costruzione `new Number` è principalmente usata come segnaposto. Un'istanza dell'oggetto Number non è analoga alla funzione Number che converte un parametro in un valore di base.

Esempio

Nel codice seguente vengono costruiti nuovi oggetti Number.

```
n1 = new Number(3.4);  
n2 = new Number(-10);
```

Vedere anche

Number (funzione)

Number.MAX_VALUE

Disponibilità

Flash Player 5.

Uso

Number.MAX_VALUE

Descrizione

Proprietà; il numero rappresentabile più elevato (doppia precisione IEEE-754). Questo numero è circa $1,79E+308$.

Number.MIN_VALUE

Disponibilità

Flash Player 5.

Uso

Number.MIN_VALUE

Descrizione

Proprietà; il numero rappresentabile più basso (doppia precisione IEEE-754). Questo numero è circa $5e-324$.

Number.NaN

Disponibilità

Flash Player 5.

Uso

Number.NaN

Descrizione

Proprietà; il valore IEEE-754 che rappresenta NaN (Not A Number, non un numero).

Number.NEGATIVE_INFINITY

Disponibilità

Flash Player 5.

Uso

Number.NEGATIVE_INFINITY

Descrizione

Proprietà; restituisce il valore IEEE-754 che rappresenta l'infinito negativo.

L'infinito negativo è un valore numerico speciale restituito quando un'operazione matematica o una funzione restituisce un valore negativo maggiore di quello rappresentabile.

Number.POSITIVE_INFINITY

Disponibilità

Flash Player 5.

Uso

`Number.POSITIVE_INFINITY`

Descrizione

Proprietà; restituisce il valore IEEE-754 che rappresenta l'infinito positivo. Questo valore è lo stesso della variabile globale `Infinity`.

L'infinito positivo è un valore numerico speciale restituito quando un'operazione matematica o una funzione restituisce un valore maggiore di quello rappresentabile.

Number.toString

Disponibilità

Flash Player 5.

Uso

`myNumber.toString(radice)`

Parametri

radice Specifica la base numerica (da 2 a 36) da usare per la conversione da numero a stringa. Se non si specifica il parametro *radice*, il valore predefinito è 10.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una rappresentazione in formato stringa dell'oggetto `Number` specificato (*Numero*).

Esempio

Nell'esempio seguente, viene usato il metodo `Number.toString` che specifica 2 per il parametro *radice* e restituisce una stringa contenente una rappresentazione binaria del numero 1000.

```
myNumber = new Number (1000);  
myNumber.toString(2);
```

Number.valueOf

Disponibilità

Flash Player 5.

Uso

`myNumber.valueOf()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il tipo di valore di base dell'oggetto Number specificato.

Object (oggetto)

L'oggetto generico Object si trova alla radice della gerarchia di classi ActionScript e usa alcune delle funzioni fornite dall'oggetto Object di JavaScript. In Flash MX, l'oggetto Object è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

L'oggetto generico Object è supportato in Flash Player 5.

Riepilogo dei metodi per l'oggetto Object

Metodo	Descrizione
<code>Object.addProperty</code>	Crea una proprietà getter/setter in un oggetto.
<code>Object.registerClass</code>	Assegna una classe ActionScript a un'istanza di clip filmato.
<code>Object.toString</code>	Converte l'oggetto specificato in una stringa e la restituisce.
<code>Object.unwatch</code>	Rimuove la registrazione creata da un metodo <code>Object.watch</code> .
<code>Object.valueOf</code>	Restituisce il valore di base di un oggetto Object.
<code>Object.watch</code>	Registra una funzione di callback da invocare quando una determinata proprietà di un oggetto ActionScript viene modificata.

Riepilogo delle proprietà per l'oggetto Object

Proprietà	Descrizione
<code>Object.__proto__</code>	Riferimento alla proprietà <code>prototype</code> della funzione di costruzione dell'oggetto.

Funzione di costruzione per l'oggetto Object

Disponibilità

Flash Player 5.

Uso

```
new Object([valore])
```

Parametri

valore Numero, valore booleano o stringa da convertire in un oggetto. Questo parametro è opzionale. Se *valore* non viene specificato, la funzione di costruzione crea un nuovo oggetto senza proprietà definite.

Descrizione

Funzione di costruzione; crea un nuovo oggetto Object.

Object.addProperty

Disponibilità

Flash Player 6.

Uso

```
myObject.addProperty( prop, getFunc, setFunc )
```

Parametri

prop Il nome della proprietà dell'oggetto da creare.

getFunc La funzione invocata per recuperare il valore della proprietà; questo parametro è un oggetto della funzione.

setFunc La funzione invocata per impostare il valore della proprietà; questo parametro è un oggetto della funzione. Se viene passato il valore `null` per questo parametro, la proprietà è in sola lettura.

Valori restituiti

Restituisce `true` se la proprietà è stata creata correttamente; altrimenti, restituisce `false`.

Descrizione

Metodo; crea una proprietà getter/setter. Quando Flash legge una proprietà getter/setter, invoca la funzione `getFunc` e il valore restituito diventa un valore di *prop*. Quando Flash scrive una proprietà getter/setter, invoca una funzione `setFunc` e le passa il nuovo valore come parametro. Se esiste già una proprietà con lo stesso nome assegnato alla nuova proprietà, quest'ultima la sovrascrive.

Una funzione `get` non prevede l'uso di parametri, può restituire qualsiasi tipo di valore e può essere di tipo diverso a seconda dell'invocazione. Il valore restituito viene considerato come valore corrente della proprietà.

Una funzione `set` è una funzione che assume un parametro come nuovo valore della proprietà. Ad esempio, se la proprietà `x` viene assegnata mediante l'istruzione `x = 1`, alla funzione `set` viene passato il parametro `1` di tipo numerico. Il valore restituito dalla funzione `setter` viene ignorato.

È possibile aggiungere proprietà getter/setter agli oggetti prototipo. In tal caso, tutte le istanze associate all'oggetto che ereditano l'oggetto prototipo ereditano anche la proprietà getter/setter. Ciò rende possibile aggiungere una proprietà getter/setter in un'unica posizione, ovvero nell'oggetto prototipo, e distribuirla a tutte le istanze di una classe, in modo simile a come vengono aggiunti i metodi agli oggetti prototipo. Se una funzione `get/set` viene invocata per una proprietà getter/setter in un oggetto prototipo ereditato, il riferimento passato a tale funzione sarà l'oggetto a cui era originariamente destinato il riferimento e non l'oggetto prototipo.

Se non viene invocata correttamente, `Object.addProperty` può determinare un errore. La tabella seguente descrive gli errori che possono verificarsi.

Condizione di errore	Conseguenze
<code>prop</code> non è un nome valido per una proprietà, ad esempio una stringa vuota.	Restituisce <code>false</code> e la proprietà non viene aggiunta.
<code>getFunc</code> non è un oggetto valido per la funzione.	Restituisce <code>false</code> e la proprietà non viene aggiunta.
<code>setFunc</code> non è un oggetto valido per la funzione.	Restituisce <code>false</code> e la proprietà non viene aggiunta.

Esempio

Uso 1: le proprietà incorporate in `TextField.scroll` e `TextField.maxscroll` sono proprietà getter/setter. L'oggetto `TextField` ha metodi interni `getScroll`, `setScroll` e `getMaxScroll`. La funzione di costruzione `TextField` crea le proprietà getter/setter e le indirizza ai metodi get/set interni, come nell'esempio seguente.

```
this.addProperty("scroll", this.getScroll, this.setScroll);  
this.addProperty("maxscroll", this.getMaxScroll, null);
```

Quando uno script recupera il valore di `myTextField.scroll`, l'interprete di `ActionScript` invoca automaticamente `myTextField.getScroll`. Quando uno script modifica il valore di `myTextField.scroll`, l'interprete invoca `myTextField.setScroll`. La proprietà `maxscroll` non specifica una funzione set, per cui i tentativi di modificare `maxscroll` vengono ignorati.

Uso 2: analogo all'esempio precedente con `TextField.scroll` e `TextField.maxscroll`, ma in questo caso le proprietà `scroll` e `maxscroll` vengono aggiunte a ogni istanza dell'oggetto `TextField`. Ciò significa che, per ciascuna istanza dell'oggetto, saranno necessarie due posizioni per le proprietà. Se molte proprietà come `scroll` e `maxscroll` sono presenti all'interno di una classe, esse possono occupare molto spazio nella memoria. In alternativa, è possibile aggiungere le proprietà `scroll` e `maxscroll` a `TextField.prototype`:

```
TextField.prototype.addProperty("scroll", this.getScroll, this.setScroll);  
TextField.prototype.addProperty("maxscroll", this.getMaxScroll, null);
```

In questo modo, le proprietà `scroll` e `maxscroll` saranno presenti in un'unica posizione: l'oggetto `TextField.prototype`. L'effetto è comunque analogo a quello prodotto dal codice riportato nell'esempio precedente in cui `scroll` e `maxscroll` sono state aggiunte direttamente a ogni singola istanza. Se si accede a `scroll` o a `maxscroll` in un'istanza `TextField` risalendo nella catena di prototipi la proprietà getter/setter viene individuata in `TextField.prototype`.

Object.__proto__

Disponibilità

Flash Player 5.

Uso

```
myObject.__proto__
```

Parametri

Nessuno.

Descrizione

Proprietà; si riferisce alla proprietà `prototype` della funzione di costruzione che ha creato *myObject*. La proprietà `__proto__` è automaticamente assegnata a tutti gli oggetti quando vengono creati. L'interprete di `ActionScript` usa la proprietà `__proto__` per accedere alla proprietà `prototype` della funzione di costruzione dell'oggetto per individuare le proprietà e i metodi che l'oggetto eredita dalla sua classe.

Object.registerClass

Disponibilità

Flash Player 6.

Uso

```
Object.registerClass(IDSimbolo, Classe)
```

Parametri

IDSimbolo L'identificatore di concatenamento del simbolo di clip filmato oppure l'identificatore di stringa per la classe ActionScript.

Classe Riferimento alla funzione di costruzione della classe ActionScript, o null per annullare la registrazione del simbolo.

Valori restituiti

Se la classe viene registrata correttamente, il valore restituito è true; in caso contrario, viene restituito false.

Descrizione

Metodo; associa un simbolo di clip filmato a una classe oggetto di ActionScript. Se un simbolo risulta inesistente, Flash crea un'associazione tra un identificatore di stringa e una classe oggetto.

Quando un'istanza del simbolo di clip filmato specificato viene posizionata sulla linea temporale, viene registrata nella classe specificata dal parametro *Classe* e non nella classe MovieClip.

Quando un'istanza del simbolo di clip filmato specificato viene creata usando i metodi `attachMovie` o `duplicateMovieClip`, essa viene registrata nella classe specificata dal parametro *Classe* e non nella classe MovieClip.

Se il parametro *Classe* risulta null, `Object.registerClass` rimuove ogni definizione di classe ActionScript o identificatore di classe associati al simbolo di clip filmato specificato. Per quanto concerne i simboli di clip filmato, ogni istanza del clip filmato già esistente rimane inalterata, ma nuove istanze del simbolo vengono associate alla classe predefinita MovieClip.

Se un simbolo è già registrato in una classe, il metodo `Object.registerClass` lo sostituisce con la nuova registrazione.

Quando un'istanza di clip filmato viene posizionata sulla linea temporale o creata usando `attachMovie` o `duplicateMovieClip`, ActionScript invoca la funzione di costruzione per la classe appropriata con la parola chiave `this` associata all'oggetto. La funzione di costruzione viene invocata senza parametri.

Se si usa il metodo `Object.registerClass` per registrare un clip filmato con una classe ActionScript diversa da MovieClip, il simbolo di clip filmato non eredita i metodi, le proprietà e gli eventi della classe incorporata MovieClip, a meno che questa non venga inclusa nella catena di prototipi della nuova classe. Il codice seguente crea una nuova classe ActionScript denominata `theClass` che eredita le proprietà della classe MovieClip:

```
theClass.prototype = new MovieClip();
```

Esempio

Nell'esempio seguente, viene creato un componente per un widget d'interfaccia utente standard costituito da una casella di controllo.

Innanzitutto, occorre creare nella libreria un simbolo di clip filmato denominato Casella di controllo. Quindi, creare in ActionScript la classe CheckBox che rappresenterà la casella di controllo.

```
// Definisce la funzione di costruzione per definire
// la classe CheckBox e la classe stessa

function CheckBox() {
    ...
}

// Imposta la catena di prototipi CheckBox
// affinché erediti le proprietà da MovieClip

CheckBox.prototype = new MovieClip();

// Definisce i metodi per la classe CheckBox

CheckBox.prototype.setLabel = function (newLabel) {
    this.label = newLabel;
    ...
};
CheckBox.prototype.setValue = function (newValue) {
    this.value = value;
    ...
};
CheckBox.prototype.getValue = function () {
    return this.value;
};
CheckBox.prototype.getLabel = function () {
    return this.label;
};
```

A questo punto, occorre associare la classe CheckBox al simbolo di clip filmato Casella di controllo. Innanzitutto, è necessario poter identificare il simbolo di clip filmato Casella di controllo. A tal fine, specificare un identificatore nella finestra di dialogo Concatenamento nella libreria e selezionare Esporta per ActionScript.

Quindi, creare le istruzioni di ActionScript per associare la classe CheckBox al simbolo Casella di controllo.

```
Object.registerClass("Casella di controllo" /*IDsimbolo*/, CheckBox /*Classe*/
);
```

Uso 1 (posizionamento sulla linea temporale): è ora possibile posizionare le istanze di CheckBox sullo stage nello strumento di creazione e, in fase di esecuzione, le istanze riceveranno automaticamente la classe CheckBox di ActionScript. Se si posizionano due istanze, myCheckBox1 e myCheckBox2, è possibile controllarle invocando dei metodi, come nell'esempio seguente.

```
myCheckBox1.setValue(true);
myCheckBox2.setValue(false);
myCheckBox2.setLabel("nuova etichetta per #2");
```

Usò 2 (istanze dinamiche): è possibile usare il metodo `attachMovie` per creare una nuova istanza della casella di controllo sullo stage durante la riproduzione del filmato. Dato che il simbolo Casella di controllo è registrato nella classe `CheckBox` di `ActionScript`, la nuova istanza dinamica riceverà automaticamente tale classe.

```
// createCheckBox è una funzione di supporto che
// crea dinamicamente Caselle di controllo
function createCheckBox(name, depth) {
    attachMovie("Casella di controllo", name, depth);
}
createCheckBox("myCheckBox3", 100);
myCheckBox3.setValue(false);
myCheckBox2.setLabel("nuova etichetta per #3");
```

Vedere anche

`MovieClip.attachMovie`, `MovieClip.duplicateMovieClip`

Object.toString

Disponibilità

Flash Player 5.

Uso

```
myObject.toString()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; converte l'oggetto specificato in una stringa e la restituisce.

Object.unwatch

Disponibilità

Flash Player 6.

Uso

```
myObject.unwatch (prop)
```

Parametri

prop Il nome della proprietà dell'oggetto che non deve più essere controllata, sotto forma di stringa.

Valori restituiti

Un valore booleano.

Descrizione

Metodo; rimuove un punto di controllo creato dal metodo `Object.watch`. Questo metodo restituisce `true` se il punto di controllo viene rimosso correttamente; in caso contrario, restituisce `false`.

Object.valueOf

Disponibilità

Flash Player 5.

Uso

```
myOggetto.valueOf()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il valore di base dell'oggetto specificato. Se l'oggetto non possiede alcun valore di base, viene restituito l'oggetto stesso.

Object.watch

Disponibilità

Flash Player 6.

Uso

```
myObject.watch( prop, callback [, datiUtente] )
```

Parametri

prop Stringa che indica il nome della proprietà dell'oggetto da controllare.

callback La funzione da invocare quando la proprietà controllata viene modificata. Questo parametro è l'oggetto di una funzione e non il nome di una funzione sotto forma di stringa. Il modulo formato dal parametro *callback* è `callback(prop, oldval, newval, datiUtente)`.

datiUtente Elemento facoltativo dei dati ActionScript passato al metodo *callback*. Se il parametro *datiUtente* viene ommesso, il valore `undefined` viene passato al metodo *callback*. Questo parametro è opzionale.

Valori restituiti

Se il punto di controllo è stato creato correttamente, restituisce `true`; in caso contrario, restituisce `false`.

Descrizione

Metodo; registra una funzione di callback da invocare quando una determinata proprietà di un oggetto ActionScript viene modificata. In tal caso, la funzione di callback viene invocata con `myObject` come oggetto che la contiene.

Un punto di controllo può filtrare (o annullare) l'assegnazione di un valore modificando il valore `newval` o `oldval` restituito. Se si elimina una proprietà per la quale era stato creato un punto di controllo, quest'ultimo non scompare. Se in seguito si ricrea tale proprietà, il punto di controllo sarà ancora valido. Per rimuovere un punto di controllo, usare il metodo `Object.unwatch`.

Per ciascuna proprietà è possibile registrare un solo punto di controllo. Le successive chiamate di `Object.watch` sulla stessa proprietà sostituiscono il punto di controllo originale.

Il metodo `Object.watch` è simile alla funzione `Object.watch` in Netscape JavaScript 1.2 e versioni successive. La differenza principale consiste nel parametro `datiUtente`, aggiunto al metodo `Object.watch` in Flash e non supportato da Netscape Navigator. Il parametro `datiUtente` può essere passato alla funzione di callback e usato in essa.

Il metodo `Object.watch` non può controllare le proprietà `getter/setter`. Queste ultime, infatti, operano per "lazy evaluation" (valutazione pigra), ossia il valore della proprietà non viene determinato finché questa non viene richiesta. Questo tipo di valutazione è spesso efficace poiché la proprietà non viene costantemente aggiornata, ma viene invece valutata solo quando necessario. Tuttavia, `Object.watch` deve poter valutare la proprietà per poter attivare dei punti di controllo su di essa. Per funzionare con una proprietà `getter/setter`, `Object.watch` necessita di una valutazione costante della stessa, con conseguente inefficienza.

In genere, le proprietà predefinite di `ActionScript`, quali `_x`, `_y`, `_width` e `_height`, sono proprietà `getter/setter` e quindi non possono essere controllate con `Object.watch`.

Esempio

L'esempio seguente illustra un componente Casella di controllo con metodi che ne determinano l'etichetta o il valore di ciascuna istanza della casella di controllo.

```
myCheckBox1.setValue(true);
myCheckBox1.setLabel("nuova etichetta");
...
```

È opportuno considerare il valore e l'etichetta di una casella di controllo come proprietà. Usando `Object.watch`, è possibile rendere l'accesso al valore e all'etichetta simile all'accesso alla proprietà piuttosto che all'invocazione di un metodo, come nell'esempio seguente:

```
// Definisce la funzione di costruzione per la classe CheckBox e la classe
// stessa function CheckBox() {
...
  this.watch('value', function (id, oldval, newval) {
    ...
  })
  this.watch('label', function (id, oldval, newval) {
    ...
  })
}
```

Quando le proprietà `value` o `label` vengono modificate, la funzione specificata dal componente viene invocata per eseguire ogni azione necessaria all'aggiornamento dell'aspetto e dello stato del componente con i nuovi parametri. Pertanto, la seguente istruzione di assegnazione usa un gestore `Object.watch` per notificare al componente che la variabile è stata modificata e per aggiornarne la rappresentazione grafica.

```
myCheckBox1.value = false;
```

Questa sintassi è più concisa della precedente.

```
myCheckBox1.setValue(false);
```

Vedere anche

`Object.addProperty`, `Object.unwatch`

onClipEvent

Disponibilità

Flash Player 5.

Uso

```
onClipEvent(eventoFilmato){  
    istruzione/i;  
}
```

Parametri

Un *eventoFilmato* è un trigger denominato *evento*. Quando esso si verifica, vengono eseguite le istruzioni che lo seguono tra parentesi graffe. Per il parametro *eventoFilmato* è possibile specificare uno dei valori seguenti:

- **caricamento** L'azione viene attivata appena il clip filmato diventa un'istanza e appare nella linea temporale.
- **scaricamento** L'azione viene attivata nel primo fotogramma dopo la rimozione del clip filmato dalla linea temporale. Le azioni associate all'evento clip filmato `Unload` vengono eseguite prima di qualsiasi azione associata al fotogramma interessato.
- **enterFrame** L'azione viene attivata costantemente con frequenza pari a quella dei fotogrammi del filmato. Le azioni associate all'evento di clip filmato `enterFrame` vengono eseguite prima di qualsiasi azione associata ai fotogrammi interessati.
- **mouseMove** L'azione viene avviata a ogni spostamento del mouse. L'uso delle proprietà `_xmouse` e `_ymouse` consente di determinare la posizione corrente del mouse.
- **mouseDown** L'azione viene avviata quando il pulsante sinistro del mouse viene premuto.
- **mouseUp** L'azione viene avviata quando il pulsante sinistro del mouse viene rilasciato.
- **keyDown** L'azione viene avviata quando viene premuto un tasto. L'uso del metodo `Key.getCode` consente di recuperare informazioni sull'ultimo tasto premuto.
- **keyUp** L'azione viene avviata quando viene rilasciato un tasto. L'uso del metodo `Key.getCode` consente di recuperare informazioni sull'ultimo tasto premuto.
- **dati** L'azione viene avviata al ricevimento di dati in un'azione `loadVariables` o `loadMovie`. Se specificato con un'azione `loadVariables`, l'evento `data` si verifica una sola volta, al caricamento dell'ultima variabile. Se specificato con un'azione `loadMovie`, l'evento `data` si verifica ripetutamente, al ricevimento di ogni sezione di dati.

istruzione/i Le istruzioni da eseguire quando si verifica *eventoMouse*.

Descrizione

Gestore di evento; innesca le azioni definite per un'istanza specifica di un clip filmato.

Esempio

La seguente istruzione include lo script di un file esterno quando il filmato viene esportato; le azioni nello script incluso vengono eseguite quando viene caricato il clip filmato a cui sono associate.

```
onClipEvent(load) {  
    #include "myScript.as"  
}
```

Nell'esempio seguente, viene usato `onClipEvent` con l'evento filmato `keyDown`. L'evento filmato `keyDown` viene solitamente usato insieme a uno o più metodi e proprietà dell'oggetto `Key`. Lo script sotto riportato usa il metodo `Key.getCode` per determinare il tasto premuto dall'utente; se il tasto premuto corrisponde alla proprietà `Key.RIGHT`, il filmato viene indirizzato al fotogramma successivo; se invece corrisponde alla proprietà `Key.LEFT`, il filmato viene indirizzato al fotogramma precedente.

```
onClipEvent(keyDown) {  
    if (Key.getCode() == Key.RIGHT) {  
        _parent.nextFrame();  
    } else if (Key.getCode() == Key.LEFT){  
        _parent.prevFrame();  
    }  
}
```

L'esempio seguente usa `onClipEvent` con l'evento filmato `mouseMove`. Le proprietà `_xmouse` e `_ymouse` segnalano la posizione del mouse a ogni suo spostamento.

```
onClipEvent(mouseMove) {  
    stageX=_root.xmouse;  
    stageY=_root.ymouse;  
}
```

Vedere anche

`Key` (oggetto), `MovieClip._xmouse`, `MovieClip._ymouse`, `on`

on

Disponibilità

Flash 2. Non tutti gli eventi sono supportati in Flash 2.

Uso

```
on(eventoMouse) {  
    istruzione/i;  
}
```

Parametri

istruzione/i Le istruzioni da eseguire quando si verifica *eventoMouse*.

Un *eventoMouse* è un trigger denominato "evento". Quando esso si verifica, vengono eseguite le istruzioni che lo seguono tra parentesi graffe. Per il parametro *eventoMouse* è possibile specificare uno dei valori seguenti:

- `pressione` Il pulsante del mouse viene premuto mentre il puntatore si trova sopra il pulsante.
- `rilascio` Il pulsante del mouse viene rilasciato mentre il puntatore si trova sopra il pulsante.
- `releaseOutside` Il pulsante del mouse viene rilasciato mentre il puntatore si trova al di fuori dell'area del pulsante; dopo la pressione del pulsante se il puntatore è all'interno di essa.

- `rollOut` Il puntatore del mouse viene spostato fuori dall'area del pulsante.
- `rollOver` Il puntatore del mouse scorre sopra il pulsante.
- `dragOut` Mentre il puntatore si trova sopra il pulsante, il pulsante del mouse viene premuto, quindi il puntatore viene trascinato fuori dall'area del pulsante.
- `dragOver` Mentre il puntatore si trova sopra il pulsante, il pulsante del mouse viene premuto, quindi il puntatore viene spostato fuori dall'area del pulsante e infine riportato di nuovo sopra quest'ultima.
- `keyPress ("tasto")` Il *tasto* specificato viene premuto. La parte del parametro relativa al *tasto* viene specificata tramite uno dei codici tasto elencati nell'Appendice C "Tasti della tastiera e valori dei codici tasto" della *Guida all'uso di Flash* o tramite una delle costanti tasto elencate in Riepilogo delle proprietà valide per l'oggetto `Key`.

Descrizione

Gestore di evento; specifica l'evento mouse o la pressione di tasto che innesca un'azione.

Esempio

Nello script seguente, l'azione `startDrag` viene eseguita quando il mouse viene premuto e lo script condizionale viene eseguito quando viene rilasciato il pulsante del mouse e di conseguenza l'oggetto.

```
on(press) {
    startDrag("coniglio");
}
on(release) {
    trace(_root.rabbit._y);
    trace(_root.rabbit._x);
    stopDrag();
}
```

Vedere anche

`onClipEvent`

or

Disponibilità

Flash 4. Questo operatore è diventato obsoleto ed è consigliato l'uso dell'operatore `||` (OR logico).

Uso

condizione1 or *condizione2*

Parametri

condizione1, *condizione2* Espressione che restituisce `true` o `false`.

Valori restituiti

Nessuno.

Descrizione

Operatore; valuta *condizione1* e *condizione2*; se una delle due espressioni risulta `true`, l'intera espressione viene valutata come `true`.

Vedere anche

|| (OR logico), | (OR bit a bit)

ord

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta ed è consigliato l'uso dei metodi e delle proprietà di String (oggetto).

Uso

```
ord(carattere)
```

Parametri

carattere Il carattere da convertire in un numero di codice ASCII.

Valori restituiti

Nessuno.

Descrizione

Funzione stringa; converte i caratteri in numeri di codice ASCII.

Vedere anche

String (oggetto)

_parent

Disponibilità

Flash Player 4.

Uso

```
_parent.property
```

```
_parent._parent.property
```

Descrizione

Proprietà; specifica o restituisce un riferimento al clip filmato contenente l'oggetto o il clip filmato corrente. L'oggetto corrente è quello contenente il codice ActionScript che fa riferimento a `_parent`. Usare `_parent` per specificare il percorso relativo a clip filmato o oggetti gerarchicamente superiori all'oggetto o al clip filmato attuale.

Esempio

Nell'esempio seguente, il clip filmato `desk` è l'elemento secondario del clip filmato `classroom`. Quando lo script seguente viene eseguito all'interno del clip filmato `desk`, l'indicatore di riproduzione passa al fotogramma 10 nella linea temporale del clip filmato `classroom`.

```
_parent.gotoAndStop(10);
```

Vedere anche

`_root`, `targetPath`

parseFloat

Disponibilità

Flash Player 5.

Uso

`parseFloat(stringa)`

Parametri

stringa La stringa da analizzare sintatticamente e convertire in un numero in virgola mobile.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte una stringa in un numero in virgola mobile. La funzione legge o esegue l'analisi sintattica e restituisce i numeri nella stringa finché non raggiunge un carattere che non è parte del numero iniziale. Se la stringa non inizia con un numero che può essere analizzato sintatticamente, `parseFloat` restituisce NaN. Gli spazi vuoti prima di numeri interi validi vengono ignorati, così come i caratteri finali non numerici.

Esempio

Gli esempi seguenti descrivono l'uso di `parseFloat` per valutare diversi tipi di numeri:

`parseFloat("-2")` restituisce -2

`parseFloat("2,5")` restituisce 2,5

`parseFloat("3,5e6")` restituisce 3,5e6 o 3500000

`parseFloat("foobar")` restituisce NaN

`parseFloat(" 5,1")` restituisce 5,1

`parseFloat("3.75math")` restituisce 3,75

`parseFloat("0garbage")` restituisce 0

parseInt

Disponibilità

Flash Player 5.

Uso

`parseInt(espressione, [radice])`

Parametri

espressione Stringa da convertire in un numero intero.

radice Numero intero che rappresenta la radice (base) del numero da analizzare sintatticamente. I valori consentiti sono compresi tra 2 e 36. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Funzione; converte una stringa in un numero intero. Se non è possibile convertire la stringa specificata nei parametri in un numero, la funzione restituisce NaN. Se la stringa inizia con 0 o se si specifica una radice di 8 il numero viene interpretato come numero ottale. Le stringhe che iniziano con 0x vengono interpretate come numeri esadecimali. Gli spazi vuoti prima di numeri interi validi vengono ignorati, così come i caratteri finali non numerici.

Esempio

Gli esempi seguenti descrivono l'uso di `parseInt` per valutare diversi tipi di numeri.

```
parseInt("3,5")  
// Restituisce 3,5
```

```
parseInt("bar")  
// Restituisce NaN
```

```
parseInt("4foo")  
// Restituisce 4
```

Di seguito sono riportati alcuni esempi di conversione esadecimale:

```
parseInt("0x3F8")  
// Restituisce 1016
```

```
parseInt("3E8", 16)  
// Restituisce 1000
```

Di seguito sono riportati alcuni esempi di conversione binaria:

```
parseInt("1010", 2)  
// Restituisce 10 (la rappresentazione decimale del numero binario 1010)
```

Di seguito viene riportato un esempio di analisi di un numero ottale (in questo caso il numero ottale è identificato tramite la radice, 8):

```
parseInt("777", 8)  
// Restituisce 511 (la rappresentazione decimale del numero ottale 777)
```

play

Disponibilità

Flash 2.

Uso

```
play()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; sposta l'indicatore di riproduzione in avanti nella linea temporale.

Esempio

Il codice seguente usa un'istruzione `if` per verificare il valore di un nome immesso dall'utente. Se l'utente immette `Steve`, viene richiamata l'azione `play` e l'indicatore di riproduzione si sposta in avanti lungo la linea temporale. Se l'utente immette un valore diverso da `Steve`, il filmato non viene riprodotto e viene visualizzato un campo di testo con il nome di variabile `alert`.

```
stop();
if (name == "Steve") {
    play();
} else
    alert="Tu non sei Steve!";
}
```

prevFrame

Disponibilità

Flash 2.

Uso

`prevFrame()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al fotogramma precedente e lo blocca in tale posizione. Se il fotogramma corrente è 1, l'indicatore di riproduzione non si sposta.

Esempio

Quando l'utente sceglie un pulsante al quale è stato associato l'evento seguente, l'indicatore di riproduzione viene inviato al fotogramma precedente.

```
on(release) {
    prevFrame();
}
```

Vedere anche

`MovieClip.prevFrame`

prevScene

Disponibilità

Flash 2.

Uso

`prevScene()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; invia l'indicatore di riproduzione al fotogramma3 1 della scena precedente e lo blocca in tale posizione.

Vedere anche

`nextScene`

print

Disponibilità

Flash Player 4.20.

Uso

```
print (livello)
print (livello, "Riquadro di limitazione")
print ("target")
print ("target", "Riquadro di limitazione")
printAsBitmap (livello)
printAsBitmap (livello, "Riquadro di limitazione")
printAsBitmap ("target")
printAsBitmap ("target", "Riquadro di limitazione")
```

Parametri

stampa Nel pannello Azioni in Modalità normale, scegliere Come vettori per stampare fotogrammi che non contengono immagini bitmap, trasparenza (alfa) o effetti di colore; scegliere Come bitmap per stampare fotogrammi che contengono immagini bitmap, trasparenze o effetti di colore. Se si sceglie il parametro Stampa Come bitmap, la sintassi `AsBitmap` viene aggiunta all'azione `print` nel pannello Azioni.

livello Il livello di Flash Player da stampare. Se si sceglie un livello nel pannello Azioni in Modalità normale, l'azione `print` diventa `printNum` o `printAsBitmapNum`; in Modalità esperto, è necessario specificare sia `printNum` che `printAsBitmapNum`. Per impostazione predefinita, tutti i fotogrammi presenti nella stampa del livello. Se si desidera stampare dei fotogrammi specifici presenti nel livello, assegnare un'etichetta `#p` a tali fotogrammi.

target Il nome dell'istanza di clip filmato da stampare. Per impostazione predefinita, tutti i fotogrammi dell'istanza `target` vengono stampati. Se si desidera stampare dei fotogrammi specifici del clip filmato, assegnare un'etichetta `#p` a tali fotogrammi.

Riquadro di limitazione Modificatore che imposta l'area di stampa del filmato. Questo parametro è opzionale. È possibile scegliere tra le opzioni seguenti:

- *bfilmato* Indica il riquadro di limitazione di un fotogramma specifico del filmato come area di stampa per tutti i fotogrammi stampabili nel filmato. Assegna un'etichetta `#b` al fotogramma di cui si vuole usare il riquadro di limitazione come area di stampa.
- *bmax* Indica come area di stampa un riquadro di limitazione composito formato da tutti i fotogrammi stampabili. Specificare il parametro `bmax` quando i fotogrammi stampabili nel filmato sono di dimensioni diverse.
- *bfotogramma* Indica che il riquadro di limitazione di un determinato fotogramma stampabile deve essere usato come area di stampa per quel fotogramma. Ciò modifica l'area di stampa per ogni fotogramma e agisce sulla scala degli oggetti per adattarli all'area di stampa. L'uso di `bfotogramma` consente, nel caso di oggetti di diverse dimensioni in ogni fotogramma, di riempire la pagina stampata con l'oggetto.

Valori restituiti

Nessuno.

Descrizione

Azione; stampa il clip filmato *target* in base ai limiti dell'area di stampa specificati nel parametro *bfilmato*, *bmax* o *bfotogramma*. Se si desidera stampare dei fotogrammi specifici del filmato *target*, assegnare un'etichetta *#p* a tali fotogrammi. Sebbene l'azione *print* offra una qualità di stampa più elevata rispetto all'azione *printAsBitmap*, essa non può essere impiegata per stampare filmati che usano trasparenze alfa o effetti colore speciali.

Se non si specifica un parametro che delimiti l'area di stampa, quest'ultima viene determinata, per impostazione predefinita, dalla dimensione dello stage del filmato caricato. Il filmato non eredita la dimensione dello stage del filmato principale. È possibile controllare l'area di stampa specificando i parametri *bfilmato*, *bmax* o *bfotogramma*.

Tutti gli elementi stampabili di un filmato devono essere completamente caricati prima dell'inizio della stampa.

La funzione di stampa di Flash Player supporta stampanti PostScript e non PostScript. Le stampanti non PostScript convertono i vettori in bitmap.

Esempio

L'esempio seguente stampa tutti i fotogrammi stampabili nel clip filmato *myMovie* con l'area di stampa definita dal riquadro di delimitazione del fotogramma a cui è associata l'etichetta di fotogramma *#b*.

```
print("myMovie", "bmovie");
```

Nell'esempio seguente, vengono stampati tutti i fotogrammi stampabili in *myMovie* con l'area di stampa definita dal riquadro di delimitazione di ogni fotogramma.

```
print("myMovie", "bmovie");
```

Vedere anche

`printNum`, `printAsBitmap`, `printAsBitmapNum`

printAsBitmap

Disponibilità

Flash Player 4.20.

Uso

```
printAsBitmap (target, "Riquadro di limitazione")
```

Parametri

target Il nome dell'istanza di clip filmato da stampare. Per impostazione predefinita vengono stampati tutti i fotogrammi del filmato. Se si desidera stampare dei fotogrammi specifici del filmato, assegnare un'etichetta *#p* a tali fotogrammi.

Riquadro di limitazione Modificatore che imposta l'area di stampa del filmato. È possibile scegliere tra i parametri seguenti:

- *bfilmato* Indica il riquadro di limitazione di un fotogramma specifico del filmato come area di stampa per tutti i fotogrammi stampabili nel filmato. Assegna un'etichetta *#b* al fotogramma di cui si vuole usare il riquadro di limitazione come area di stampa.

- `bmax` Indica come area di stampa un riquadro di limitazione composito formato da tutti i fotogrammi stampabili. Specificare il parametro `bmax` quando i fotogrammi stampabili nel filmato sono di dimensioni diverse.
- `bfotogramma` Indica che il riquadro di limitazione di un determinato fotogramma stampabile deve essere usato come area di stampa per quel fotogramma. Ciò modifica l'area di stampa per ogni fotogramma e agisce sulla scala degli oggetti per adattarli all'area di stampa. L'uso di `bfotogramma` consente, nel caso di oggetti di diverse dimensioni in ogni fotogramma, di riempire la pagina stampata con l'oggetto.

Valori restituiti

Nessuno.

Descrizione

Azione; stampa il clip filmato *target* come bitmap. L'uso dell'azione `printAsBitmap` consente di stampare i filmati contenenti fotogrammi con oggetti che usano trasparenza o effetti colore. L'azione `printAsBitmap` consente la stampa con la più alta risoluzione disponibile della stampante in modo da mantenere la massima definizione e qualità.

Per ottenere risultati ottimali se il filmato non contiene trasparenze alfa o effetti colore, si consiglia di usare l'azione `print`.

Per impostazione predefinita, l'area di stampa è determinata dalla dimensione dello stage del filmato caricato. Il filmato non eredita la dimensione dello stage del filmato principale. È possibile controllare l'area di stampa specificando i parametri `bfilmato`, `bmax` o `bfotogramma`.

Tutti gli elementi stampabili di un filmato devono essere completamente caricati prima dell'inizio della stampa.

La funzione di stampa di Flash Player supporta stampanti PostScript e non PostScript. Le stampanti non PostScript convertono i vettori in bitmap.

Vedere anche

`print`, `printAsBitmapNum`, `printNum`

printAsBitmapNum

Disponibilità

Flash Player 5.

Uso

```
printAsBitmapNum(livello)
printAsBitmapNum(livello, "Riquadro di limitazione")
```

Parametri

livello Il livello di Flash Player da stampare. Per impostazione predefinita, tutti i fotogrammi presenti nella stampa del livello. Se si desidera stampare dei fotogrammi specifici presenti nel livello, assegnare un'etichetta `#p` a tali fotogrammi.

Riquadro di limitazione Modificatore che imposta l'area di stampa del filmato. Questo parametro è opzionale. È possibile scegliere tra i parametri seguenti:

- `bfilmato` Indica il riquadro di limitazione di un fotogramma specifico del filmato come area di stampa per tutti i fotogrammi stampabili nel filmato. Assegna un'etichetta `#b` al fotogramma di cui si vuole usare il riquadro di limitazione come area di stampa.

- `bmax` Indica come area di stampa un riquadro di limitazione composito formato da tutti i fotogrammi stampabili. Specificare il parametro `bmax` quando i fotogrammi stampabili nel filmato sono di dimensioni diverse.
- `bfotogramma` Indica che il riquadro di limitazione di un determinato fotogramma stampabile deve essere usato come area di stampa per quel fotogramma. Ciò modifica l'area di stampa per ogni fotogramma e agisce sulla scala degli oggetti per adattarli all'area di stampa. L'uso di `bfotogramma` consente, nel caso di oggetti di diverse dimensioni in ogni fotogramma, di riempire la pagina stampata con l'oggetto.

Valori restituiti

Nessuno.

Descrizione

Azione; stampa un livello in Flash Player come bitmap. L'uso dell'azione `printAsBitmapNum` consente di stampare i filmati contenenti fotogrammi con oggetti che usano trasparenze o effetti colore. L'azione `printAsBitmapNum` consente di stampare alla risoluzione massima disponibile della stampante per conservare la migliore definizione e qualità. Per calcolare la dimensione del file stampabile di un fotogramma da stampare come bitmap, moltiplicare la larghezza in pixel per l'altezza in pixel per la risoluzione della stampante.

Per ottenere risultati ottimali se il filmato non contiene trasparenze alfa o effetti colore, si consiglia di usare l'azione `printNum`.

Per impostazione predefinita, l'area di stampa è determinata dalla dimensione dello stage del filmato caricato. Il filmato non eredita la dimensione dello stage del filmato principale. È possibile controllare l'area di stampa specificando i parametri `bfilmato`, `bmax` o `bfotogramma`.

Tutti gli elementi stampabili di un filmato devono essere completamente caricati prima dell'inizio della stampa.

La funzione di stampa di Flash Player supporta stampanti PostScript e non PostScript. Le stampanti non PostScript convertono i vettori in bitmap.

Vedere anche

`print`, `printAsBitmap`, `printNum`

printNum

Disponibilità

Flash Player 5.

Uso

```
printNum (livello, "Riquadro di limitazione")
```

Parametri

livello Il livello di Flash Player da stampare. Per impostazione predefinita, tutti i fotogrammi presenti nella stampa del livello. Se si desidera stampare dei fotogrammi specifici presenti nel livello, assegnare un'etichetta `#p` a tali fotogrammi.

Riquadro di limitazione Modificatore che imposta l'area di stampa del filmato. È possibile scegliere tra i parametri seguenti:

- `bfilmato` Indica il riquadro di limitazione di un fotogramma specifico del filmato come area di stampa per tutti i fotogrammi stampabili nel filmato. Assegna un'etichetta `#b` al fotogramma di cui si vuole usare il riquadro di limitazione come area di stampa.
- `bmax` Indica come area di stampa un riquadro di limitazione composito formato da tutti i fotogrammi stampabili. Specificare il parametro `bmax` quando i fotogrammi stampabili nel filmato sono di dimensioni diverse.
- `bfotogramma` Indica che il riquadro di limitazione di un determinato fotogramma stampabile deve essere usato come area di stampa per quel fotogramma. Ciò modifica l'area di stampa per ogni fotogramma e agisce sulla scala degli oggetti per adattarli all'area di stampa. L'uso di `bfotogramma` consente, nel caso di oggetti di diverse dimensioni in ogni fotogramma, di riempire la pagina stampata con l'oggetto.

Valori restituiti

Nessuno.

Descrizione

Azione; stampa il livello in Flash Player rispettando le limitazioni specificate nel parametro *Riquadro di limitazione* ("`bfilmato`", "`bmax`", "`bfotogramma`"). Se si desidera stampare dei fotogrammi specifici del filmato target, assegnare un'etichetta `#p` a tali fotogrammi. Sebbene l'uso dell'azione `printNum` consenta di ottenere risultati qualitativamente superiori rispetto all'azione `printAsBitmapNum`, non è possibile usare `printNum` per stampare filmati con trasparenze alfa o effetti di colore speciali.

Se non si specifica un parametro che delimiti l'area di stampa, quest'ultima viene determinata, per impostazione predefinita, dalla dimensione dello stage del filmato caricato. Il filmato non eredita la dimensione dello stage del filmato principale. È possibile controllare l'area di stampa specificando i parametri `bfilmato`, `bmax` o `bfotogramma`.

Tutti gli elementi stampabili di un filmato devono essere completamente caricati prima dell'inizio della stampa.

La funzione di stampa di Flash Player supporta stampanti PostScript e non PostScript. Le stampanti non PostScript convertono i vettori in bitmap.

Vedere anche

`print`, `printAsBitmap`, `printAsBitmapNum`

`_quality`

Disponibilità

Flash Player 5.

Uso

`_quality`

Descrizione

Proprietà (globale); imposta o recupera la qualità di rendering usata per un filmato. È sempre disponibile una copia dei caratteri dispositivo che non vengono alterati dalla proprietà `_quality`.

È possibile impostare la proprietà `_quality` con i valori riportati di seguito:

- "LOW" Bassa qualità di rendering. I grafici non sono sottoposti ad antialiasing e le bitmap non vengono smussate.
- "MEDIUM" Media qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 2x2 pixel, ma le bitmap non vengono smussate. Questo valore è adatto a filmati che non contengono testo.
- "HIGH" Alta qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 pixel; le bitmap vengono smussate se il filmato è statico. Questa è l'impostazione predefinita per la qualità di rendering usata da Flash.
- "BEST" Ottima qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 pixel; le bitmap vengono sempre smussate.

Esempio

Nell'esempio seguente, la qualità di rendering è impostata su LOW:

```
_quality = "LOW";
```

Vedere anche

`_highquality`, `toggleHighQuality`

random

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta in Flash 5 ed è consigliato l'uso del nuovo metodo `Math.random`.

Uso

```
random(valore)
```

Parametri

valore Un numero intero.

Valori restituiti

Un numero integrale.

Descrizione

Funzione; restituisce un numero intero random tra 0 e 1 meno il numero intero specificato nel parametro *valore*.

Esempio

Nell'esempio seguente, `random` restituisce il valore 0, 1, 2, 3 o 4.

```
random(5);
```

Vedere anche

`Math.random`

removeMovieClip

Disponibilità

Flash Player 4.

Uso

```
removeMovieClip(target)
```

Parametri

target Il percorso *target* di un'istanza di clip filmato creata con `duplicateMovieClip` o il nome di istanza di un clip filmato creato con il metodo `attachMovie` o `duplicateMovie` dell'oggetto `MovieClip`.

Valori restituiti

Nessuno.

Descrizione

Azione; elimina un'istanza di clip filmato creata con il metodo `attachMovie` o `duplicateMovieClip` dell'oggetto `MovieClip` oppure creata con l'azione `duplicateMovieClip`.

Vedere anche

`duplicateMovieClip`, `MovieClip.duplicateMovieClip`, `MovieClip.attachMovie`, `MovieClip.removeMovieClip`

return

Disponibilità

Flash Player 5.

Uso

```
return[espressione]  
return
```

Parametri

espressione Tipo, numero, oggetto, stringa o matrice da valutare e restituire come valore della funzione. Questo parametro è opzionale.

Valori restituiti

Il parametro *espressione* valutato, se disponibile.

Descrizione

Azione; specifica il valore restituito da una funzione. L'azione `return` valuta l'*espressione* e restituisce il risultato come valore della funzione in cui viene eseguita. L'azione `return` interrompe l'esecuzione della funzione e la sostituisce con il valore restituito. Se l'istruzione `return` viene usata da sola, restituisce `null`.

Esempio

Nell'esempio seguente, viene usata l'azione `return` nel corpo della funzione `sum` per restituire il valore sommato dei tre parametri. La riga di codice successiva richiama la funzione `sum` e assegna il valore restituito alla variabile `newValue`:

```
function sum(a, b, c){
    return a + b + c;
}

newValue = sum(4, 32, 78);
trace(newValue);
// Invia 114 alla finestra Output
```

Vedere anche

`function`

_root

Disponibilità

Flash Player 4.

Uso

`_root.Clipfilmato`
`_root.azione`
`_root.proprietà`

Parametri

ClipFilmato Il nome di istanza di un clip filmato.

azione Un'azione o metodo.

proprietà Una proprietà dell'oggetto `MovieClip`.

Descrizione

Proprietà; specifica o restituisce un riferimento alla linea temporale del filmato principale. Se un filmato ha più livelli, la linea temporale del filmato principale si trova sul livello contenente lo script in corso di esecuzione. Ad esempio, se uno script nel livello 1 valuta `_root`, viene restituito `_level1`.

L'uso di `_root` equivale all'uso della notazione della barra inclinata (`/`) per specificare un percorso assoluto all'interno del livello corrente.

Esempio

Nell'esempio seguente, viene interrotta la riproduzione della linea temporale del livello contenente lo script in corso di esecuzione.

```
_root.stop();
```

Nell'esempio seguente, al fotogramma 3 viene inviata la linea temporale del livello corrente.

```
_root.gotoAndStop(3);
```

Vedere anche

`_parent`, `targetPath`

scroll

Disponibilità

Flash Player 4.

Uso

```
textFieldVariableName.scroll = x
```

Descrizione

Proprietà; una proprietà obsoleta per il controllo della visualizzazione delle informazioni in un campo di testo associato a una variabile. La proprietà `scroll` definisce il punto in cui inizia la visualizzazione del contenuto nel campo di testo; una volta impostata la proprietà, Flash Player la aggiorna man mano che l'utente scorre il campo di testo. La proprietà `scroll` è utile per indirizzare gli utenti verso un paragrafo specifico incluso in lunghi passaggi o per creare campi di testo scorrevoli. È possibile recuperare il valore di questa proprietà e modificarla.

Esempio

Il codice seguente è associato al pulsante Su per lo scorrimento del campo di testo `myText`:

```
on (release) {  
    myText.scroll = myText.scroll + 1;  
}
```

Vedere anche

`TextField.maxscroll`, `TextField.scroll`

Selection (oggetto)

L'oggetto `Selection` consente di impostare e controllare in quale campo di testo è posizionato il cursore in un filmato Flash. Il campo di testo attivo è quello su cui è attualmente posizionato il cursore. Gli indici di estensione della selezione iniziano da zero (la prima posizione è 0, la seconda è 1 e così via).

Non esiste un metodo di costruzione per l'oggetto `Selection` poiché è attivo un solo campo alla volta.

Riepilogo dei metodi validi per l'oggetto Selection

Metodo	Descrizione
<code>Selection.addListener</code>	Registra un oggetto per ricevere la notifica quando viene invocato il metodo <code>onSetFocus</code> .
<code>Selection.getBeginIndex</code>	Restituisce l'indice all'inizio dell'estensione di selezione. Restituisce -1 in assenza di indice o di campo correntemente selezionato.
<code>Selection.getCaretIndex</code>	Restituisce la posizione corrente del cursore lampeggiante nell'estensione di selezione correntemente attiva. Restituisce -1 in assenza del cursore lampeggiante o di estensione di selezione correntemente attiva.
<code>Selection.getEndIndex</code>	Restituisce l'indice alla fine dell'estensione di selezione. Restituisce -1 in assenza di indice o di campo correntemente selezionato.
<code>Selection.setFocus</code>	Restituisce il nome della variabile del campo di testo attualmente attivo. Restituisce <code>null</code> se non è attualmente attivo alcun campo di testo.
<code>Selection.removeListener</code>	Rimuove un oggetto precedentemente registrato con <code>addListener</code> .
<code>Selection.setFocus</code>	Rende attivo il campo di testo associato alla variabile specificata nel parametro.
<code>Selection.setSelection</code>	Imposta gli indici iniziale e finale dell'estensione di selezione.

Riepilogo dei metodi listener validi per l'oggetto Selection

Metodo	Descrizione
<code>Selection.onSetFocus</code>	Viene inviata una notifica quando l'elemento attivo cambia.

Selection.addListener

Disponibilità

Flash Player 6.

Uso

```
Selection.addListener(nuovoListener)
```

Parametri

nuovoListener Un oggetto con un metodo `onSetFocus`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto per ricevere le notifiche di modifica di attivazione da tastiera. Quando si modifica l'obiettivo (ad esempio, quando si invoca il metodo `Selection.SetFocus`), ogni oggetto registrato con `addListener` dispone di un proprio metodo `onSetFocus`. È possibile che alle notifiche di variazione dell'obiettivo rispondano più oggetti. Se il listener *nuovoListener* è già registrato, non sono necessarie modifiche.

Selection.getBeginIndex

Disponibilità

Flash Player 5.

Uso

```
Selection.getBeginIndex()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'indice all'inizio dell'estensione di selezione. In assenza di un indice o di un campo attivo, il metodo restituisce -1. Gli indici di estensione della selezione iniziano da zero (ad esempio, la prima posizione è 0, la seconda è 1 e così via).

Selection.getCaretIndex

Disponibilità

Flash Player 5.

Uso

```
Selection.getCaretIndex()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'indice della posizione del cursore lampeggiante. In assenza di cursore lampeggiante visualizzato, il metodo restituisce -1. Gli indici di estensione della selezione iniziano da zero (ad esempio, la prima posizione è 0, la seconda è 1 e così via).

Selection.getEndIndex

Disponibilità

Flash Player 5.

Uso

```
Selection.getEndIndex()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce l'indice finale dell'estensione di selezione attualmente attiva. In assenza dell'indice o dell'estensione di selezione attualmente attiva, il metodo restituisce -1. Gli indici di estensione della selezione iniziano da zero (ad esempio, la prima posizione è 0, la seconda è 1 e così via).

Selection.getFocus

Disponibilità

Flash Player 5. I nomi di istanza per pulsanti e campi di testo sono disponibili in Flash Player 6.

Uso

```
Selection.getFocus()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il nome della variabile del campo di testo attivo. Se nessun campo di testo è attivo, il metodo restituisce `null`. Se il pulsante attualmente attivo è un'istanza dell'oggetto `Button`, `getFocus` restituisce il percorso target come stringa. Se il campo di testo attualmente attivo è un'istanza dell'oggetto `TextField`, `getFocus` restituisce il percorso target come stringa.

Se il clip filmato di un pulsante è il pulsante attualmente attivo, `Selection.getFocus` restituisce il percorso target del clip filmato del pulsante. Se è attualmente attivo un campo di testo con nome di istanza, `Selection.getFocus` restituisce il percorso target dell'oggetto `TextField`. In caso contrario, viene restituito il nome della variabile `Text Field`.

Selection.onSetFocus

Disponibilità

Flash Player 6.

Uso

```
someListener.onSetFocus = function(vecchioFocus, nuovoFocus){  
  istruzioni;  
}
```

Descrizione

Listener; riceve una notifica quando l'elemento attivo cambia. Per usare `onSetFocus`, è necessario creare un oggetto `Listener`. È inoltre possibile definire una funzione per `onSetFocus` e usare il metodo `addListener` per registrare il `Listener` insieme all'oggetto `Selection`, come riportato nell'esempio seguente:

```
someListener = new Object();  
someListener.onSetFocus = function () { ... };  
Selection.addListener(someListener);
```

I listener consentono a parti di codice diverse di operare contemporaneamente poiché più listener possono ricevere notifiche relative a un singolo evento.

Vedere anche

`Selection.addListener`

Selection.removeListener

Disponibilità

Flash Player 6.

Uso

```
Selection.removeListener(listener)
```

Parametri

listener L'oggetto che non riceve più notifiche di attivazione.

Valori restituiti

Se il *listener* è stato rimosso correttamente, il metodo restituisce un valore `true`. Se il *listener* non è stato rimosso correttamente, poiché non era ad esempio incluso nell'elenco dei *listener* dell'oggetto `Selection`, il metodo restituisce `false`.

Descrizione

Metodo; rimuove un oggetto registrato in precedenza con `addListener`.

Selection.setFocus

Disponibilità

Flash Player 5. I nomi di istanza per pulsanti e campi di testo sono disponibili solo in Flash Player 6.

Uso

```
Selection.setFocus("percorsoVariabile")
```

Parametri

percorsoVariabile Una stringa che specifica il percorso relativo al nome di una variabile associata a un campo di testo.

Valori restituiti

Un evento.

Descrizione

Metodo; rende attivo il campo di testo modificabile associato alla variabile specificata in *percorsoVariabile*. Il parametro *percorsoVariabile* deve essere una stringa letterale del percorso relativo alla variabile. È possibile usare la notazione del punto o della barra inclinata per specificare il percorso. Può anche essere usato un percorso relativo o assoluto.

Se un percorso target di un'istanza di pulsante viene passato come parametro di *percorsoVariabile*, il pulsante diventa il nuovo elemento attivo. Se un percorso target di un'istanza di campo di testo viene passato come parametro di *percorsoVariabile*, il campo di testo diventa il nuovo elemento attivo. Se viene inviato null, il focus attuale viene rimosso.

Se il clip filmato di un pulsante viene passato a `Selection.setFocus`, esso diventa il pulsante attualmente attivo. Se viene specificato un oggetto TextField, esso diventa l'elemento attivo. Se viene specificato un oggetto Button, esso diventa il pulsante attualmente attivo.

Esempio

Il seguente esempio imposta il focus su un campo di testo nella linea temporale principale associata alla variabile *myVar*. Poiché il parametro *percorsoVariabile* è un percorso assoluto, è possibile richiamare l'azione da una qualsiasi linea temporale.

```
Selection.setFocus("_root.myVar");
```

Nell'esempio seguente, il campo di testo associato a *myVar* è il clip filmato *myClip* sulla linea temporale principale. È possibile usare uno dei due seguenti percorsi per impostare il focus; il primo è relativo, il secondo è assoluto.

```
Selection.setFocus("myClip.myVar");  
Selection.setFocus("_root.myClip.myVar");
```

Selection.setSelection

Disponibilità

Flash Player 5.

Uso

```
Selection.setSelection(inizio, fine)
```

Parametri

inizio L'indice iniziale dell'estensione di selezione.

fine L'indice finale dell'estensione di selezione.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta l'estensione di selezione del campo di testo attualmente attivo. La nuova estensione di selezione inizia dall'indice specificato nel parametro *inizio* e termina all'indice specificato nel parametro *fine*. Gli indici di estensione della selezione iniziano da zero (ad esempio, la prima posizione è 0, la seconda è 1 e così via). Questo metodo non ha alcun effetto in assenza di campo di testo attualmente attivo.

set variable

Disponibilità

Flash Player 4.

Uso

```
set(variabile, espressione)
```

Parametri

variabile Un identificatore che contiene il valore del parametro *espressione*.

espressione Un valore assegnato alla variabile.

Valori restituiti

Nessuno.

Descrizione

Azione; assegna un valore a una variabile. Una *variabile* è un contenitore di dati. Il contenitore resta invariato, ma il contenuto può variare. Modificando il valore di una variabile durante la riproduzione di un filmato, è possibile registrare e salvare informazioni sulle azioni svolte dall'utente, registrare valori modificati durante la riproduzione del filmato o verificare se una condizione è *true* o *false*.

Le variabili possono contenere qualsiasi tipo di dati (ad esempio, una stringa, un valore booleano, un oggetto o un clip filmato). La linea temporale di ciascun filmato e clip filmato dispone di un proprio insieme di variabili; a ciascuna variabile è assegnato un valore indipendente dalle variabili di altre linee temporali.

ActionScript è un linguaggio a immissione dinamica. Ogni variabile ha un tipo. Tale tipo viene assegnato all'inizio dell'esecuzione e può essere modificato in fase di esecuzione. Non si tratta di un linguaggio a immissione statica come Java o C++, dove il tipo viene assegnato durante la compilazione e non può essere modificato in fase di esecuzione.

Esempio

Nell'esempio seguente, viene impostata una variabile `orig_x_pos` che memorizza la posizione originale dell'asse *x* del clip filmato `ship`, in modo da poterla ripristinare successivamente.

```
on(release) {
    set(orig_x_pos, getProperty ("nave", _x ));
}
```

Il codice precedente produce lo stesso risultato del codice riportato di seguito:

```
on(release) {
    orig_x_pos = ship._x;
}
```

Vedere anche

`var`, `call`

setInterval

Disponibilità

Flash Player 6.

Uso

```
setInterval( funzione, intervallo[, arg1, arg2, ..., argn] )
```

```
setInterval( oggetto, nomeMetodo, intervallo[, arg1, arg2, ..., argn] )
```

Parametri

funzione Un nome di funzione o un riferimento a una funzione anonima.

oggetto Un oggetto derivato dall'oggetto `Object`.

nomeMetodo Il nome del metodo per richiamare il parametro *oggetto*.

intervallo Il tempo in millisecondi tra le chiamate al parametro *funzione* o *nomeMetodo*.

arg1, *arg2*, ..., *argn* I parametri opzionali passati al parametro *funzione* o *nomeMetodo*.

Valori restituiti

Un identificatore di intervallo che è possibile passare alla funzione `clearInterval` per annullare l'intervallo.

Descrizione

Azione; chiama una funzione, un metodo o un oggetto a intervalli regolari durante la riproduzione di un filmato. È possibile usare una funzione di intervallo per aggiornare le variabili da un database o aggiornare una visualizzazione oraria.

Se il valore di *intervallo* è inferiore alla velocità dei fotogrammi del filmato (ad esempio, 10 fotogrammi al secondo equivale a 100 millisecondi), viene richiamata la funzione di intervallo più vicina a tale valore. È necessario usare la funzione `updateAfterEvent` per aggiornare lo schermo con una frequenza adeguata. Se *intervallo* è superiore alla velocità dei fotogrammi del filmato, la funzione di intervallo viene richiamata tutte le volte che l'indicatore di riproduzione accede a un fotogramma per ridurre al minimo l'impatto ad ogni ripristino dello schermo.

Il primo esempio di sintassi riportato sopra corrisponde alla sintassi predefinita per la funzione `setInterval` nel pannello Azioni in modalità normale. Per usare il secondo esempio di sintassi, è necessario usare il pannello Azioni, in modalità esperto.

Esempio

Uso 1: nell'esempio seguente, viene richiamata la funzione anonima ogni 1000 millisecondi (ovvero, ogni secondo).

```
setInterval( function(){ trace("richiamato intervallo"); }, 1000 );
```

Uso 2: nell'esempio seguente vengono definite e richiamate due funzioni di callback. Entrambe le chiamate alla funzione `setInterval` inviano la stringa "richiamato intervallo" alla finestra Output ogni 1000 millisecondi. Quando si richiama `setInterval` la prima volta, viene attivata la funzione `callback1` che contiene un'azione `trace`. Quando si richiama `setInterval` la seconda volta, la stringa "richiamato intervallo" viene passata come parametro alla funzione `callback2`.

```
function callback1() {  
    trace("richiamato intervallo");  
}
```

```
function callback2(arg) {  
    trace(arg);  
}
```

```
setInterval( callback1, 1000 );  
setInterval( callback2, 1000, "richiamato intervallo" );
```

Uso 3: in questo esempio viene usato un metodo di un oggetto. È necessario usare questa sintassi per richiamare un metodo definito per un oggetto. È possibile usare questa sintassi solo in modalità esperto.

```
obj = new Object();  
obj.interval = function() {  
    trace("richiamato funzione di intervallo");  
}
```

```
setInterval( obj, "intervallo", 1000 );
```

```
obj2 = new Object();  
obj2.interval = function(s) {  
    trace(s);  
}  
setInterval( obj2, "intervallo", 1000, "richiamato funzione di intervallo" );
```

È necessario usare la seconda forma della sintassi `setInterval` per richiamare un metodo di un oggetto, come indicato di seguito:

```
setInterval( obj2, "intervallo", 1000, "richiamato funzione di intervallo" );
```

Vedere anche

`clearInterval`, `updateAfterEvent`

setProperty

Disponibilità

Flash Player 4.

Uso

```
setProperty("target", proprietà, valore/espressione)
```

Parametri

target Il percorso per il nome di istanza del clip filmato di cui si desidera impostare la proprietà.

proprietà La proprietà da impostare.

valore Il nuovo valore letterale della proprietà.

espressione Un'equazione che restituisce un nuovo valore della proprietà.

Valori restituiti

Nessuno.

Descrizione

Azione; modifica un valore di proprietà di un clip filmato durante la riproduzione di un filmato.

Esempio

L'istruzione seguente imposta la proprietà `_alpha` del clip filmato `star` al 30 % quando si preme il pulsante:

```
on(release) {  
    setProperty("star", _alpha, "30");  
}
```

Vedere anche

`getProperty`

Sound (oggetto)

L'oggetto `Sound` consente di controllare il suono di un filmato. È possibile aggiungere suoni a un clip filmato dalla libreria mentre il filmato viene riprodotto e controllare tali suoni. Se non si specifica un *target* quando si crea un nuovo oggetto `Sound`, è possibile usare i metodi per controllare il sonoro di tutto il filmato. Prima di richiamare i metodi dell'oggetto `Sound` è necessario crearne un'istanza usando la funzione di costruzione `new Sound`.

L'oggetto `Sound` è supportato in Flash Player 5 e Flash Player 6.

Riepilogo dei metodi validi per l'oggetto Sound

Metodo	Descrizione
<code>Sound.attachSound</code>	Allega il suono specificato nel parametro.
<code>Sound.getBytesLoaded</code>	Restituisce il numero di byte caricati per il suono specificato.
<code>Sound.getBytesTotal</code>	Restituisce la dimensione del suono in byte.
<code>Sound.getPan</code>	Restituisce il valore dell'istruzione <code>setPan</code> precedente.
<code>Sound.getTransform</code>	Restituisce il valore dell'istruzione <code>setTransform</code> precedente.
<code>Sound.getVolume</code>	Restituisce il valore dell'istruzione <code>setVolume</code> precedente.
<code>Sound.loadSound</code>	Carica il file MP3 in Flash Player.
<code>Sound.setPan</code>	Imposta il bilanciamento destra/sinistra dell'audio.
<code>Sound.setTransform</code>	Imposta la quantità di ciascun canale, destro e sinistro, da riprodurre su ciascun altoparlante.
<code>Sound.setVolume</code>	Imposta il livello del volume dell'audio.
<code>Sound.start</code>	Avvia la riproduzione di un suono dall'inizio o a partire dal punto di offset impostato nel parametro.
<code>Sound.stop</code>	Arresta il suono specificato o tutti i suoni in corso di riproduzione.

Riepilogo delle proprietà valide per l'oggetto Sound

Metodo	Descrizione
<code>Sound.duration</code>	Durata di un suono in millisecond.
<code>Sound.position</code>	Numero di millisecondi di riproduzione di un suono.

Riepilogo dei gestori di evento validi per l'oggetto Sound

Metodo	Descrizione
<code>Sound.onLoad</code>	Invocato quando si carica un suono.
<code>Sound.onSoundComplete</code>	Invocato quando si interrompe la riproduzione di un suono.

Funzione di costruzione per l'oggetto Sound

Disponibilità

Flash Player 5.

Uso

```
new Sound([target])
```

Parametri

target L'istanza del clip filmato su cui opera l'oggetto Sound. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un nuovo oggetto Sound per un clip filmato specificato. Se non si specifica un'istanza target, l'oggetto Sound controlla tutti gli elementi audio del filmato.

Esempio

Nell'esempio seguente, viene creata una nuova istanza dell'oggetto Sound denominata `GlobalSound`. La seconda riga richiama il metodo `setVolume` e regola il volume di tutti suoni del filmato al 50%.

```
globalsound = new Sound();
globalsound.setVolume(50);
```

Nell'esempio seguente, una nuova istanza dell'oggetto Sound viene creata e inviata al clip filmato target `myMovie`; per iniziare la riproduzione di tutti i suoni di `myMovie`, viene richiamato il metodo `start`.

```
moviesound = new Sound(myMovie);
moviesound.start();
```

Sound.attachSound

Disponibilità

Flash Player 5.

Uso

```
mySound.attachSound("idNome");
```

Parametri

idNome L'identificatore di un suono esportato nella libreria. L'identificatore è visualizzato nella finestra di dialogo Proprietà di concatenamento del simbolo.

Valori restituiti

Nessuno.

Descrizione

Metodo; associa il suono indicato nel parametro *idNome* all'oggetto Sound specificato. Il suono deve essere nella libreria del filmato corrente e selezionato per l'esportazione nella finestra di dialogo Proprietà di concatenamento del simbolo. È necessario richiamare `Sound.start` per avviare la riproduzione del suono.

Vedere anche

`Sound.start`

Sound.duration

Disponibilità

Flash Player 6.

Uso

```
mySound.duration
```

Descrizione

Proprietà (sola lettura); la durata di un suono in millisecondi.

Sound.getBytesLoaded

Disponibilità

Flash Player 6.

Uso

```
Sound.getBytesLoaded()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che indica il numero dei byte caricati.

Descrizione

Metodo; restituisce il numero di byte caricati (ovvero sottoposti a streaming) per l'oggetto Sound specificato. Il confronto tra il valore restituito da `getBytesLoaded` e il valore restituito da `getBytesTotal` fornisce la percentuale caricata di un suono.

Vedere anche

`Sound.getBytesTotal`

Sound.getBytesTotal

Disponibilità

Flash Player 6.

Uso

```
Sound.getBytesTotal()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che indica la dimensione complessiva, in byte, dell'oggetto Sound specificato.

Descrizione

Metodo; restituisce la dimensione in byte dell'oggetto Sound specificato.

Vedere anche

`Sound.getBytesLoaded`

Sound.getPan

Disponibilità

Flash Player 5.

Uso

```
mySound.getPan();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il livello di bilanciamento impostato l'ultima volta che si è richiamato `setPan`, ovvero un numero intero compreso tra -100 (sx) e 100 (dx). L'impostazione 0 corrisponde a una ripartizione uniforme tra i canali sinistro e destro. Tale impostazione controlla il bilanciamento sx-dx dei suoni attuali e futuri presenti in un filmato.

Questo metodo è cumulabile con il metodo `setVolume` o `setTransform`.

Vedere anche

`Sound.setPan`

Sound.getTransform

Disponibilità

Flash Player 5.

Uso

```
mySound.getTransform();
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce le informazioni sulla trasformazione dell'audio per l'oggetto `Sound` specificato, impostato l'ultima volta che si è richiamato `setTransform`.

Vedere anche

`Sound.setTransform`

Sound.getVolume

Disponibilità

Flash Player 5.

Uso

```
mySound.getVolume()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il livello del volume dell'audio come numero intero compreso tra 0 e 100, dove 0 corrisponde all'assenza di audio e 100 al volume massimo. L'impostazione predefinita è 100.

Vedere anche

`Sound.setVolume`

Sound.loadSound

Disponibilità

Flash Player 6.

Uso

```
mySound.loadSound("url", èStreaming)
```

Parametri

url La posizione su un server di un file audio MP3.

èStreaming Un valore booleano che identifica un suono protratto o momentaneo.

Valori restituiti

Nessuno.

Descrizione

Metodo; carica un file MP3 in un'istanza dell'oggetto Sound. È possibile usare il parametro *èStreaming* per indicare se il suono è protratto o momentaneo.

I suoni momentanei vengono caricati completamente prima della riproduzione, vengono gestiti dall'oggetto Sound di ActionScript e rispondono a tutti i metodi e proprietà di questo oggetto.

I suoni protratti vengono riprodotti durante lo scaricamento. La riproduzione inizia quando è stata ricevuta una quantità di dati sufficiente per avviare il sistema di decompressione.

Analogamente ai suoni momentanei, i suoni protratti esistono solo nella memoria virtuale e non vengono scaricati sul disco rigido.

Esempio

Nell'esempio seguente, viene caricato un suono momentaneo:

```
s.loadSound( "http://serverpath:port/mp3filename", false);
```

Nell'esempio seguente, viene caricato un suono protratto:

```
loadSound( "http://serverpath:port/mp3filename", true);
```

Sound.onLoad

Disponibilità

Flash Player 6.

Uso

```
oggettoSuono.onLoad = funzioneCallback
```

Parametri

oggettoSuono Un oggetto Sound.

funzioneCallback Una funzione.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato automaticamente quando si carica un suono. È necessario creare una funzione che venga eseguita quando si invoca l'evento `onLoad`. È possibile usare una funzione anonima o denominata.

Vedere anche

`Sound.onSoundComplete`

Sound.onSoundComplete

Disponibilità

Flash Player 6.

Uso

oggettoSuono.onSoundComplete = funzioneCallback

Parametri

oggettoSuono Un oggetto Sound.

funzioneCallback Una funzione.

Valori restituiti

Nessuno.

Descrizione

Evento; invocato automaticamente quando termina la riproduzione di un suono. È possibile usare l'evento `onSoundComplete` per avviare gli eventi di un filmato dopo la riproduzione di un suono.

È necessario creare una funzione che venga eseguita quando si invoca l'evento `onSoundComplete`. È possibile usare una funzione anonima o denominata.

Esempio

Uso 1: nell'esempio seguente viene usata una funzione anonima.

```
s = new Sound();
s.attachSound("Suono");
s.onSoundComplete = function() { trace("Suono completato"); };
s.start();
```

Uso 2: nell'esempio seguente viene usata una funzione denominata:

```
function callback1() {
    trace("Suono completato");
}

s = new Sound();
s.attachSound("Suono");

s.onSoundComplete = callback1;

s.start();
```

Sound.position

Disponibilità

Flash Player 6.

Uso

mySound.position

Parametri

Nessuno.

Valori restituiti

Numero di millisecondi di riproduzione di un suono.

Descrizione

Proprietà (sola lettura); restituisce il numero di millisecondi di riproduzione di un suono. Se il suono è diviso in cicli, la posizione viene ripristinata su 0 all'inizio di ciascun ciclo.

Sound.setPan

Disponibilità

Flash Player 5.

Uso

```
mySound.setPan(pan);
```

Parametri

bilanciamento Un numero intero che specifica il bilanciamento sx/dx di un suono. L'intervallo di valori validi è compreso tra -100 e 100 (-100 corrisponde all'uso del solo canale sinistro e 100 all'uso del solo canale destro). L'impostazione 0 ripartisce uniformemente il suono tra i due canali.

Valori restituiti

Nessuno.

Descrizione

Metodo; determina la modalità di riproduzione del suono nei canali destro e sinistro (altoparlanti). Per i suoni mono, *bilanciamento* determina l'altoparlante (destro o sinistro) che diffonde il suono.

Esempio

Nell'esempio seguente, un'istanza dell'oggetto Sound *s* viene creata e associata al suono L7 ripreso dalla libreria. È anche possibile richiamare i metodi `setVolume` e `setPan` per controllare il suono L7.

```
onClipEvent(mouseDown) {  
    // Crea un nuovo oggetto Sound  
    s = new Sound(this);  
    // Collega un suono ripreso dalla libreria  
    s.attachSound("L7");  
    //Imposta il volume al 50%  
    s.setVolume(50);  
    // Disattiva il suono nel canale destro  
    s.setPan(-100);  
    // Viene avviato 30 secondi dopo l'inizio della traccia e riproduce  
    // il suono 5 volte  
    s.start(30, 5);  
}
```

Vedere anche

`Sound.attachSound`, `Sound.setPan`, `Sound.setTransform`, `Sound.setVolume`, `Sound.start`

Sound.setTransform

Disponibilità

Flash Player 5.

Uso

```
mySound.setTransform(oggettoTrasformazioneSuono)
```

Parametri

oggettoTrasformazioneSuono Un oggetto creato tramite la funzione di costruzione dell'oggetto generico Object.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta la trasformazione del suono o le informazioni di bilanciamento per un oggetto Sound.

Il parametro *oggettoTrasformazioneSuono* è un oggetto creato tramite il metodo di costruzione dell'oggetto generico Object, specificando i parametri che indicano la modalità di distribuzione del suono sui canali destro e sinistro (altoparlanti).

L'audio può usare un notevole quantitativo di spazio su disco e di memoria. Poiché i suoni stereo usano una quantità di dati doppia rispetto ai suoni mono, è preferibile usare suoni mono a 22 KHz/6 bit. È possibile usare il metodo `setTransform` per riprodurre l'audio mono in modalità stereo, quello stereo in modalità mono e per aggiungere effetti audio interessanti.

I parametri per *oggettoTrasformazioneSuono* sono i seguenti:

l1 Un valore percentuale che indica il quantitativo di input sinistro da riprodurre nell'altoparlante sinistro (da 0 a 100).

lr Un valore percentuale che indica il quantitativo di input destro da riprodurre nell'altoparlante sinistro (da 0 a 100).

rr Un valore percentuale che indica il quantitativo di input destro da riprodurre nell'altoparlante destro (da 0 a 100).

r1 Un valore percentuale che indica il quantitativo di input sinistro da riprodurre nell'altoparlante destro (da 0 a 100).

Il risultato netto dei parametri è rappresentato dalle formule seguenti:

```
leftOutput = left_input * l1 + right_input * lr  
rightOutput = right_input * rr + left_input * r1
```

I valori di input sinistro o di input destro sono determinati dal tipo (stereo o mono) di suono del filmato.

Nel caso di suoni stereo si ha una ripartizione uniforme dell'input tra gli altoparlanti destro e sinistro, con le seguenti impostazioni di trasformazione:

```
l1 = 100  
lr = 0  
rr = 100  
r1 = 0
```

Le impostazioni di trasformazione predefinite dell'audio mono che riproduce tutto l'input audio nell'altoparlante sinistro sono:

```
ll = 100  
lr = 100  
rr = 0  
rl = 0
```

Esempio

Nell'esempio seguente, viene riportata un'impostazione ottenibile mediante il metodo `setTransform`, ma non tramite i metodi `setVolume` o `setPan`, anche se combinati.

Il codice seguente crea un nuovo oggetto `soundTransformObject` e ne imposta le proprietà in modo che il suono di entrambi i canali venga riprodotto solo attraverso il canale sinistro.

```
mySoundTransformObject = new Object;  
mySoundTransformObject.ll = 100;  
mySoundTransformObject.lr = 100;  
mySoundTransformObject.rr = 0;  
mySoundTransformObject.rl = 0;
```

Per applicare l'oggetto `soundTransformObject` a un oggetto `Sound`, è necessario passare il primo oggetto al secondo impiegando il metodo `setTransform` come indicato di seguito:

```
mySound.setTransform(mySoundTransformObject);
```

Nell'esempio seguente un suono stereo viene riprodotto come mono; `soundTransformObjectMono` ha i parametri indicati di seguito.

```
mySoundTransformObjectMono = new Object;  
mySoundTransformObjectMono.ll = 50;  
mySoundTransformObjectMono.lr = 50;  
mySoundTransformObjectMono.rr = 50;  
mySoundTransformObjectMono.rl = 50;  
mySound.setTransform(soundTransformObjectMono);
```

In questo esempio, il canale sinistro è attivo solo al 50% delle capacità e la parte residua del canale viene aggiunta al canale destro; `soundTransformObjectHalf` ha i parametri riportati di seguito.

```
mySoundTransformObjectHalf = new Object;  
mySoundTransformObjectHalf.ll = 50;  
mySoundTransformObjectHalf.lr = 0;  
mySoundTransformObjectHalf.rr = 100;  
mySoundTransformObjectHalf.rl = 50;  
setTransform(soundTransformObjectHalf);
```

Vedere anche

Funzione di costruzione per l'oggetto `Object`

Sound.setVolume

Disponibilità

Flash Player 5.

Uso

```
mySound.setVolume(volume)
```

Parametri

volume Un numero compreso tra 0 e 100 che indica un livello del volume. 100 corrisponde al massimo volume e 0 all'assenza di audio. L'impostazione predefinita è 100.

Valori restituiti

Nessuno.

Descrizione

Metodo; imposta il volume per l'oggetto Sound.

Esempio

L'esempio seguente imposta il volume al 50% e quindi trasferisce l'audio dall'altoparlante sinistro a quello destro.

```
onClipEvent (load) {  
    i = -100;  
    s = new Sound();  
    s.setVolume(50);  
}  
onClipEvent (enterFrame) {  
    if (i <= 100) {  
        S.setPan(i++);  
    }  
}
```

Vedere anche

Sound.setPan, Sound.setTransform

Sound.start

Disponibilità

Flash Player 5.

Uso

```
mySound.start([OffsetInSecondi, ciclo])
```

Parametri

OffsetInSecondi Un parametro opzionale che consente di avviare la riproduzione del suono da un punto specifico. Ad esempio, se si desidera avviare la riproduzione di un suono della durata di 30 secondi da metà, è sufficiente specificare 15 come parametro *OffsetInSecondi*. Il suono non viene ritardato di 15 secondi, ma viene invece avviato all'altezza dell'indicatore dei 15 secondi.

ciclo Un parametro opzionale che consente di specificare il numero di riproduzioni consecutive di un suono.

Valori restituiti

Nessuno.

Descrizione

Metodo; avvia la riproduzione dell'ultimo suono associato dall'inizio, se non è stato specificato alcun parametro, o a partire dal punto indicato nel parametro *OffsetInSeconds*.

Vedere anche

`Sound.stop`

Sound.stop

Disponibilità

Flash Player 5.

Uso

```
mySound.stop(["idNome"]);
```

Parametri

idNome Un parametro opzionale che determina l'interruzione della riproduzione di un determinato suono. Il parametro *idNome* deve essere racchiuso tra virgolette (" ").

Valori restituiti

Nessuno.

Descrizione

Metodo; interrompe la riproduzione di tutti i suoni se non è specificato alcun parametro, oppure dell'unico suono specificato nel parametro *idNome*.

Vedere anche

`Sound.start`

_soundbuftime

Disponibilità

Flash Player 4.

Uso

```
_soundbuftime = intero
```

Parametri

intero Il numero di secondi che precedono l'inizio dello streaming del filmato.

Descrizione

Proprietà (globale); stabilisce il numero di secondi di streaming dell'audio da accumulare nel buffer prima della riproduzione. Il valore predefinito è 5 secondi.

Stage (oggetto)

L'oggetto Stage è un oggetto di primo livello a cui si può accedere senza usare una funzione di costruzione.

Usare i metodi e le proprietà di questo oggetto per accedere ed elaborare le informazioni relative alle limitazioni di un filmato Flash.

L'oggetto Stage è disponibile in Flash Player 6 e versione successiva.

Riepilogo dei metodi validi per l'oggetto Stage

Metodo	Descrizione
<code>Stage.addListener</code>	Aggiunge un oggetto Listener all'oggetto Stage.
<code>Stage.removeListener</code>	Rimuove un oggetto Listener dall'oggetto Stage.

Riepilogo delle proprietà valide per l'oggetto Stage

Metodo	Descrizione
<code>Stage.align</code>	Allineamento del filmato Flash nel browser.
<code>Stage.height</code>	Altezza dello Stage, in pixel.
<code>Stage.width</code>	Larghezza dello Stage, in pixel.
<code>Stage.scaleMode</code>	L'attuale ridimensionamento in scala del filmato Flash.

Riepilogo dei gestori di evento validi per l'oggetto Stage

Metodo	Descrizione
<code>Stage.onResize</code>	Indica che un filmato è stato ridimensionato.

Stage.addListener

Disponibilità

Flash Player 6.

Uso

```
Stage.addListener(Listener)
```

Parametri

Listener Un oggetto che percepisce la notifica di callback dall'evento `onResize`.

Valori restituiti

Nessuno.

Descrizione

Metodo; rileva il ridimensionamento di un filmato Flash `Stage.scaleMode = "noScale"`. Il metodo `addListener` non funziona con l'impostazione di ridimensionamento in scala del filmato predefinito ("`showAll`") o altre impostazioni di ridimensionamento in scala ("`exactFit`" e "`noBorder`").

Per usare `addListener`, è necessario creare un *oggetto Listener*. Un oggetto `Listener` è un oggetto che riceve notifica da un evento attivato in un filmato. Gli oggetti `Listener` dell'oggetto `Stage` ricevono notifica da `Stage.onResize`.

Esempio

In questo esempio viene creato un nuovo oggetto `Listener` chiamato `myListener`. L'oggetto `myListener` viene usato per richiamare `onResize` e definire una funzione da richiamare in caso di attivazione di `onResize`. Infine, il codice aggiunge l'oggetto `myListener` all'elenco di callback dell'oggetto `Stage`. Gli oggetti `Listener` consentono a più oggetti di percepire le notifiche di ridimensionamento.

```
myListener = new Object();
myListener.onKeyDown = function () {
    Stage.addListener(Listener);
};
```

Stage.align

Disponibilità

Flash Player 6.

Uso

`Stage.align`

Descrizione

Proprietà; indica l'attuale allineamento del filmato Flash nello stage.

Nella seguente tabella vengono elencati i valori per la proprietà `align`. Qualsiasi valore non inserito in questo elenco posiziona il filmato al centro dell'area dello `Stage`.

Valore	Verticale	Orizzontale
"T"	sopra	centro
"B"	sotto	centro
"L"	centro	sinistra
"R"	centro	destra
"TL"	sopra	sinistra
"TR"	sopra	destra
"BL"	sotto	sinistra
"BR"	sotto	destra

Stage.height

Disponibilità

Flash Player 6.

Uso

Stage.height

Descrizione

Proprietà (sola lettura); indica l'altezza attuale, in pixel, dello Stage del filmato Flash. Quando la proprietà `Stage.noScale` è `true`, `height` rappresenta l'altezza di Flash Player. Quando il valore `Stage.noScale` è `false` (in questo caso il filmato viene ridimensionato in scala quando si modificano le dimensioni della finestra del lettore), `height` rappresenta l'altezza del filmato Flash.

Stage.onResize

Disponibilità

Flash Player 6.

Uso

```
Stage.onResize() = function() {...}
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo di callback; indica che il filmato Flash è stato ridimensionato. È possibile usare questo evento per scrivere una funzione che collochi gli oggetti sullo Stage quando un filmato viene ridimensionato.

Stage.removeListener

Disponibilità

Flash Player 6.

Uso

```
Stage.removeListener(Listener)
```

Parametri

Listener Un oggetto viene aggiunto all'elenco di callback di un oggetto con il metodo `addListener`.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove un oggetto `Listener` creato con `addListener`.

Vedere anche

`Stage.addListener`

Stage.scaleMode

Disponibilità

Flash Player 6.

Uso

```
Stage.scaleMode = "valore"
```

Descrizione

Proprietà; indica l'attuale ridimensionamento in scala del filmato Flash nello Stage. La proprietà `scaleMode` forza il filmato in una modalità di ridimensionamento in scala specifico. Per impostazione predefinita, il filmato usa i parametri HTML impostati nella finestra di dialogo delle impostazioni di pubblicazione.

La proprietà `scaleMode` può usare i valori "exactFit", "showAll", "noBorder" e "noScale". Qualsiasi altro valore imposta la proprietà `scaleMode` in base all'impostazione predefinita "showAll".

Stage.width

Disponibilità

Flash Player 6.

Uso

```
Stage.width
```

Descrizione

Proprietà (sola lettura); indica l'altezza attuale, in pixel, dello Stage del filmato Flash. Quando il valore di `Stage.noScale` è `true`, la proprietà `width` rappresenta la larghezza della finestra del lettore. Quando il valore di `Stage.noScale` è `false` (in questo caso il filmato viene ridimensionato in scala quando si modificano le dimensioni della finestra del lettore), `width` rappresenta la larghezza del filmato Flash.

startDrag

Disponibilità

Flash Player 4.

Uso

```
startDrag(target [,bloccato [,sinistra , superiore , destra, inferiore])
```

Parametri

target Il percorso target del clip filmato da trascinare.

bloccato Un valore booleano che specifica se il clip filmato mobile è bloccato al centro rispetto alla posizione del mouse (*true*) oppure ancorato al punto del clip filmato selezionato inizialmente dall'utente (*false*). Questo parametro è opzionale.

sinistra, superiore, destra, inferiore Valori relativi alle coordinate del filmato principale del clip filmato che definiscono un rettangolo di delimitazione per quest'ultimo. Si tratta di parametri opzionali.

Valori restituiti

Nessuno.

Descrizione

Azione; rende il clip filmato *target* trascinabile durante la riproduzione del filmato. È possibile trascinare un solo clip filmato alla volta. Una volta eseguita un'operazione `startDrag`, il clip filmato rimane trascinabile fino all'attivazione esplicita di un'azione `stopDrag` o di un'azione `startDrag` per un altro clip filmato.

Esempio

Per creare un clip filmato che gli utenti possono collocare in qualsiasi posizione, associare le azioni `startDrag` e `stopDrag` a un pulsante all'interno del clip filmato.

```
on(press) {  
    startDrag(this,true);  
}  
on(release) {  
    stopDrag();  
}
```

Vedere anche

`MovieClip._droptarget`, `stopDrag`

stop

Disponibilità

Flash 2.

Uso

`stop`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; interrompe il clip filmato attualmente in riproduzione. Questa azione viene comunemente usata per controllare i clip filmato tramite pulsanti.

stopAllSounds

Disponibilità

Flash Player 3.

Uso

`stopAllSounds()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; interrompe tutti i suoni in corso di riproduzione in un filmato senza arrestare l'indicatore di riproduzione. La riproduzione dell'audio in streaming ricomincia quando l'indicatore di riproduzione si sposta sui fotogrammi in cui si trovano.

Esempio

Il codice seguente è applicabile a un pulsante che interrompe tutti i suoni di un filmato quando viene premuto.

```
on(release) {  
    stopAllSounds();  
}
```

Vedere anche

Sound (oggetto)

stopDrag

Disponibilità

Flash Player 4.

Uso

```
stopDrag()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; arresta l'operazione di trascinamento corrente.

Esempio

Questa istruzione interrompe l'azione di trascinamento sull'istanza `mc` quando il pulsante del mouse viene rilasciato:

```
on(press) {  
    startDrag("mc");  
}  
on(release) {  
    stopdrag();  
}
```

Vedere anche

MovieClip.stopDrag, MovieClip._droptarget, startDrag,

String (funzione)

Disponibilità

Flash Player 4.

Uso

`String(espressione)`

Parametri

espressione Un'espressione da convertire in stringa.

Valori restituiti

Nessuno.

Descrizione

Funzione; restituisce una rappresentazione in formato stringa del parametro specificato, come indicato di seguito:

Se *espressione* è un valore booleano, la stringa restituita è `true` o `false`.

Se *espressione* è un numero, la stringa restituita è la rappresentazione testuale del numero.

Se *espressione* è una stringa, la stringa restituita è *espressione*.

Se *espressione* è un oggetto, il valore restituito è una rappresentazione in formato stringa dell'oggetto generato richiamando la relativa proprietà `String` oppure `object.toString`, in assenza di tale proprietà.

Se *espressione* è un clip filmato, il valore restituito è il percorso target del clip filmato nella notazione della barra inclinata (/).

Se *espressione* è `undefined`, il valore restituito è una stringa vuota (`"`).

Vedere anche

`Number.toString`, `Object.toString`, `String` (oggetto), `" "` (delimitatore di stringa)

" " (delimitatore di stringa)

Disponibilità

Flash Player 4.

Uso

`"testo"`

Parametri

testo Un carattere.

Valori restituiti

Nessuno.

Descrizione

Delimitatore di stringa; se usato prima e dopo i caratteri, le virgolette indicano che i caratteri hanno un valore letterale e sono considerati una *stringa* e non una variabile, un valore numerico o un altro elemento di `ActionScript`.

Esempio

In questo esempio, le virgolette indicano che il valore della variabile *yourGuess* corrisponde alla stringa letterale "Isola del Principe Edoardo" e non al nome della variabile. Il valore di *province* è una variabile non un valore letterale. Per determinare il valore di *province*, è necessario individuare il valore di *yourGuess*.

```
yourGuess = "Isola del principe Edoardo";
on(release){
    province = yourGuess
    trace(province);
}

// Visualizza Isola del Principe Edoardo nella finestra Output
```

Vedere anche

[String \(oggetto\)](#), [String \(funzione\)](#)

String (oggetto)

L'oggetto `String` è un wrapper per il tipo di dati di base stringa che consente di usare i metodi e le proprietà dell'oggetto `String` per elaborare i valori di tipo stringa. È possibile convertire il valore di qualsiasi oggetto in una stringa tramite la funzione `String()`. In Flash MX, l'oggetto `String` è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

Tutti i metodi dell'oggetto `String` sono generici, ad eccezione di `concat`, `fromCharCode`, `slice` e `substr`. In questo modo, i metodi stessi chiamano `this.toString` prima di effettuare le relative operazioni e possono quindi essere usati con oggetti diversi da `String`.

Poiché tutti gli indici di stringa iniziano da 0, l'indice dell'ultimo carattere per qualsiasi stringa `x` è il seguente:

```
x.length - 1
```

È possibile richiamare qualsiasi metodo dell'oggetto `String` usando il metodo di costruzione `new String` o un valore letterale di tipo stringa. Se si specifica un letterale stringa, l'interprete di ActionScript lo converte automaticamente in un oggetto `String` temporaneo, chiama il metodo, quindi elimina l'oggetto creato. È inoltre possibile usare la proprietà `String.length` con un letterale stringa.

È importante non confondere un letterale stringa e un'istanza dell'oggetto `String`. Nell'esempio seguente, la prima riga di codice crea il letterale stringa `s1` e la seconda riga di codice crea un'istanza `s2` dell'oggetto `String`.

```
s1 = "foo"
s2 = new String("foo")
```

Usare i letterali stringa a meno che non sia necessario usare un oggetto `String`.

Riepilogo dei metodi validi per l'oggetto String

Metodo	Descrizione
<code>String.charAt</code>	Restituisce il carattere in una posizione specifica all'interno di una stringa.
<code>String.charCodeAt</code>	Restituisce il valore del carattere in una determinata posizione come numero intero a 16 bit compreso tra 0 e 65535.
<code>String.concat</code>	Concatena il testo di due stringhe e restituisce una nuova stringa.
<code>String.fromCharCode</code>	Restituisce una stringa composta dai caratteri specificati nei parametri.
<code>String.indexOf</code>	Ricerca la stringa e restituisce l'indice della sottostringa specificata nei parametri. Se il valore è presente più volte, viene restituito l'indice della prima occorrenza. Se il valore è assente, viene restituito -1.
<code>String.lastIndexOf</code>	Restituisce l'indice dell'ultima sottostringa all'interno della stringa che appare prima della posizione iniziale specificata nel parametro o -1 se la sottostringa non viene trovata.
<code>String.slice</code>	Estrae una sezione di una stringa e restituisce una nuova stringa.
<code>String.split</code>	Divide un oggetto String in una matrice di stringhe separandolo in sottostringhe.
<code>String.substr</code>	Restituisce il numero di caratteri specificato in una stringa, iniziando dalla posizione specificata nel parametro.
<code>String.substring</code>	Restituisce i caratteri tra i due indici specificati nei parametri come stringa.
<code>String.toLowerCase</code>	Converte la stringa in lettere minuscole e restituisce il risultato; non modifica il contenuto dell'oggetto originale.
<code>String.toUpperCase</code>	Converte la stringa in lettere maiuscole e restituisce il risultato; non modifica il contenuto dell'oggetto originale.

Riepilogo delle proprietà valide per l'oggetto String

Proprietà	Descrizione
<code>String.length</code>	Restituisce la lunghezza della stringa.

Funzione di costruzione dell'oggetto String

Disponibilità

Flash Player 5.

Uso

```
new String(valore)
```

Parametri

valore Il valore iniziale del nuovo oggetto String.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un nuovo oggetto String.

Vedere anche

`String` (funzione), " " (delimitatore di stringa)

String.charAt

Disponibilità

Flash Player 5.

Uso

```
myString.charAt(indice)
```

Parametri

indice Il numero del carattere nella stringa da restituire.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce il carattere nella posizione specificata dal parametro *indice*. L'indice del primo carattere in una stringa è 0. Se *indice* non è un numero compreso tra 0 e `string.length - 1`, viene restituita una stringa vuota.

String.charCodeAtAt

Disponibilità

Flash Player 5.

Uso

```
myString.charCodeAtAt(indice)
```

Parametri

indice Un numero intero che specifica la posizione di un carattere nella stringa. Il primo carattere è indicato da 0 e l'ultimo carattere è indicato da `myString.length-1`.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce un numero intero a 16 bit da 0 a 65535 che rappresenta il carattere specificato dall'*indice*.

Questo metodo è simile a `string.charAt` con l'unica differenza che il valore restituito è un codice di carattere di un numero intero a 16 bit, non un carattere.

Esempio

Nell'esempio seguente, il metodo `charCodeAt` viene richiamato alla prima lettera della stringa "Mauro".

```
s = new String("Mauro");  
i = s.charCodeAt(0);  
// i = 67
```

String.concat

Disponibilità

Flash Player 5.

Uso

```
myString.concat(valore0,...valoreN)
```

Parametri

valore0,...*valoreN* Zero o più valori da concatenare.

Valori restituiti

Nessuno.

Descrizione

Metodo; unisce il valore dell'oggetto String ai parametri e restituisce la stringa appena formata; il valore originale *myString* non viene modificato.

String.fromCharCode

Disponibilità

Flash Player 5.

Uso

```
String.fromCharCode(c1,c2,...cN)
```

Parametri

c1,*c2*,...*cN* Numeri interi decimali che rappresentano i valori ASCII.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una stringa costituita da caratteri rappresentati dai valori ASCII nei parametri.

Esempio

In questo esempio, il metodo `fromCharCode` viene usato per inserire un carattere "@" nell'indirizzo di posta elettronica.

```
address = "dog" + String.fromCharCode(64) + "house.net";  
trace(address);  
// output: dog@house.net
```

String.indexOf

Disponibilità

Flash Player 5.

Uso

```
myString.indexOf(sottostringa, [avviaIndice])
```

Parametri

sottostringa Un numero intero o una stringa che specifica la sottostringa da ricercare all'interno di *myString*.

avviaIndice Un numero intero che specifica il punto iniziale di *myString* in cui cercare la sottostringa. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; ricerca nella stringa e restituisce la posizione della prima occorrenza della *sottostringa* specificata. Se il valore non viene trovato, il metodo restituisce -1.

String.lastIndexOf

Disponibilità

Flash Player 5.

Uso

```
myString.lastIndexOf(sottostringa, [avviaIndice])
```

Parametri

sottostringa Un numero intero o una stringa che specifica la stringa da ricercare.

avviaIndice Un numero intero che specifica il punto iniziale da ricercare nella *sottostringa*. Questo parametro è opzionale.

Valori restituiti

Nessuno.

Descrizione

Metodo; ricerca la stringa da destra a sinistra e restituisce l'indice dell'ultima occorrenza della *sottostringa* individuata prima di *avviaIndice* nella stringa contenente l'istruzione. Se la *sottostringa* non è presente, il metodo restituisce -1.

String.length

Disponibilità

Flash Player 5.

Uso

```
string.length
```

Parametri

Nessuno.

Descrizione

Proprietà; restituisce il numero di caratteri nell'oggetto String specificato.

String.slice

Disponibilità

Flash Player 5.

Uso

```
Stringa.slice(inizio, [fine])
```

Parametri

inizio Un numero che specifica l'indice del punto di inizio della porzione. Se *inizio* è un numero negativo, il punto di inizio è specificato a partire dalla fine della stringa, dove -1 è l'ultimo carattere.

fine Un numero che specifica l'indice del punto finale della porzione. Se *fine* non è specificato, la porzione include tutti i caratteri dall'*inizio* alla fine della stringa. Se *fine* è un numero negativo, il punto finale è specificato a partire dalla fine della stringa, dove -1 è l'ultimo carattere.

Valori restituiti

Nessuno.

Descrizione

Metodo; estrae una porzione o una sottostringa dell'oggetto String specificato e la restituisce come nuova stringa senza modificare l'oggetto String originale. La stringa restituita include il carattere *inizio* e tutti i caratteri compresi tra questo e il carattere identificato da *fine* che è escluso.

Esempio

Nell'esempio seguente, viene impostata una variabile `text` e creata un'istanza dell'oggetto String `s` da passare alla variabile `text`. Il metodo `slice` consente di estrarre una sezione della stringa contenuta nella variabile, che verrà inviata dall'azione `trace` alla finestra Output.

```
text = "lexington";  
s = new String( text );  
trace(s.slice( 1, 3 ));  
trace(s);
```

Nella finestra Output viene visualizzato il risultato `ex`.

Il codice indicato di seguito produce lo stesso risultato; tuttavia, alla funzione String viene passata una stringa e non una variabile.

```
s = new String( "lexington" );  
trace(s.slice( 1, 3 ));  
trace(s);
```

Nella finestra Output viene visualizzato il risultato `ex`.

String.split

Disponibilità

Flash Player 5.

Uso

```
myString.split("delimitatore", [limite])
```

Parametri

delimitatore Il carattere o la stringa in corrispondenza della quale *myString* si divide. Se il parametro *delimitatore* non è definito, l'intera stringa viene inserita nel primo elemento della matrice.

limite Il numero di elementi da inserire nella matrice. Questo parametro è opzionale.

Valori restituiti

Una matrice contenente le sottostringhe di *myString*.

Descrizione

Metodo; divide un oggetto String in sottostringhe ogni qualvolta incontra il parametro *delimitatore* specificato e restituisce le sottostringhe in una matrice. Se si usa una stringa vuota ("") come delimitatore, ciascun carattere della stringa viene inserito come elemento nella matrice, come nel codice seguente.

```
myString = "Leo";  
i = myString.split("");  
trace (i);
```

Nella finestra Output viene visualizzato il risultato seguente:

```
L, E, O
```

Se il parametro *delimitatore* non è definito, l'intera stringa viene inserita nel primo elemento della matrice restituita.

Esempio

Nell'esempio seguente viene restituita una matrice con cinque elementi.

```
myString = "M, A, R, I, A";  
myString.split(",");
```

Nell'esempio seguente viene restituita una matrice con due elementi.

```
myString.split(", ", 2);
```

String.substr

Disponibilità

Flash Player 5.

Uso

```
myString.substr(inizio, lunghezza)
```

Parametri

inizio Un numero intero che indica la posizione del primo carattere in *myString* da usare per creare una sottostringa. Se *inizio* è un numero negativo, il punto di inizio è specificato a partire dalla fine della stringa, dove -1 è l'ultimo carattere.

lunghezza Il numero di caratteri nella sottostringa creata. Se *lunghezza* non è specificato, la sottostringa include tutti i caratteri dall'inizio alla fine della stringa.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce i caratteri in una stringa dall'indice specificato nel parametro *inizio* attraverso il numero di caratteri specificato nel parametro *lunghezza*. Il metodo *substr* non modifica la stringa specificata da *myString* e restituisce una nuova stringa.

String.substring

Disponibilità

Flash Player 5.

Uso

```
String.substring(da, a)
```

Parametri

da Un numero intero che indica la posizione del primo carattere di *myString* usato per creare la sottostringa. I valori validi per il parametro *da* sono compresi tra 0 e `string.length - 1`. Se *da* è un valore negativo, viene usato 0.

a Un numero intero corrispondente a 1+ l'indice dell'ultimo carattere da estrarre in *myString*. I valori validi per *a* sono compresi tra 1 e `string.length`. Il carattere indicizzato dal parametro *a* non è incluso nella stringa estratta. Se il parametro viene ommesso, si usa `string.length`. Se il parametro è un valore negativo, viene usato 0.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una stringa costituita dai caratteri inclusi tra i punti specificati dai parametri *da* e *a*. Se il parametro *a* non è specificato, la fine della sottostringa corrisponde alla fine della stringa. Se il valore *da* equivale al valore *a*, il metodo restituisce una stringa vuota. Se il valore *da* è maggiore del valore *a*, i parametri vengono automaticamente scambiati prima dell'esecuzione della funzione e il valore originale rimane invariato.

String.toLowerCase

Disponibilità

Flash Player 5.

Uso

```
myString.toLowerCase()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una copia dell'oggetto String con tutti i caratteri maiuscoli convertiti in minuscoli. Il valore originale non viene modificato.

String.toUpperCase

Disponibilità

Flash Player 5.

Uso

```
myString.toUpperCase()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce una copia dell'oggetto String con tutti i caratteri minuscoli convertiti in maiuscoli. Il valore originale non viene modificato.

substring

Disponibilità

Flash Player 4. Questa funzione è diventata obsoleta ed è preferibile usare `String.substr`.

Uso

```
substring("stringa", indice, numero)
```

Parametri

stringa La stringa da cui estrarre la nuova stringa.

indice Il numero del primo carattere da estrarre.

numero Il numero di caratteri da includere nella stringa estratta, escluso il carattere specificato nell'argomento indice.

Valori restituiti

Nessuno.

Descrizione

Funzione di stringa; estrae parte di una stringa. Questa funzione inizia da 1 mentre i metodi dell'oggetto String iniziano da 0.

Vedere anche

`String.substr`

super

Disponibilità

Flash Player 6.

Uso

```
super.metodo([arg1, ..., argN])
```

```
super([arg1, ..., argN])
```


Parametri

metodo Il metodo da invocare nella superclasse.

arg1 I parametri opzionali passati alla versione superclasse del metodo (sintassi 1) o alla funzione di costruzione della superclasse (sintassi 2).

Valori restituiti

Entrambe le forme invocano una funzione. La funzione può restituire qualsiasi valore.

Descrizione

Operatore: è possibile usare il primo stile sintattico all'interno del corpo del metodo di un oggetto per invocare la versione superclasse di un metodo; è inoltre possibile passare opzionalmente dei parametri (*arg1 ... argN*) al metodo della superclasse. Ciò consente sia di creare metodi di sottoclasse per potenziare i metodi di superclasse sia di invocare metodi di superclasse che si comportino normalmente.

È possibile usare il secondo stile sintattico all'interno del corpo di una funzione di costruzione per invocare la versione superclasse della funzione di costruzione e per passarla opzionalmente ai parametri. Ciò consente di creare una sottoclasse per eseguire un'inizializzazione supplementare e di invocare la funzione di costruzione della superclasse per eseguire l'inizializzazione di quest'ultima.

Esempio

Nell'esempio seguente vengono create due classi, `ParentClass` e `ChildClass`. Si definisce inoltre un metodo (`method`) per ciascuna classe. Ciascuna definizione di metodo dispone di un'azione `trace` per inviare un messaggio alla finestra `Output`. Dalla seconda all'ultima riga del codice viene creata un'istanza di classe `ChildClass`, con richiamo del metodo (`method`) corrispondente:

```
function ParentClass() {
}
ParentClass.prototype.method = function () {
    trace("Implementazione ParentClass di metodo");
};
function ChildClass() {
}
ChildClass.prototype = new ParentClass();
ChildClass.prototype.method = function () {
    trace("Implementazione ChildClass di metodo");
    super.method();
};
x = new ChildClass();
x.method();
```

Nella finestra `Output` viene visualizzato il seguente risultato:

```
Implementazione ChildClass di metodo
Implementazione ParentClass di metodo
```

Nell'esempio seguente, i parametri vengono passati alla superfunzione di costruzione:

```
function SuperClass(arg){
    trace("È stata passata la funzione di costruzione SuperClass " + arg);
}
function SubClass(arg){
    super(arg);
    trace("Funzione di costruzione SubClass");
}
```

switch

Disponibilità

Flash Player 4.

Uso

```
switch (espressione){  
    Clausolacase:  
    [Clausolapredefinita:]  
}
```

Parametri

espressione Un'espressione qualsiasi.

Clausolacase Una parola chiave di tipo *case* seguita da un'espressione, da una virgola e da un gruppo di istruzioni da eseguire se l'espressione coincide con il parametro di *espressione* `switch`, per il criterio di uguaglianza rigorosa (`===`).

Clausolapredefinita Una parola chiave *predefinita* seguita dalle istruzioni da eseguire quando nessuna delle espressioni *case* coincide con il parametro di *espressione* `switch`, per il criterio di uguaglianza rigorosa (`===`).

Valori restituiti

Nessuno.

Descrizione

Azione; crea una struttura ramificata per le istruzioni di ActionScript. Analogamente all'azione `if`, l'azione `switch` verifica una condizione ed esegue le istruzioni se la condizione restituisce il valore `true`.

Esempio

Nell'esempio seguente, se il parametro `numero` restituisce 1 viene eseguita l'azione `trace` successiva a `case 1`; se il parametro `numero` restituisce 2, viene eseguita l'azione `trace` successiva a `case 2` e così via. Se nessuna espressione `case` coincide con il parametro `numero`, viene eseguita l'azione `trace` successiva alla parola chiave `default`.

```
switch (numero) {  
    case 1:  
        trace ("la verifica di case 1 ha restituito true");  
        break;  
    case 2:  
        trace ("la verifica di case 2 ha restituito true");  
        break;  
    case 3:  
        trace ("la verifica di case 3 ha restituito true");  
        break;  
    default:  
        trace ("nessuna verifica di case restituisce true")  
}
```

Nell'esempio seguente, non vi è un comando break nel primo gruppo di case. In questo caso se il numero è 1, A e B vengono inviati alla finestra Output:

```
switch (numero) {
  case 1:
    trace ("A");
  case 2:
    trace ("B");
    break;
  default
    trace ("D")
}
```

Vedere anche

=== (uguaglianza rigorosa), break, case, default, if

System (oggetto)

È un oggetto di livello superiore contenente l'oggetto Capabilities. È necessario usare l'oggetto Capabilities e le proprietà corrispondenti. Ad esempio, il codice sotto indicato verifica se il sistema dispone di funzioni audio.

```
System.capabilites.hasAudio
```

System.capabilities (oggetto)

È possibile usare l'oggetto System.capabilities per determinare le capacità del lettore e del sistema in cui risiede il filmato Flash. Questo consente di adattare il contenuto a formati diversi. Ad esempio, lo schermo di un telefono cellulare (bianco e nero, 100 pixel quadrati) è diverso dallo schermo (a colori, 1000 pixel quadrati) del computer. Per fornire un contenuto soddisfacente al maggior numero di utenti, è possibile usare l'oggetto Capabilities per determinare il tipo di dispositivo in uso. Si può quindi impostare il server in modo da inviare file SWF diversi a seconda delle capacità del dispositivo oppure programmare la presentazione del filmato Flash in modo che vari a seconda delle capacità del dispositivo.

Le informazioni sulle capacità possono essere inviate tramite il metodo GET o POST HTTP. Di seguito, viene fornito l'esempio di una stringa di server per un dispositivo sprovvisto di supporto MP3, con schermo di 400 x 200 pixel, 8 x 4 cm:

```
"A=t&MP3=f&AE=gsm&VE=h11&ACC=f&V=WIN%206%2C0%2C0%2C129&M=Macromedia%WINDOWS&R=400x200&DP=72&COL=color&AR=1.0&OS=WINDOWS%2000&L=en-US"
```

L'oggetto Capabilities è disponibile in Flash Player 6.

Tutte le proprietà dell'oggetto Capabilities sono accessibili tramite l'oggetto System.capabilities.

Riepilogo delle proprietà valide per l'oggetto Capabilities

Proprietà	Descrizione
<code>System.capabilities.hasAudioEncoder</code>	Indica i codificatori audio supportati.
<code>System.capabilities.hasAccessibility</code>	Indica se il dispositivo soddisfa gli standard di accesso.
<code>System.capabilities.hasAudio</code>	Indica se il dispositivo dispone di capacità audio.
<code>System.capabilities.hasMP3</code>	Indica se il dispositivo è dotato di decodificatore MP3.
<code>System.capabilities.language</code>	Indica il linguaggio supportato da Flash Player.
<code>System.capabilities.manufacturer</code>	Indica il produttore di Flash Player.
<code>System.capabilities.os</code>	Indica il sistema operativo utilizzato per Flash Player.
<code>System.capabilities.pixelAspectRatio</code>	Indica le dimensioni dello schermo in pixel.
<code>System.capabilities.screenColor</code>	Indica se lo schermo è a colori, in bianco e nero, in scala di grigi e così via.
<code>System.capabilities.screenDPI</code>	Indica le dimensioni dpi dello schermo.
<code>System.capabilities.screenResolution.x</code>	Indica la larghezza dello schermo.
<code>System.capabilities.screenResolution.y</code>	Indica l'altezza dello schermo.
<code>System.capabilities.version</code>	Indica la prima versione di Flash Player supportata.
<code>System.capabilities.hasVideoEncoder</code>	Indica i codificatori video supportati.

System.capabilities.hasAudioEncoder

Disponibilità

Flash Player 6.

Uso

`System.capabilities.hasAudioEncoder`

Descrizione

Proprietà; una matrice di decodificatori audio. La stringa per il server è `AE`.

System.capabilities.hasAccessibility

Disponibilità

Flash Player 6.

Uso

`System.capabilities.hasAccessibility`

Descrizione

Proprietà; un valore booleano che indica se il dispositivo supporta la comunicazione tra Flash Player e gli strumenti di accesso. Il valore predefinito è `false`. La stringa per il server è `ACC`.

System.capabilities.hasAudio

Disponibilità

Flash Player 6.

Uso

System.capabilities.hasAudio

Descrizione

Proprietà; un valore booleano che indica se il lettore ha capacità audio. Il valore predefinito è true. La stringa per il server è A.

System.capabilities.hasMP3

Disponibilità

Flash Player 6.

Uso

System.capabilities.hasMP3

Descrizione

Proprietà; un valore booleano che indica se il lettore dispone di un decodificatore MP3. Il valore predefinito è true. La stringa per il server è MP3.

System.capabilities.language

Disponibilità

Flash Player 6.

Uso

System.capabilities.language

Descrizione

Proprietà; un codice ISO 639-1 costituito da due lettere minuscole e un'etichetta opzionale contenente il codice nazionale ISO 3166 costituito da due lettere maiuscole. Le lingue sono indicate da sigle basate sull'inglese. Ad esempio, "it-IT" è il codice della lingua in cui è redatta la presente documentazione. La stringa per il server è LAN. Flash supporta il seguente insieme secondario di etichette linguistiche:

Lingua	Etichetta	Etichette e paesi supportati
Inglese	en	Stati Uniti = US, Regno Unito = UK
Francese	fr	
Coreano	ko	
Giapponese	ja	
Svedese	sv	
Tedesco	de	
Spagnolo	es	
Italiano	it	
Cinese semplificato	zh	PRC (cinese semplificato) = CN

Lingua	Etichetta	Etichette e paesi supportati
Cinese classico	zh	Taiwan (cinese classico) = TW
Portoghese	pt	
Polacco	pl	
Ungherese	hu	
Ceco	cs	
Turco	tr	
Finlandese	fi	
Danese	da	
Norvegese	no	
Olandese	nl	
Russo	ru	
Altre/Sconosciute	xu	

System.capabilities.manufacturer

Disponibilità

Flash Player 6.

Uso

System.capabilities.manufacturer

Descrizione

Proprietà; una stringa che indica il produttore di Flash Player. L'indicazione predefinita è "Macromedia *nome sistema operativo*" (il *nome sistema operativo* potrebbe essere "Windows"/"Macintosh"/"Altro sistema operativo"). La stringa per il server è M.

System.capabilities.os

Disponibilità

Flash Player 6.

Uso

System.capabilities.os

Descrizione

Proprietà; una stringa che indica il produttore di Flash Player. La stringa predefinita è vuota (""). La proprietà os può restituire le stringhe seguenti: "Windows XP", "Windows 2000", "Windows NT", "Windows 98/ME", "Windows 95", "Windows CE" (disponibile solo in versione SDK, non nella versione per computer da tavolo) e "MacOS". La stringa per il server è OS.

System.capabilities.pixelAspectRatio

Disponibilità

Flash Player 6.

Uso

`System.capabilities.hasVideoEncoder`

Descrizione

Proprietà; un numero intero che indica le dimensioni dello schermo in pixel. Il valore predefinito è 1.0. La stringa per il server è PAR.

System.capabilities.screenColor

Disponibilità

Flash Player 6.

Uso

`System.capabilities.screenColor`

Descrizione

Proprietà; indica se lo schermo è a colori (`color`), in scala di grigi (`gray`) o in bianco e nero (`bw`). Il valore predefinito è `color`. La stringa per il server è SC.

System.capabilities.screenDPI

Disponibilità

Flash Player 6.

Uso

`System.capabilities.screenDPI`

Descrizione

Proprietà; indica la dimensione dpi (dots per inch) dello schermo, in pixel. Il valore predefinito è 72. La stringa per il server è DPI.

System.capabilities.screenResolution.x

Disponibilità

Flash Player 6.

Uso

`System.capabilities.screenResolution.x`

Descrizione

Proprietà; un numero intero che indica la risoluzione orizzontale massima dello schermo. Il valore predefinito è 800 (pixel). La stringa per il server è SRX.

System.capabilities.screenResolution.y

Disponibilità

Flash Player 6.

Uso

System.capabilities.screenResolution.y

Descrizione

Proprietà; un numero intero che indica la risoluzione verticale massima dello schermo. Il valore predefinito è 600 (pixel). La stringa per il server è SRY.

System.capabilities.version

Disponibilità

Flash Player 6.

Uso

System.capabilities.version

Descrizione

Proprietà; un numero intero che specifica la versione Flash Player supportata. Il valore predefinito è 6.0. La stringa per il server è VER.

System.capabilities.hasVideoEncoder

Disponibilità

Flash Player 6.

Uso

System.capabilities.hasVideoEncoder

Descrizione

Proprietà; una matrice di codificatori video. La stringa per il server è VE.

targetPath

Disponibilità

Flash Player 5.

Uso

```
targetpath(OggettoClipfilmato);
```

Parametri

OggettoClipfilmato Il riferimento (ad esempio, `_root` o `_parent`) al clip filmato del percorso target richiamato.

Valori restituiti

Nessuno.

Descrizione

Funzione; restituisce una stringa contenente il percorso target di *OggettoClipfilmato*. Il percorso target è restituito in notazione del punto. Per recuperare il percorso target in notazione della barra inclinata, usare la proprietà `_target`.

Esempio

Nell'esempio seguente, viene visualizzato il percorso target di un clip filmato in fase di caricamento.

```
onClipEvent(load){
    trace(targetPath(this));
}
```

Vedere anche

`eval`

tellTarget

Disponibilità

Flash Player 3. Funzione divenuta obsoleta in Flash 5; si raccomanda di usare la notazione del punto e l'azione `with`.

Uso

```
tellTarget("target") {
    istruzione/i;
}
```

Parametri

target Una stringa che specifica il percorso target della linea temporale da controllare.

istruzione/i Le istruzioni da eseguire se la condizione restituisce `true`.

Valori restituiti

Nessuno.

Descrizione

Azione; applica le istruzioni specificate nel parametro *istruzioni* alla linea temporale specificata nel parametro *target*. L'azione `tellTarget` è utile per controllare la navigazione. Assegnare `tellTarget` ai pulsanti per arrestare o avviare i clip filmato in altri punti dello stage. È inoltre possibile spostarsi in un particolare fotogramma all'interno di un clip filmato specifico. Ad esempio, è possibile assegnare `tellTarget` a dei pulsanti per interrompere o avviare la riproduzione di clip filmato sullo stage o per visualizzare un fotogramma particolare.

In Flash 5, è possibile usare la notazione del punto in sostituzione dell'azione `tellTarget`. È possibile usare l'azione `with` per inviare più azioni alla stessa linea temporale. L'azione `with` può essere usata per specificare qualsiasi oggetto; con l'azione `tellTarget` possono essere specificati solo i clip filmato.

Esempio

Questa istruzione `tellTarget` controlla l'istanza del clip filmato `ball` sulla linea temporale principale. Il fotogramma 1 dell'istanza `ball` è vuoto e ad esso è associata un'azione `stop`, che ne impedisce la visualizzazione sullo stage. Quando viene selezionato il pulsante con l'azione `tellTarget`, l'azione impartisce all'indicatore di registrazione in `ball` di andare al fotogramma 2 dove inizia l'animazione.

```
on(release) {
    tellTarget("ball") {
        gotoAndPlay(2);
    }
}
```

Nell'esempio seguente, per ottenere lo stesso risultato viene usata la notazione del punto.

```
on(release) {
    ball.gotoAndPlay(2);
}
```

Per inviare più comandi all'istanza `ball`, è possibile usare l'azione `with`, come mostrato nell'istruzione riportata di seguito.

```
on(release) {
    with(ball) {
        gotoAndPlay(2);
        _alpha = 15;
        _xscale = 50;
        _yscale = 50;
    }
}
```

Vedere anche

`with`

TextField (oggetto)

Tutti i campi di testo di input dinamici in un filmato Flash sono istanze dell'oggetto `TextField`. È possibile assegnare a un campo di testo un nome di istanza nella finestra di ispezione Proprietà per poterlo elaborare con ActionScript tramite i metodi e le proprietà dell'oggetto `TextField`. I nomi di istanza `TextField` vengono visualizzati in Esplora filmato e nella finestra di dialogo Inserisci percorso target del pannello Azioni.

L'oggetto `TextField` eredita proprietà o metodi dell'oggetto `Object`.

Per creare un campo di testo in modo dinamico, è possibile usare il metodo `MovieClip.createTextField`.

L'oggetto `TextField` è supportato da Flash Player 6 e versioni successive.

Riepilogo dei metodi validi per l'oggetto TextField

Metodo	Descrizione
<code>TextField.addListener</code>	Registra un oggetto in modo da ricevere una notifica quando si invocano i metodi <code>onChanged</code> e <code>onScroller</code> .
<code>TextField.getDepth</code>	Restituisce la profondità di un campo di testo.
<code>TextField.getNewTextFormat</code>	Fornisce il formato di testo predefinito assegnato a un testo appena inserito.
<code>TextField.removeListener</code>	Rimuove un oggetto Listener.
<code>TextField.removeTextField</code>	Rimuove un campo di testo creato con <code>MovieClip.createTextField</code> .
<code>TextField.setNewTextFormat</code>	Imposta un oggetto di formato testo per il testo inserito da un utente o tramite un metodo.
<code>TextField.replaceSel</code>	Sostituisce la selezione corrente.
<code>TextField.setTextFormat</code>	Imposta il formato di testo predefinito assegnato a un testo appena inserito.

Riepilogo delle proprietà valide per l'oggetto TextField

Proprietà	Descrizione
<code>TextField._alpha</code>	Il valore di trasparenza di un'istanza campo di testo.
<code>TextField.autoSize</code>	Controlla l'allineamento e il ridimensionamento automatici di un campo di testo.
<code>TextField.background</code>	Indica se il campo di testo ha uno sfondo.
<code>TextField.backgroundColor</code>	Indica il colore dello sfondo.
<code>TextField.border</code>	Indica se il campo di testo ha un bordo.
<code>TextField.borderColor</code>	Indica il colore del bordo.
<code>TextField.bottomScroll</code>	La linea visibile dell'estremità inferiore di un campo di testo.
<code>TextField.embedFonts</code>	Indica se il campo di testo usa profili di font incorporati o font del dispositivo.
<code>TextField._highQuality</code>	Indica la qualità di rendering del filmato.
<code>TextField._height</code>	L'altezza di un'istanza di campo di testo, in pixel. Questo valore influisce solo sul riquadro di limitazione del campo di testo e non sullo spessore del bordo o sulle dimensioni dei caratteri del testo.
<code>TextField.hscroll</code>	Indica il valore di scorrimento orizzontale di un campo di testo.
<code>TextField.html</code>	Indica la posizione corrente di scorrimento massimo di un campo di testo.
<code>TextField.htmlText</code>	Contiene la rappresentazione HTML del contenuto di un campo di testo.
<code>TextField.length</code>	Il numero di caratteri presenti in un campo di testo.
<code>TextField.maxChars</code>	Il numero massimo di caratteri che un campo di testo può contenere.
<code>TextField.maxhscroll</code>	Il valore massimo di <code>TextField.hscroll</code> .
<code>TextField.maxscroll</code>	Il valore massimo di <code>TextField.scroll</code> .
<code>TextField.multiline</code>	Indica se un campo di testo contiene più righe.
<code>TextField._name</code>	Il nome di istanza di un'istanza di campo di testo.
<code>TextField._parent</code>	Un riferimento all'istanza secondaria di un'altra istanza, di tipo <code>Button</code> o <code>MovieClip</code> .

Proprietà	Descrizione
<code>TextField.password</code>	Indica se un campo di testo nasconde i caratteri di input.
<code>TextField._quality</code>	Indica la qualità di rendering di un filmato.
<code>TextField.restrict</code>	L'insieme di caratteri che possono essere immessi in un campo di testo.
<code>TextField._rotation</code>	Il grado di rotazione di un'istanza di campo di testo.
<code>TextField.scroll</code>	Indica la posizione di scorrimento di un campo di testo.
<code>TextField.selectable</code>	Indica la possibilità di selezionare un campo di testo.
<code>TextField._soundbuftime</code>	Il tempo necessario affinché un suono venga caricato nel buffer prima che inizi la riproduzione.
<code>TextField.tabEnabled</code>	Indica se un clip filmato è incluso nell'ordinamento di tabulazione automatico.
<code>TextField.tabIndex</code>	Indica l'ordine di tabulazione di un oggetto.
<code>TextField.text</code>	Il testo attualmente presente in un campo di testo.
<code>TextField.textColor</code>	Il colore del testo presente nel campo di testo.
<code>TextField.textHeight</code>	L'altezza del riquadro di limitazione del campo di testo.
<code>TextField.textWidth</code>	La larghezza del riquadro di limitazione del campo di testo.
<code>TextField.type</code>	Indica se un campo di testo è un campo di testo di immissione o dinamico.
<code>TextField._url</code>	L'URL del file SWF che ha creato l'istanza del campo di testo.
<code>TextField.variable</code>	Il nome della variabile associata al campo di testo.
<code>TextField._visible</code>	Un valore booleano che determina se un'istanza di campo di testo è nascosta o visibile.
<code>TextField._width</code>	La larghezza di un'istanza di campo di testo, in pixel. Questo valore influisce solo sul riquadro di limitazione del campo di testo e non sullo spessore del bordo o sulle dimensioni dei caratteri del testo.
<code>TextField.wordWrap</code>	Indica se il campo di testo ha l'impostazione di a capo automatico.
<code>TextField._x</code>	La coordinata <i>x</i> di un'istanza di campo di testo.
<code>TextField._xmouse</code>	La coordinata <i>x</i> del cursore relativa a un'istanza di campo di testo.
<code>TextField._xscale</code>	Il valore che specifica la percentuale di modifica orizzontale in scala di un'istanza di campo di testo.
<code>TextField._y</code>	La coordinata <i>y</i> di un'istanza di campo di testo.
<code>TextField._ymouse</code>	La coordinata <i>y</i> del cursore relativa a un'istanza di campo di testo.
<code>TextField._yscale</code>	Il valore che specifica la percentuale di modifica verticale in scala di un'istanza di campo di testo.

Riepilogo dei gestori di evento validi per l'oggetto TextField

Metodo	Descrizione
<code>TextField.onChangeed</code>	Invocato quando il campo di testo viene modificato.
<code>TextField.onKillFocus</code>	Invocato quando il campo di testo non è più attivo.
<code>TextField.onScroller</code>	Invocato quando la proprietà <code>scroll</code> , <code>maxscroll</code> , <code>hscroll</code> , <code>maxhscroll</code> o <code>bottomscroll</code> di un campo di testo viene modificata.
<code>TextField.onSetFocus</code>	Invocato quando un campo di testo diventa attivo.

Riepilogo dell'oggetto TextField

Metodo	Descrizione
<code>TextField.onChangeed</code>	Notificato quando il campo di testo viene modificato.
<code>TextField.onScroller</code>	Notificato quando la proprietà <code>scroll</code> o <code>maxscroll</code> di un campo di testo viene modificata.

TextField._alpha

Disponibilità

Flash Player 6.

Uso

`TextField._alpha`

Descrizione

Proprietà; imposta o recupera la trasparenza alfa (*valore*) del campo di testo specificato da `TextField`. I valori validi vanno da 0 (completamente trasparente) a 100 (completamente opaco).

Esempio

Le istruzioni indicate di seguito impostano la proprietà `_alpha` di un campo di testo denominato `text1` al 30%.

```
on(release) {  
    text1._alpha = 30;  
}
```

TextField.addListener

Disponibilità

Flash Player 6.

Uso

`TextField.addListener(nuovoListener)`

Parametri

nuovoListener Un oggetto con eventi `onChangeed` e notifiche `onScroller`.

Valori restituiti

Nessuno.

Descrizione

Metodo; registra un oggetto per ricevere notifiche di eventi. Quando si verifica l'evento `onChanged` (oppure `onScroller`), vengono invocati gli eventi `TextField.onChanged` e `TextField.onScroller`, seguiti dai metodi `onChanged` e `onScroller` degli oggetti di notifica registrati con `addListener`. Più oggetti contemporaneamente possono ricevere notifiche di modifica. Se il listener *nuovoListener* è già registrato, non sono necessarie modifiche.

TextField.autoSize

Disponibilità

Flash Player 6.

Uso

TextField.autoSize

Descrizione

Proprietà; controlla automaticamente l'allineamento e il ridimensionamento automatici dei campi di testo. Se il valore di ridimensionamento automatico è "nessuno", il campo di testo si comporta normalmente e non viene allineato e ridimensionato automaticamente in base al testo. Se il valore è "sinistro", i bordi destro e inferiore del campo di testo si espandono o comprimono in base alle dimensioni del testo contenuto. I bordi sinistro e superiore restano nella stessa posizione. Se il ridimensionamento automatico è impostato su "centro", le dimensioni del campo di testo vengono modificate automaticamente e il centro orizzontale del campo di testo resta ancorato al centro della larghezza del campo di testo originale. Il lato inferiore continua ad espandersi per adattarsi al testo contenuto. Se il ridimensionamento automatico è impostato su "destra", le dimensioni del campo di testo si modificano automaticamente ma i lati sinistro e inferiore si espandono o comprimono. Il lato superiore e destro restano nella stessa posizione. Quando si imposta la proprietà `autoSize`, `true` corrisponde a "sinistra" e `false` a "nessuna".

Esempio

Nell'esempio seguente, la proprietà `autosize` del campo di testo `textField2` è impostata su "centro".

```
textField2.autosize = "centro";
```

TextField.background

Disponibilità

Flash Player 6.

Uso

TextField.background

Descrizione

Proprietà; se è impostato su `true`, il campo di testo ha uno sfondo. Se è impostato su `false`, il campo di testo non ha uno sfondo.

TextField.backgroundColor

Disponibilità

Flash Player 6.

Uso

`TextField.backgroundColor`

Descrizione

Proprietà; il colore dello sfondo del campo di testo. L'impostazione predefinita è `0xFFFFFFFF` (bianco). È possibile richiamare o impostare questa proprietà anche se attualmente non è visualizzato alcuno sfondo poiché esso diventa visibile solo se il campo di testo è delimitato da un bordo.

Vedere anche

`TextField.background`

TextField.border

Disponibilità

Flash Player 6.

Uso

`TextField.border`

Descrizione

Proprietà; se è impostata su `true`, il campo di testo è delimitato da un bordo. Se è impostata su `false`, il campo di testo non è delimitato da un bordo.

TextField.borderColor

Disponibilità

Flash Player 6.

Uso

`TextField.borderColor`

Descrizione

Proprietà; il colore del bordo del campo di testo, l'impostazione predefinita è `0x000000` (nero). È possibile richiamare o impostare questa proprietà, anche se non è attualmente visualizzato alcun bordo.

Vedere anche

`TextField.border`

TextField.bottomScroll

Disponibilità

Flash Player 6.

Uso

`TextField.bottomScroll`

Descrizione

Proprietà (sola lettura); un numero intero (con l'indice che inizia da 1) che indica l'ultima riga attualmente visibile in `TextField`. Il campo di testo è come "una finestra" attorno a un blocco di testo. La proprietà `TextField.scroll` è l'indice a partire da 1 della prima riga nella finestra.

Tutto il testo compreso tra le righe `TextField.scroll` e `TextField.bottomScroll` è attualmente visibile nel campo di testo.

TextField.embedFonts

Disponibilità

Flash Player 6.

Uso

`TextField.embedFonts`

Descrizione

Proprietà; un valore booleano che, se è impostato su `true`, visualizza il campo di testo usando i profili di font incorporati. Se impostato su `false`, visualizza il campo di testo usando font del dispositivo.

TextField._focusrect

Disponibilità

Flash Player 6.

Uso

`TextField._focusrect`

Descrizione

Proprietà; un valore booleano che specifica se un campo di testo attivo è circondato da un rettangolo giallo.

TextField.getDepth

Disponibilità

Flash Player 6.

Uso

`TextField.getDepth`

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce la profondità di un campo di testo.

TextField.getFontList

Disponibilità

Flash Player 6.

Uso

`TextField.getFontList`

Parametri

Nessuno.

Valori restituiti

Una matrice.

Descrizione

Metodo; restituisce un oggetto Array costituito dai nomi di tutti i font di Flash Player residenti nel sistema, inclusi i font del file SWF e tutti i file SWF delle risorse caricate. I nomi sono di tipo stringa.

TextField.getNewTextFormat

Disponibilità

Flash Player 6.

Uso

```
TextField.getNewTextFormat()
```

Parametri

Nessuno.

Valori restituiti

Un oggetto `TextFormat`.

Descrizione

Metodo; restituisce un oggetto `TextFormat` contenente una copia dell'oggetto di formato testo del campo di testo. L'oggetto di formato testo indica il formato del testo appena inserito, ad esempio il testo immesso tramite il metodo `replaceSel` o da un utente. Quando si invoca `getNewTextFormat`, vengono definite tutte le proprietà dell'oggetto `TextFormat` restituito. Nessuna proprietà equivale a `null`.

TextField.getTextFormat

Disponibilità

Flash Player 6.

Uso

```
TextField.getTextFormat()
```

```
TextField.getTextFormat (indice)
```

```
TextField.getTextFormat (beginIndex, endIndex)
```

Parametri

indice Un numero intero che specifica un carattere in una stringa.

Valori restituiti

Un oggetto.

Descrizione

Metodo; (Uso 1) restituisce un oggetto `TextFormat` contenente le informazioni di formattazione per tutto il testo immesso in un campo di testo. Solo le proprietà comuni a tutto il testo presente nel campo di testo vengono impostate nell'oggetto `TextFormat` risultante. Il valore di qualsiasi proprietà *mixed*, cioè con valori differenti nei diversi punti del testo, è impostato su `null`.

Uso 2: restituisce un oggetto `TextFormat` contenente una copia del formato di testo del campo di testo per *index*.

Uso 3: restituisce un oggetto `TextFormat` contenente le informazioni di formattazione per l'estensione del testo da *beginIndex* a *endIndex*.

Vedere anche

`TextField.getNewTextFormat`, `TextField.setNewTextFormat`, `TextField.setTextFormat`

TextField._height

Disponibilità

Flash Player 6.

Uso

`TextField._height`

Descrizione

Proprietà; imposta e recupera l'altezza del campo di testo, in pixel.

Esempio

Il seguente esempio di codice imposta l'altezza e la larghezza di un campo di testo.

```
myTextField._width = 200;  
myTextField._height = 200;
```

TextField._highquality

Disponibilità

Flash Player 6.

Uso

`TextField._highquality`

Descrizione

Proprietà (globale); specifica il livello di antialiasing applicato al filmato corrente. Specificare 2 (Ottima) per applicare un livello di alta qualità con smussatura delle bitmap sempre attiva. Specificare 1 (Alta qualità) per applicare l'antialiasing; in tal modo le bitmap verranno smussate se il filmato non contiene animazioni. Specificare 0 (Bassa qualità) per impedire l'antialiasing.

Esempio

```
_highquality = 1;
```

Vedere anche

`_quality`, `toggleHighQuality`

TextField.hscroll

Disponibilità

Flash Player 6.

Uso

`TextField.hscroll`

Valori restituiti

Un numero integrale.

Descrizione

Proprietà; indica la posizione corrente di scorrimento orizzontale. Se la proprietà `hscroll` è 0, il testo non viene scorso orizzontalmente.

Esempio

Nell'esempio seguente, il testo viene scorso orizzontalmente.

```
on (release) {  
    myTextField.hscroll += 1;  
}
```

Vedere anche

`TextField.maxhscroll`, `TextField.scroll`

TextField.html

Disponibilità

Flash Player 6.

Uso

`TextField.html`

Descrizione

Proprietà; flag che indica se il campo di testo contiene una rappresentazione HTML. Se la proprietà `html` risulta `true`, il campo di testo è in formato HTML. Se invece `html` risulta `false`, il campo di testo non è in formato HTML.

Vedere anche

`TextField.htmlText`

TextField.htmlText

Disponibilità

Flash Player 6.

Uso

`TextField.htmlText`

Descrizione

Proprietà; se il campo di testo è in formato HTML, questa proprietà contiene la rappresentazione HTML del contenuto. Se invece il campo di testo non è in formato HTML, essa si comporta esattamente come la proprietà `text`. È possibile indicare che il campo di testo è in formato HTML nella finestra di ispezione Proprietà oppure impostando la proprietà `html` del campo di testo su `true`.

Esempio

Nell'esempio seguente, il testo nel campo `text2` risulta in grassetto.

```
text2.html = true;  
text2.htmlText = "<b> testo in grassetto </b>";
```

TextField.length

Disponibilità

Flash Player 6.

Uso

TextField.length

Descrizione

Proprietà (sola lettura); indica il numero di caratteri in un campo di testo. Questa proprietà restituisce lo stesso valore di *text.length*, ma è più veloce. Il tasto Tab (“\t”) corrisponde a un carattere.

TextField.maxChars

Disponibilità

Flash Player 6.

Uso

TextField.maxChars

Descrizione

Proprietà; indica il numero massimo di caratteri che un campo di testo può contenere. Uno script può inserire testi di lunghezza maggiore di quella consentita da *maxChars*; la proprietà *maxChars* indica solo la lunghezza massima del testo inseribile dall'utente. Se il valore di questa proprietà è *null*, non vi sono limitazioni al testo che l'utente può inserire.

TextField.maxhscroll

Disponibilità

Flash Player 6.

Uso

TextField.maxhscroll

Descrizione

Proprietà (sola lettura); indica il valore massimo di *TextField.hscroll*.

Vedere anche

TextField.hscroll

TextField.maxscroll

Disponibilità

Flash Player 6.

Uso

maxField.maxscroll

Descrizione

Proprietà (sola lettura); indica il valore massimo di *TextField.scroll*.

Vedere anche

TextField.scroll

TextField.multiline

Disponibilità

Flash Player 6.

Uso

TextField.multiline

Descrizione

Proprietà; indica se il campo di testo è multiriga. Se il valore di questa proprietà è `true`, il campo di testo è multiriga; se invece è `false`, il campo di testo è a riga singola.

TextField._name

Disponibilità

Flash Player 6.

Uso

TextField._name

Descrizione

Proprietà; restituisce il nome dell'istanza del campo di testo specificato mediante *TextField*.

TextField.onChanged

Disponibilità

Flash Player 6.

Uso

TextField.onChanged

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando il contenuto di un campo di testo viene modificato. Per impostazione predefinita, non è definito ma può essere definito in uno script.

TextField.onKillFocus

Disponibilità

Flash Player 6.

Uso

```
TextField.onKillFocus = function (nuovoAttivo) {  
  istruzioni;  
};
```

Parametri

nuovoAttivo L'oggetto che viene attivato.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; evento invocato quando un campo di testo viene disattivato dalla tastiera. Il metodo `onKillFocus` riceve il parametro *nuovoAttivo*, che corrisponde a un oggetto rappresentante il nuovo oggetto attivato. Se nessun oggetto viene attivato, il valore di *nuovoAttivo* è `null`.

TextField.onScroller

Disponibilità

Flash Player 6.

Uso

`TextField.onScroller`

Descrizione

Gestore di evento; evento invocato quando le proprietà di scorrimento di un campo di testo vengono modificate.

Vedere anche

`TextField.hscroll`, `TextField.maxhscroll`, `TextField.maxscroll`, `TextField.scroll`

TextField.onSetFocus

Disponibilità

Flash Player 6.

Uso

```
TextField.onSetFocus = function(vecchioAttivo){  
  istruzioni;  
};
```

Parametri

vecchioAttivo L'oggetto da disattivare.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; evento invocato quando un campo di testo viene attivato dalla tastiera. Il parametro *vecchioAttivo* si riferisce all'oggetto attivato precedentemente da tastiera. Ad esempio, se l'utente preme il tasto di tabulazione per spostare il punto attivo da un pulsante a un campo di testo, *vecchioAttivo* contiene l'istanza del campo di testo.

Se precedentemente non è stato attivato alcun oggetto, *vecchioAttivo* contiene un valore `null`.

TextField._parent

Disponibilità

Flash Player 6.

Uso

`_parent.property`
`_parent._parent.property`

Descrizione

Proprietà; specifica o restituisce un riferimento al clip filmato contenente l'oggetto o il clip filmato corrente. L'oggetto corrente è quello contenente il codice ActionScript che fa riferimento a `_parent`. Usare `_parent` per specificare il percorso relativo a clip filmato od oggetti gerarchicamente superiori all'oggetto o al clip filmato attuale.

Vedere anche

`_root`, `targetPath`

TextField.password

Disponibilità

Flash Player 6.

Uso

`TextField.password`

Descrizione

Proprietà; se il valore di `password` risulta `true`, il campo di testo è protetto da password e i caratteri di input sono nascosti. Se invece il valore è `false`, il campo di testo non è protetto da password.

TextField._quality

Disponibilità

Flash Player 6.

Uso

`TextField._quality`

Descrizione

Proprietà (globale); imposta o recupera la qualità di rendering usata per un filmato. È sempre disponibile una copia dei caratteri dispositivo che non vengono alterati dalla proprietà `_quality`.

È possibile impostare la proprietà `_quality` con i valori riportati di seguito:

- "LOW" Bassa qualità di rendering. I grafici non sono sottoposti ad antialiasing e le bitmap non vengono smussate.
- "MEDIUM" Media qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 2x2 pixel, ma le bitmap non vengono smussate. Questo valore è adatto a filmati che non contengono testo.

- "HIGH" Alta qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 pixel; le bitmap vengono smussate se il filmato è statico. Questa è l'impostazione predefinita per la qualità di rendering usata da Flash.
- "BEST" Ottima qualità di rendering. I grafici sono sottoposti ad antialiasing con una griglia di 4x4 pixel; le bitmap vengono sempre smussate.

Esempio

Nell'esempio seguente, la qualità di rendering è impostata su LOW:

```
textfield._quality = "LOW";
```

Vedere anche

`_highquality`, `toggleHighQuality`

TextField.removeListener

Disponibilità

Flash Player 6.

Uso

```
Selection.removeListener(listener)
```

Parametri

listener L'oggetto che non riceve più notifiche di attivazione.

Valori restituiti

Se il *listener* è stato rimosso correttamente, il metodo restituisce un valore `true`. Se il *listener* non è stato rimosso correttamente poiché, ad esempio, non era incluso nell'elenco *listener* dell'oggetto `TextField`, il metodo restituisce `false`.

Descrizione

Metodo; rimuove un oggetto listener precedentemente registrato a un'istanza di campo di testo mediante `addListener`.

TextField.removeTextField

Disponibilità

Flash Player 6.

Uso

```
TextField.removeTextField()
```

Descrizione

Metodo; rimuove il campo di testo specificato da *TextField*. Questa operazione può essere eseguita soltanto su un campo di testo creato mediante il metodo `createTextField` dell'oggetto `MovieClip`. Essa non funziona se eseguita su campi di testo posizionati sulla linea temporale. Quando si chiama questo metodo, il campo di testo rimuove se stesso automaticamente. Questo metodo è simile al metodo `MovieClip.removeMovieClip`.

Vedere anche

`MovieClip.createTextField`

TextField.replaceSel

Disponibilità

Flash Player 6.

Uso

```
TextField.replaceSel(testo)
```

Parametri

testo Una stringa.

Valori restituiti

Nessuno.

Descrizione

Metodo; sostituisce la selezione corrente con il contenuto del parametro *testo*. Il testo viene inserito nella selezione corrente con il formato carattere e il formato paragrafo predefiniti correnti. Il testo non viene considerato come HTML, anche se si tratta di un campo di testo HTML.

È possibile usare il metodo `replaceSel` per inserire e cancellare del testo senza modificare la formattazione del carattere e del paragrafo del resto del testo.

TextField.restrict

Disponibilità

Flash Player 6.

Uso

```
TextField.restrict
```

Descrizione

Proprietà; indica il set di caratteri che l'utente può inserire in un campo di testo. Se il valore della proprietà `restrict` è `null`, è possibile inserire qualsiasi carattere. Se il valore della proprietà `restrict` è una stringa vuota, non è possibile inserire alcun carattere. Se il valore della proprietà `restrict` è una stringa di caratteri, è possibile inserire nel campo di testo soltanto i caratteri presenti nella stringa. La scansione della stringa viene eseguita da sinistra a destra. È possibile specificare un intervallo di caratteri usando il trattino (-). Questa proprietà pone delle restrizioni soltanto alle interazioni dell'utente; uno script può inserire testo in un campo senza limitazioni. Inoltre, tale proprietà non è sincronizzabile alle caselle di controllo Incorpora contorni carattere nella finestra di ispezione Proprietà.

Se la stringa comincia con `^`, vengono accettati tutti i caratteri ad eccezione di quelli a seguire nella stringa. Se la stringa non comincia con `^`, non viene accettato alcun carattere ad eccezione di quelli indicati nella stringa.

Esempio

Nell'esempio seguente, possono essere inseriti in un campo di testo solo caratteri maiuscoli, spazi e numeri:

```
my_txt.restrict = "A-Z 0-9";
```

Nell'esempio seguente, vengono accettati tutti i caratteri ad eccezione delle lettere minuscole:

```
my_txt.restrict = "^a-z";
```

È possibile usare una barra rovesciata per inserire un simbolo `^` o `-`. Le sequenze con barre rovesciate accettate sono `\-`, `\^` o `\\`. La barra rovesciata deve comparire nella stringa; quando specificato in ActionScript, si deve usare una doppia barra rovesciata. Ad esempio, il codice seguente include solo il trattino (`-`) e l'accento circonflesso (`^`):

```
my_txt.restrict = "\\-\\^";
```

L'accento circonflesso `^` può essere usato dovunque all'interno della stringa per includere o escludere determinati caratteri. Il codice seguente include solo lettere maiuscole, a eccezione della lettera maiuscola `Q`:

```
my_txt.restrict = "A-Z^Q";
```

È possibile usare la sequenza di escape `\\u` per generare stringhe con la proprietà `restrict`. Il seguente codice include solo caratteri corrispondenti ai codici compresi tra ASCII 32 (spazio) e ASCII 126 (tilde).

```
my_txt.restrict = "\\u0020-\\u007E";
```

TextField._rotation

Disponibilità

Flash Player 6.

Uso

TextField._rotation

Descrizione

Proprietà; specifica la rotazione del campo di testo in gradi.

TextField.scroll

Disponibilità

Flash Player 6.

Uso

TextField.scroll

Descrizione

Proprietà; definisce la posizione verticale del testo in un campo. La proprietà `scroll` è utile per indirizzare gli utenti verso un paragrafo specifico incluso in lunghi passaggi o per creare campi di testo scorrevoli. È possibile recuperare il valore di questa proprietà e modificarla.

Esempio

Il codice seguente è associato al pulsante `Su` che fa scorrere il campo di testo `myText`.

```
on (release) {  
    myText.scroll = myText.scroll + 1;  
}
```

Vedere anche

`TextField.maxscroll`, `TextField.scroll`

TextField.selectable

Disponibilità

Flash Player 6.

Uso

TextField.selectable

Descrizione

Proprietà; valore booleano che indica se il campo di testo è selezionabile. Il valore `true` indica che il testo è selezionabile.

TextField.setNewTextFormat

Disponibilità

Flash Player 6.

Uso

TextField.setNewTextFormat(formatoTesto)

Parametri

formatoTesto Istanza di un oggetto `TextFormat`.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un oggetto `TextFormat` per il testo appena inserito mediante il metodo `replaceSel` o direttamente dall'utente in un campo di testo. Ciascun campo di testo ha un nuovo formato di testo che viene assegnato quando si inserisce il nuovo testo.

Il formato del testo è specificato in una nuova istanza dell'oggetto `TextFormat` che contiene le informazioni relative sia alla formattazione del carattere che a quella del paragrafo. Le informazioni sulla formattazione del carattere contengono dati sull'aspetto dei singoli caratteri: ad esempio, nome, dimensione in punti, colore e URL a essi associato. Le informazioni sulla formattazione del paragrafo contengono informazioni sull'aspetto dei paragrafi: ad esempio, margine sinistro, margine destro, rientro della prima riga e allineamento a sinistra, a destra o al centro.

Vedere anche

`TextField.getNewTextFormat`, `TextField.getTextFormat`, `TextField.setTextFormat`

TextField.setTextFormat

Disponibilità

Flash Player 6.

Uso

TextField.setTextFormat(formatoTesto)

TextField.setTextFormat(indice, formatoTesto)

TextField.setTextFormat(inizioIndice, fineIndice, formatoTesto)

Parametri

inizioIndice Un numero intero.

fineIndice Numero intero che specifica il primo carattere dopo l'estensione di testo desiderata.

formatoTesto Istanza di un oggetto `TextFormat`. Un oggetto `TextFormat` contiene le informazioni relative alla formattazione del carattere e del paragrafo.

Valori restituiti

Nessuno.

Descrizione

Metodo; specifica un oggetto `TextFormat` per un dato intervallo di testo in un campo di testo. È possibile assegnare un formato testo a ciascun carattere nel campo di testo. Il formato testo del primo carattere di un paragrafo viene esaminato per eseguire la formattazione dell'intero paragrafo. Il metodo `setTextFormat` modifica il formato testo assegnato a singoli caratteri, gruppi di caratteri o all'intero testo presente in un campo.

Il formato del testo è specificato in una nuova istanza dell'oggetto `TextFormat` che contiene le informazioni relative sia alla formattazione del carattere che a quella del paragrafo. Le informazioni sulla formattazione del carattere contengono dati sull'aspetto dei singoli caratteri: ad esempio, nome, dimensione in punti, colore e URL a essi associato. Le informazioni sulla formattazione del paragrafo contengono informazioni sull'aspetto dei paragrafi: ad esempio, margine sinistro, margine destro, rientro della prima riga e allineamento a sinistra a destra o al centro.

Uso 1: applica le proprietà di *formatoTesto* a tutto il testo presente in un campo di testo.

Uso 2: applica le proprietà di *formatoTesto* al carattere in posizione *indice*.

Uso 3: applica le proprietà del parametro *formatoTesto* all'intervallo di testo compreso tra i parametri *inizioIndice* e *fineIndice*.

Esempio

Nell'esempio seguente, viene creato un nuovo oggetto `TextFormat` denominato `myTextFormat` la cui proprietà `bold` viene impostata su `true`. Viene poi chiamato il metodo `setTextFormat` e applicato il nuovo formato testo al campo di testo `myTextField`.

```
myTextFormat = new TextFormat();
myTextFormat.bold = true;
myTextField.setTextFormat(myTextFormat);
```

Vedere anche

`TextFormat` (oggetto)

TextField._soundbuftime

Disponibilità

Flash Player 6.

Uso

`TextField._soundbuftime`

Descrizione

Proprietà (globale); numero intero che specifica in secondi il tempo di prebuffer di un suono prima che abbia inizio lo streaming.

TextField.tabEnabled

Disponibilità

Flash Player 6.

Uso

TextField.tabEnabled

Descrizione

Proprietà; è possibile impostarla su un'istanza degli oggetti MovieClip, Button o TextField. Per impostazione predefinita, questa proprietà è di tipo "undefined".

Se la proprietà `tabEnabled` risulta `undefined` o il suo valore è `true`, l'oggetto viene incluso nell'ordinamento di tabulazione automatico; se anche il valore della proprietà `tabIndex` è specificato, l'oggetto viene incluso nell'ordinamento personalizzato. Se `tabEnabled` è `false`, l'oggetto non viene incluso nell'ordinamento di tabulazione automatico. Nel caso di un clip filmato, se `tabEnabled` è `false`, gli elementi secondari del clip filmato possono essere ancora inclusi nell'ordinamento di tabulazione automatico, finché anche la proprietà `tabChildren` è impostata su `false`.

Se `tabEnabled` è `undefined` o `true`, l'oggetto viene incluso nell'ordinamento di tabulazione personalizzato se viene impostata anche la proprietà `tabIndex`. Se `tabEnabled` è `false`, l'oggetto non viene incluso nell'ordinamento di tabulazione personalizzato, anche se viene impostata la proprietà `tabIndex`. Se `tabEnabled` è impostata su `false` in un clip filmato, gli elementi secondari del filmato possono ancora essere inclusi nell'ordinamento di tabulazione automatico.

TextField.tabIndex

Disponibilità

Flash Player 6.

Uso

TextField.tabIndex

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Proprietà; consente di personalizzare l'ordinamento di tabulazione degli oggetti in un filmato. È possibile impostare la proprietà `tabIndex` su un pulsante, clip filmato o istanza del campo di testo; per impostazione predefinita, tale proprietà è `undefined`.

Se qualsiasi oggetto attualmente visualizzato nel filmato Flash contiene una proprietà `tabIndex`, l'ordinamento di tabulazione automatico viene disattivato e l'ordinamento di tabulazione è calcolato in base alle proprietà `tabIndex` degli oggetti del filmato. L'ordinamento di tabulazione personalizzato riguarda solo gli oggetti che hanno proprietà `tabIndex`.

La proprietà `tabIndex` deve essere un numero intero positivo. Gli oggetti vengono ordinati in base alle relative proprietà `tabIndex` in ordine ascendente. Un oggetto `tabIndex 1` precede un oggetto `tabIndex 2`. Se i due oggetti hanno lo stesso valore `tabIndex`, il primo nell'ordinamento di tabulazione è `undefined`.

L'ordinamento di tabulazione personalizzato definito dalla proprietà `tabIndex` è flat. Questo significa che non vengono considerate le relazioni gerarchiche degli oggetti nel filmato Flash. Tutti gli oggetti nel filmato Flash con proprietà `tabIndex` vengono inseriti nell'ordine di tabulazione, stabilito in base all'ordine dei valori `tabIndex`. Se due oggetti hanno lo stesso valore `tabIndex`, il primo è `undefined`. Non usare lo stesso valore `tabIndex` per più oggetti.

TextField._target

Disponibilità

Flash Player 6.

Uso

`TextField._target`

Descrizione

Proprietà (sola lettura); restituisce il percorso target dell'istanza del campo di testo specificata nel parametro `TextField`.

TextField.text

Disponibilità

Flash Player 6.

Uso

`TextField.text`

Descrizione

Proprietà; indica il testo attualmente presente nel campo di testo. Le righe sono separate mediante il carattere di ritorno a capo (`\r`, ASCII 13). Questa proprietà contiene il testo normale del campo di testo, non formattato e senza tag HTML, anche se il campo di testo è in formato HTML.

Vedere anche

`TextField.htmlText`

TextField.textColor

Disponibilità

Flash Player 6.

Uso

`TextField.textColor`

Descrizione

Proprietà; indica il colore del testo contenuto in un campo di testo.

TextField.textHeight

Disponibilità

Flash Player 6.

Uso

TextField.textHeight

Descrizione

Proprietà; indica l'altezza del testo.

TextField.textWidth

Disponibilità

Flash Player 6.

Uso

TextField.textWidth

Descrizione

Proprietà; indica la larghezza del testo.

TextField.type

Disponibilità

Flash Player 6.

Uso

TextField.type

Descrizione

Proprietà; specifica il tipo di campo di testo. Questa proprietà può avere due valori: "dynamic" specifica un campo di testo dinamico (non modificabile dall'utente) e "input" un campo di testo di input.

Esempio

```
TextField.type = "dynamic";
```

TextField._url

Disponibilità

Flash Player 6.

Uso

TextField._url

Descrizione

Proprietà (sola lettura); recupera l'URL del file SWF che ha creato il campo di testo.

TextField.variable

Disponibilità

Flash Player 6.

Uso

TextField.variable

Descrizione

Proprietà; il nome della variabile a cui è associato il campo di testo. Questa proprietà è di tipo String.

TextField._visible

Disponibilità

Flash Player 6.

Uso

TextField._visible

Descrizione

Proprietà; valore booleano che indica se il campo di testo specificato dal parametro *TextField* è visibile. I campi di testo non visibili (proprietà *_visible* impostata su *false*) vengono disattivati.

TextField._width

Disponibilità

Flash Player 6.

Uso

TextField._width

Descrizione

Proprietà; imposta e recupera la larghezza di un campo di testo, in pixel.

Esempio

Nell'esempio seguente, vengono impostate le proprietà di altezza e ampiezza di un campo di testo.

```
myTextField._width=200;  
myTextField._height=200;
```

Vedere anche

MovieClip._height

TextField.wordWrap

Disponibilità

Flash Player 6.

Uso

TextField.wordWrap

Descrizione

Proprietà; valore booleano che indica se il campo di testo ha il ritorno a capo automatico. Se il valore di `wordWrap` è `true`, il campo di testo ha il ritorno a capo automatico; se invece il valore è `false`, il campo di testo non ha il ritorno a capo automatico.

TextField._x

Disponibilità

Flash Player 6.

Uso

TextField._x

Descrizione

Proprietà; numero intero che imposta la coordinata x di un campo di testo rispetto alle coordinate locali del clip filmato principale. Se un campo di testo è nella linea temporale principale, il sistema di coordinate corrispondente considera l'angolo superiore sinistro dello stage come (0, 0). Se il campo di testo è all'interno di un clip filmato a cui sono state applicate trasformazioni, il campo di testo rientra nel sistema di coordinate locali del clip che lo contiene. Nel caso di un clip filmato ruotato di 90° in senso antiorario, il campo di testo incluso eredita un sistema di coordinate con la stessa rotazione. Le coordinate del campo di testo si riferiscono alla posizione del punto di registrazione.

Vedere anche

TextField._xscale, *TextField._y*, *TextField._yscale*

TextField._xmouse

Disponibilità

Flash Player 6.

Uso

TextField._xmouse

Descrizione

Proprietà (sola lettura); restituisce la coordinata x della posizione del mouse relativa al campo di testo.

Vedere anche

TextField._ymouse

TextField._xscale

Disponibilità

Flash Player 6.

Uso

TextField._xscale

Descrizione

Proprietà; determina la scala orizzontale (*percentuale*) del campo di testo applicata dal punto di registrazione del campo di testo. Il punto di registrazione predefinito è (0,0).

Vedere anche

TextField._x, *TextField._y*, *TextField._yscale*

TextField._y

Disponibilità

Flash Player 6.

Uso

TextField._y

Descrizione

Proprietà; imposta la coordinata *x* di un campo di testo rispetto alle coordinate locali del clip filmato principale. Se un campo di testo è nella linea temporale principale, il sistema di coordinate corrispondente considera l'angolo superiore sinistro dello stage come (0, 0). Se il campo di testo è all'interno di un altro clip filmato a cui sono state applicate trasformazioni, il campo di testo rientra nel sistema di coordinate locali del clip che lo contiene. Nel caso di un clip filmato ruotato di 90° in senso antiorario, il campo di testo incluso eredita un sistema di coordinate con la stessa rotazione. Le coordinate del campo di testo si riferiscono alla posizione del punto di registrazione.

Vedere anche

TextField._x, *TextField._xscale*, *TextField._yscale*

TextField._ymouse

Disponibilità

Flash Player 6.

Uso

TextField._ymouse

Descrizione

Proprietà (sola lettura); indica la coordinata *y* della posizione del mouse relativa al campo di testo.

Vedere anche

TextField._xmouse

TextField._yscale

Disponibilità

Flash Player 6.

Uso

`TextField._yscale`

Descrizione

Proprietà; determina la scala verticale (*percentuale*) del campo di testo applicata dal punto di registrazione del campo di testo. Il punto di registrazione predefinito è (0,0).

Vedere anche

`TextField._x`, `TextField._xscale`, `TextField._y`

TextFormat (oggetto)

L'oggetto `TextFormat` rappresenta le informazioni sulla formattazione del carattere.

Prima di chiamare i metodi dell'oggetto `TextFormat`, è necessario crearne un'istanza usando la funzione di costruzione `new TextFormat`.

È possibile impostare i parametri `TextFormat` su `null` per indicare che non sono definiti.

Quando si applica un oggetto `TextFormat` a un campo di testo usando il metodo `setTextFormat`, vengono applicate solo le sue proprietà definite come nell'esempio seguente:

```
myTextFormat = new TextFormat();
myTextFormat.bold = true;
myTextField.setTextFormat(myTextFormat);
```

Questo codice crea prima un oggetto `TextFormat` vuoto con tutte le proprietà non definite, quindi imposta la proprietà `bold` su un determinato valore.

Il codice `myTextField.setTextFormat(myTextFormat)` modifica soltanto la proprietà `bold` del formato testo predefinito del campo di testo poiché la proprietà `bold` è l'unica definita in `myTextFormat`. Tutti gli altri aspetti del formato testo predefinito rimangono invariati.

Quando `getTextFormat` viene invocato, un oggetto `TextFormat` viene restituito con tutte le proprietà definite; pertanto, nessun valore risulta `null`.

Riepilogo dei metodi validi per l'oggetto TextFormat

Metodo	Descrizione
<code>TextFormat.getTextExtent</code>	Restituisce un oggetto con due proprietà, <code>width</code> e <code>height</code> , che indicano le dimensioni del testo in un campo di testo.

Riepilogo delle proprietà valide per l'oggetto `TextFormat`

Proprietà	Descrizione
<code>TextFormat.align</code>	Indica l'allineamento di un paragrafo.
<code>TextFormat.blockIndent</code>	Indica il rientro del testo, in punti.
<code>TextFormat.bold</code>	Indica se il testo è in grassetto.
<code>TextFormat.bullet</code>	Indica se il testo è in un elenco puntato o meno.
<code>TextFormat.color</code>	Indica il colore del testo.
<code>TextFormat.font</code>	Indica il nome del font del testo in un dato formato.
<code>TextFormat.indent</code>	Indica il rientro dal margine sinistro del primo carattere del paragrafo.
<code>TextFormat.italic</code>	Indica se il testo è in corsivo.
<code>TextFormat.leading</code>	Indica il valore dell'interlinea.
<code>TextFormat.leftMargin</code>	Indica il margine sinistro del paragrafo, in punti.
<code>TextFormat.rightMargin</code>	Indica il margine destro del paragrafo, in punti.
<code>TextFormat.tabStops</code>	Specifica gli spazi di tabulazione personalizzati.
<code>TextFormat.target</code>	Indica la finestra di un browser in cui viene visualizzato un ipertesto.
<code>TextFormat.size</code>	Indica la dimensione del testo, in punti.
<code>TextFormat.underline</code>	Indica se il testo è sottolineato.
<code>TextFormat.url</code>	Indica l'URL a cui rinvia un collegamento nel testo.

Funzione di costruzione per l'oggetto `TextFormat`

Uso

```
new TextFormat([font, [dimensione, [colore, [grassetto, [corsivo,  
[sottolineatura, [url, [target, [allineamento, [margineSinistro,  
[margineDestro, [rientro, [interlinea]]]]]]]]]]])
```

Parametri

font Stringa che indica il nome di un font per il testo.

dimensione Numero intero che indica la dimensione del testo, in punti.

colore Colore del testo che usa questo formato. Un numero contenente le tre componenti RGB a 8 bit; ad esempio, 0xFF0000 corrisponde al rosso, 0x00FF00 al verde.

grassetto Valore booleano che indica se il testo è in grassetto.

corsivo Valore booleano che indica se il testo è in corsivo.

sottolineatura Valore booleano che indica se il testo è sottolineato.

url URL a cui rinvia un collegamento ipertestuale nel testo. Se *url* è una stringa vuota, il testo non ha collegamenti ipertestuali.

target Finestra target in cui viene visualizzato un ipertesto. Se la finestra target è una stringa vuota, il testo viene visualizzato nella finestra target predefinita `_self`. Se la proprietà `TextFormat.url` è impostata come stringa vuota o sul valore `null`, essa può essere di tipo `get` o `set` ma non ha alcun effetto.

allineamento Allineamento del paragrafo, rappresentato come stringa. Se la stringa è "left", il paragrafo viene allineato a sinistra. Se è "center", il paragrafo viene centrato verticalmente. Se è "right", il paragrafo viene allineato a destra.

marginSinistro Indica il margine sinistro del paragrafo, in punti.

marginDestro Indica il margine destro del paragrafo, in punti.

rientro Numero intero che indica il rientro dal margine sinistro del primo carattere del paragrafo.

interlinea Numero che indica il valore dell'interlinea.

Descrizione

Funzione di costruzione; crea un'istanza dell'oggetto `TextFormat` con le proprietà specificate. È quindi possibile modificare le proprietà dell'oggetto `TextFormat` per modificare la formattazione dei campi di testo.

Ogni parametro può essere impostato sul valore `null` per indicare che non è definito. Tutti i parametri sono opzionali; ogni parametro viene trattato come `null`.

Disponibilità

Flash Player 6.

TextFormat.align

Disponibilità

Flash Player 6.

Uso

`TextFormat.align`

Descrizione

Proprietà; indica l'allineamento del paragrafo, rappresentato come una stringa. Allineamento del paragrafo, rappresentato come stringa. Se la stringa è "left", il paragrafo viene allineato a sinistra. Se è "center", il paragrafo viene centrato verticalmente. Se è "right", il paragrafo viene allineato a destra. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.blockIndent

Disponibilità

Flash Player 6.

Uso

`TextFormat.blockIndent`

Descrizione

Proprietà; numero che indica il rientro del testo, in punti. Il rientro blocco viene applicato a un intero blocco di testo e, quindi, a tutte le righe del testo. In compenso, il rientro normale (`TextFormat.indent`) viene applicato soltanto alla prima riga di ciascun paragrafo. Se la proprietà risulta `null`, l'oggetto `TextFormat` non specifica un rientro blocco.

TextFormat.bold

Disponibilità

Flash Player 6.

Uso

TextFormat.bold

Descrizione

Proprietà; valore booleano che indica se il testo è in grassetto. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.bullet

Disponibilità

Flash Player 6.

Uso

TextFormat.bullet

Descrizione

Proprietà; valore booleano che indica se il testo fa parte di un elenco puntato. In un elenco puntato, ogni paragrafo del testo viene rientrato e a sinistra della prima riga di ciascun paragrafo viene visualizzato un simbolo di elenco puntato. Se questa proprietà risulta `null`, l'oggetto `TextFormat` non specifica se il testo fa parte di un elenco puntato o meno.

TextFormat.color

Disponibilità

Flash Player 6.

Uso

TextFormat.color

Descrizione

Proprietà; indica il colore del testo. Un numero contenente le tre componenti RGB a 8 bit; ad esempio, `0xFF0000` corrisponde al rosso, `0x00FF00` al verde.

TextFormat.font

Disponibilità

Flash Player 6.

Uso

TextFormat.font

Descrizione

Proprietà; stringa che indica il nome del font in questo formato di testo. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.getTextExtent

Disponibilità

Flash Player 6.

Uso

`TextFormat.getTextExtent (testo)`

Parametri

testo Una stringa.

Valori restituiti

Un oggetto con le proprietà `width` e `height`.

Descrizione

Metodo; restituisce la dimensione della stringa di testo specificata nel parametro *testo* in questo formato carattere. Il valore restituito è un oggetto della classe `Object` con due proprietà: `width` e `height`. Il parametro *testo* viene considerato come testo semplice (non HTML). Il parametro *testo* equivale a una singola riga di testo; il ritorno a capo manuale e l'avanzamento riga vengono ignorati e il ritorno a capo automatico non viene applicato.

TextFormat.indent

Disponibilità

Flash Player 6.

Uso

`TextFormat.indent`

Descrizione

Proprietà; numero intero che indica il rientro dal margine sinistro del primo carattere del paragrafo. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.italic

Disponibilità

Flash Player 6.

Uso

`TextFormat.italic`

Descrizione

Proprietà; valore booleano che indica se il testo in questo formato è in corsivo. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.leading

Disponibilità

Flash Player 6.

Uso

TextFormat.leading

Descrizione

Proprietà; indica il valore dell'interlinea. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.leftMargin

Disponibilità

Flash Player 6.

Uso

TextFormat.leftMargin

Descrizione

Proprietà; il margine sinistro del paragrafo, in punti. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.rightMargin

Disponibilità

Flash Player 6.

Uso

TextFormat.rightMargin

Descrizione

Proprietà; il margine destro del paragrafo, in punti. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.size

Disponibilità

Flash Player 6.

Uso

TextFormat.size

Descrizione

Proprietà; la dimensione del testo in questo formato, in punti. Il valore predefinito è `null` e indica che la proprietà non è definita.

TextFormat.tabStops

Disponibilità

Flash Player 6.

Uso

TextFormat.tabStops

Descrizione

Proprietà; matrice di numeri interi non negativi che specifica gli spazi di tabulazione personalizzati. Ciascuna tabulazione è specificata in punti. Se gli spazi di tabulazione personalizzati non vengono specificati (*null*), il loro valore predefinito è 4 (larghezza media del carattere).

TextFormat.target

Disponibilità

Flash Player 6.

Uso

TextFormat.target

Descrizione

Proprietà; indica la finestra target in cui viene visualizzato l'ipertesto. Se la finestra target è una stringa vuota, il testo viene visualizzato nella finestra target predefinita *_self*. Se la proprietà *TextFormat.url* è impostata come stringa vuota o sul valore *null*, essa può essere di tipo get o set ma non ha alcun effetto.

TextFormat.underline

Disponibilità

Flash Player 6.

Uso

TextFormat.underline

Descrizione

Proprietà; valore booleano che indica se il testo che usa questo formato è sottolineato. Ciò si verifica se *underline* è impostato su *true*. Se invece risulta *false*, il testo in questo formato non viene sottolineato. Si noti che la sottolineatura è la stessa eseguita mediante il tag `<U>`; quest'ultima però non è una vera sottolineatura poiché non salta correttamente i tratti discendenti. Il valore predefinito è *null* e indica che la proprietà non è definita.

TextFormat.url

Disponibilità

Flash Player 6.

Uso

TextFormat.url

Descrizione

Proprietà; indica l'URL a cui rinvia un collegamento ipertestuale nel testo. Se la proprietà `url` è una stringa vuota, nel testo non ci sono collegamenti ipertestuali. Il valore predefinito è `null` e indica che la proprietà non è definita.

this

Disponibilità

Flash Player 5.

Uso

`this`

Descrizione

Parola chiave; fa riferimento a un oggetto o a un'istanza di clip filmato. Quando viene eseguito uno script, `this` fa riferimento all'istanza di clip filmato contenente lo script. Quando viene chiamato un metodo, `this` contiene un riferimento all'oggetto contenente il metodo chiamato.

All'interno di un'azione gestore di evento `on` associata a un pulsante, `this` fa riferimento alla linea temporale che contiene tale pulsante. All'interno di un'azione gestore di evento `on` associata a un clip filmato, `this` fa riferimento alla linea temporale che contiene tale clip filmato.

Esempio

Nell'esempio seguente, la parola chiave `this` fa riferimento all'oggetto `Circle`.

```
function Circle(radius) {
    this.radius = radius;
    this.area = Math.PI * radius * radius;
}
```

Nell'istruzione seguente assegnata a un fotogramma, la parola chiave `this` fa riferimento al clip filmato corrente.

```
// Imposta la proprietà alfa del filmato corrente su 20.
this._alpha = 20;
```

Nell'istruzione seguente interna a un gestore `onClipEvent`, la parola chiave `this` fa riferimento al clip filmato corrente.

```
// Quando il clip filmato viene caricato, un'operazione startDrag
// viene avviata per il clip filmato corrente.
```

```
onClipEvent (load) {
    startDrag (this, true);
}
```

Vedere anche

`new`

toggleHighQuality

Disponibilità

Flash 2.

Uso

`toggleHighQuality()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; attiva/disattiva l'antialiasing in Flash Player. L'antialiasing smussa i bordi degli oggetti e rallenta la riproduzione del filmato. L'azione `toggleHighQuality` influenza tutti i filmati in Flash Player.

Esempio

Il codice seguente è applicabile a un pulsante che attiva/disattiva l'antialiasing quando viene premuto.

```
on(release) {  
    toggleHighQuality();  
}
```

Vedere anche

`_quality`, `_highquality`

trace

Disponibilità

Flash Player 4.

Uso

`trace(espressione)`

Parametri

espressione Un'espressione da valutare. Quando un file SWF viene aperto nello strumento di creazione del codice di Flash (mediante il comando Prova filmato), il valore del parametro *espressione* viene visualizzato nella finestra Output.

Valori restituiti

Nessuno.

Descrizione

Azione; valuta *espressione* e visualizza il risultato nella finestra Output in modalità di prova filmato.

Usare la funzione `trace` per registrare note di programmazione o per visualizzare messaggi nella finestra Output durante le prove del filmato. Usare il parametro *espressione* per verificare se una condizione esiste o per visualizzare valori nella finestra Output. L'azione `trace` è simile alla funzione `alert` in JavaScript.

È possibile usare il comando Ometti azioni Trace in Impostazioni pubblicazione per rimuovere le azioni `trace` dal file SWF esportato.

Esempio

Questo esempio è tratto da un gioco in cui l'istanza di clip filmato trascinabile `rabbi` deve essere rilasciata su un target specifico. Un'istruzione condizionale valuta la proprietà `_droptarget` ed esegue diverse azioni a seconda di dove il clip filmato `rabbi` viene rilasciato. L'azione `trace` è usata alla fine dello script per valutare la posizione del clip filmato `rabbi` e visualizzare il risultato nella finestra Output. Se `rabbi` non si comporta come previsto (ad esempio, se viene rilasciato nel posto sbagliato), i valori inviati alla finestra Output dall'azione `trace` possono contribuire a identificare il problema nello script.

```
on(press) {
    rabbi.startDrag();
}

on(release) {
    if(eval(_droptarget) != target) {
        rabbi._x = rabbi_x;
        rabbi._y = rabbi_y;
    } else {
        rabbi._x = rabbi_x;
        rabbi._y = rabbi_y;
        target = "_root.pasture";
    }
    trace("rabbi_y = " + rabbi_y);
    trace("rabbi_x = " + rabbi_x);
    stopDrag();
}
```

true

Disponibilità

Flash Player 5.

Uso

true

Descrizione

Valore booleano univoco che rappresenta l'opposto di `false`.

Vedere anche

false

typeof

Disponibilità

Flash Player 5.

Uso

`typeof espressione`

Parametri

espressione Stringa, clip filmato, oggetto o funzione.

Descrizione

Operatore; un operatore unario collocato prima di un parametro singolo. L'operatore `typeof` comporta la valutazione di *espressione* da parte dell'interprete di Flash; il risultato è una stringa che indica se l'espressione è una stringa, un clip filmato, un oggetto o una funzione. La tabella seguente illustra i risultati dell'operatore `typeof` su ciascun tipo di espressione:

Parametro	Risultato
Stringa	string
MovieClip	movieclip
Pulsante	object
Campo di testo	object
Numero	number
Valore booleano	boolean
Oggetto	object
Funzione	function

undefined

Disponibilità

Flash Player 5.

Uso

undefined

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Valore speciale abitualmente usato per indicare che a una variabile non è ancora stato assegnato alcun valore. Un riferimento a un valore non definito restituisce il valore speciale `undefined`. Il codice ActionScript `typeof(undefined)` restituisce la stringa `"undefined"`. L'unico tipo di valore `undefined` è `undefined`.

Quando `undefined` viene convertito in una stringa, questa risulta vuota.

Il valore `undefined` è simile al valore speciale `null`. In effetti, quando `null` e `undefined` vengono confrontati mediante l'operatore di uguaglianza numerica, risultano uguali.

Esempio

In questo esempio, la variabile `x` non è stata dichiarata e quindi ha valore `undefined`. Nella prima sezione di codice, l'operatore di uguaglianza numerica (`==`) confronta il valore di `x` con il valore `undefined` e il risultato appropriato viene visualizzato nella finestra **Output**. Nella seconda sezione di codice, l'operatore di uguaglianza numerica confronta i valori `null` e `undefined`.

```
// x non è stata dichiarata.
trace ("Il valore di x è " + x);
if (x == undefined) {
    trace ("x è undefined");
} else {
    trace ("x non è undefined");
}

trace ("typeof (x) è " + typeof (x));
if (null == undefined) {
    trace ("null e undefined sono uguali");
} else {
    trace ("null e undefined non sono uguali");
}
```

Nella finestra **Output** viene visualizzato il risultato seguente:

```
Il valore di x è x è undefined
typeof (x) è undefined
null e undefined sono uguali
```

Note: nella specifica ECMA-262, `undefined` viene convertito nella stringa `"undefined"` e non in una stringa vuota, a differenza di quanto accade in **ActionScript**.

unescape

Disponibilità

Flash Player 5.

Uso

```
unescape(x)
```

Parametri

`x` Una stringa con sequenze di escape esadecimali.

Valori restituiti

Nessuno.

Descrizione

Funzione di primo livello; valuta il parametro `x` come stringa, decodifica la stringa a partire da un formato con codifica URL (converte le sequenze esadecimali in caratteri ASCII) e restituisce la stringa.

Esempio

L'esempio seguente illustra la conversione di una stringa con sequenze di escape nella relativa stringa senza sequenze di escape.

```
escape("Ciao{[Mondo]}");
```

Il risultato che utilizza le sequenze di escape è il seguente:

```
("Ciao%7B%5Bmondo%5D%7D");
```

Per ritornare al formato originale, usare `unescape`:

```
unescape("Ciao%7B%5Bmondo%5D%7D")
```

Il risultato è il seguente:

```
Ciao{[mondo]}
```

unloadMovie

Disponibilità

Flash Player 3.

Uso

```
unloadMovie[Num](livello/"target")
```

Parametri

livello Il livello (`_levelN`) in cui è caricato un filmato. Quando si scarica un filmato o un'immagine in un livello, l'azione `unloadMovie` nel pannello Azioni in modalità normale passa a `unloadMovieNum`; in modalità esperto, è necessario specificare `unloadMovieNum` o selezionarla dalla casella degli strumenti Azioni.

target Il percorso target di un clip filmato.

Valori restituiti

Nessuno.

Descrizione

Azione; rimuove un filmato o un clip filmato caricato da Flash Player. Per scaricare un filmato caricato in un livello in Flash Player, usare il parametro *level*. Per scaricare un clip filmato caricato, usare il parametro *target*.

Esempio

Nell'esempio seguente, viene scaricato il clip filmato `draggable` sulla linea temporale principale, quindi viene caricato il filmato `movie.swf` nel livello 4.

```
on (press) {
    unloadMovie ("_root.draggable");
    loadMovieNum ("movie.swf", 4);
}
```

Nell'esempio seguente, viene scaricato il filmato caricato nel livello 4.

```
on (press) {
    unloadMovieNum (4);
}
```

Vedere anche

`loadMovie`, `loadMovieNum`, `unloadMovieNum`

unloadMovieNum

Disponibilità

Flash Player 3.

Uso

```
unloadMovieNum(livello)
```

Parametri

livello Il livello (`_levelN`) in cui è caricato un filmato.

Valori restituiti

Nessuno.

Descrizione

Azione; rimuove un filmato caricato da Flash Player.

Vedere anche

`loadMovie`, `loadMovieNum`

updateAfterEvent

Disponibilità

Flash Player 5.

Uso

```
updateAfterEvent()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Azione; aggiorna la visualizzazione (a prescindere dalla frequenza di fotogrammi al secondo impostata per il filmato) quando la si chiama all'interno del gestore `onClipEvent` o come parte di una funzione o metodo passati a `setInterval`. Flash ignora le chiamate a `updateAfterEvent` che non si trovano all'interno di un gestore `onClipEvent` o non sono parte di una funzione o metodo passati a `setInterval`.

Vedere anche

`onClipEvent`, `setInterval`

var

Disponibilità

Flash Player 5.

Uso

```
var nomeVariabile1 [= valore1] [...,nomeVariabileN [=valoreN]]
```

Parametri

nomeVariabile Un identificatore.

valore Il valore da assegnare alla variabile.

Valori restituiti

Nessuno.

Descrizione

Azione; usata per dichiarare variabili locali. Se si dichiarano variabili locali all'interno di una funzione, le variabili vengono definite per la funzione e non sono più valide quando si esce dalla funzione. Se le variabili non sono dichiarate all'interno di un blocco ({}) e l'elenco delle azioni è stato eseguito con un'azione `call`, le variabili sono locali e non sono più valide quando si esce dall'elenco corrente. Se le variabili non sono dichiarate all'interno di un blocco e l'elenco delle azioni corrente non è stato eseguito con un'azione `call`, le variabili non sono locali.

Esempio

Nell'esempio seguente, viene usata l'azione `var` per dichiarare e assegnare le variabili:

```
var x;  
var y = 1;  
var z = 3, w = 4;  
var s, t, u = z;
```

void

Disponibilità

Flash Player 5.

Uso

```
void (espressione)
```

Descrizione

Operatore; un operatore unario che ignora il valore *espressione* e restituisce un valore non definito. L'operatore `void` viene spesso usato in operazioni di confronto con l'operatore `==` per verificare se un valore non è definito.

while

Disponibilità

Flash Player 4.

Uso

```
while(condizione) {  
    istruzione/i;  
}
```

Parametri

condizione L'espressione valutata a ogni esecuzione dell'azione `while`. Se il valore restituito risulta `true`, vengono eseguite le istruzioni *istruzione/i*.

istruzione/i Il codice da eseguire se la condizione restituisce il valore `true`.

Valori restituiti

Nessuno.

Descrizione

Azione; verifica un'istruzione o una serie di istruzioni ripetutamente in un ciclo fino a quando l'espressione restituisce il valore `true`.

Prima che il blocco di istruzioni venga eseguito, la *condizione* viene verificata; se il valore restituito è `true`, le istruzioni vengono eseguite. Se invece la condizione risulta `false`, il blocco di istruzioni viene saltato e viene eseguito il primo blocco di istruzioni dopo l'azione `while`.

I cicli sono generalmente usati per eseguire un'azione finché una variabile contatore è minore di un valore specificato. Al termine di ogni ciclo il contatore viene incrementato fino al raggiungimento del valore specificato. A quel punto, la *condizione* non risulta più `true` e il ciclo si conclude.

L'istruzione `while` esegue le operazioni seguenti. Ogni ripetizione dei passaggi da 1 a 4 viene denominata *iterazione* del ciclo. La *condizione* viene verificata all'inizio di ciascuna iterazione, come nell'esempio seguente.

- 1 La *condizione* dell'espressione viene valutata.
- 2 Se il valore restituito dalla *condizione* è `true` oppure è un valore convertibile nel valore booleano `true`, ad esempio un numero diverso da zero, passare al punto 3.
In caso contrario, l'istruzione `while` viene completata e la sua esecuzione rimanda all'istruzione immediatamente successiva al ciclo `while`.
- 3 Eseguire il blocco di istruzioni *istruzione/i*.
- 4 Passare al punto 1.

Esempio

Questo esempio duplica cinque clip filmato sullo stage, ognuno con posizioni x e y , `xscale` e `yscale` e proprietà `_alpha` generati a caso per realizzare un effetto sparpagliato. La variabile `foo` è inizializzata con il valore 0. Il parametro *condizione* è impostato in modo che il ciclo `while` venga eseguito cinque volte, oppure un numero di volte equivalente al valore della variabile `foo` se questa è minore di 5. All'interno del ciclo `while`, un clip filmato viene duplicato e `setProperty` usato per modificare le diverse proprietà del clip filmato duplicato. L'ultima istruzione del ciclo incrementa `foo` in modo che quando il valore raggiunge 5, il parametro *condizione* restituisce il valore `false` e il ciclo non viene più eseguito.

```
on(release) {
    foo = 0;
    while(foo < 5) {
        duplicateMovieClip("_root.flower", "mc" + foo, foo);
        setProperty("mc" + foo, _x, random(275));
        setProperty("mc" + foo, _y, random(275));
        setProperty("mc" + foo, _alpha, random(275));
        setProperty("mc" + foo, _xscale, random(200));
        setProperty("mc" + foo, _yscale, random(200));
        foo++;
    }
}
```

Vedere anche

do while, continue, for, for..in

with

Disponibilità

Flash Player 5.

Uso

```
with (oggetto) {
    istruzione/i;
}
```

Parametri

oggetto Istanza di un oggetto ActionScript o di un clip filmato.

istruzione/i Un'azione o un gruppo di azioni racchiuso tra parentesi graffe.

Valori restituiti

Nessuno.

Descrizione

Azione; consente di specificare un oggetto (ad esempio, un clip filmato) con il parametro *oggetto* e di valutare le espressioni e le azioni all'interno di tale oggetto con il parametro *istruzione/i*. In questo modo si evita di dovere continuamente scrivere il nome dell'oggetto o il suo percorso di identificazione.

Il parametro *oggetto* diviene il contesto in cui vengono lette le proprietà, le variabili e le funzioni presenti nel parametro *istruzione/i*. Ad esempio, se *oggetto* è `myArray` e due delle proprietà specificate sono `length` e `concat`, queste proprietà vengono automaticamente lette come `myArray.length` e `myArray.concat`. Come altro esempio, se *oggetto* è `state.california`, è come se qualsiasi azione o istruzione interna all'azione `with` fosse richiamata dall'interno dell'istanza `california`.

Per trovare il valore di un identificatore nel parametro *istruzione/i*, ActionScript comincia all'inizio della catena dell'area di validità specificata da *oggetto* e ricerca l'identificatore a ogni livello della catena dell'area di validità seguendo un ordine specifico.

La catena dell'area di validità usata dall'azione *with* per risolvere gli identificatori inizia con la prima voce dell'elenco e prosegue fino all'ultima, come segue:

- Oggetto specificato nel parametro *oggetto* nell'azione *with* più interna.
- Oggetto specificato nel parametro *oggetto* nell'azione *with* più esterna.
- Oggetto di attivazione. Un oggetto temporaneo automaticamente creato quando viene attivata una funzione contenente le variabili locali richiamate nella funzione
- Clip filmato contenente lo script in esecuzione.
- Oggetto globale: oggetti incorporati come Math o String.

Per impostare una variabile dentro un'azione *with*, la variabile deve essere stata dichiarata all'esterno dell'azione *with* oppure è necessario definire il percorso completo della linea temporale sulla quale risiede la variabile. Se si imposta una variabile in un'azione *with* senza dichiararla, l'azione *with* ricercherà il valore seguendo la catena dell'area di validità. Se la variabile non esiste ancora, il nuovo valore verrà impostato sulla linea temporale da cui l'azione *with* è stata richiamata.

In Flash 5 l'azione *with* sostituisce l'azione *tellTarget* diventata obsoleta. Si consiglia di usare *with* invece di *tellTarget* perché si tratta di un'estensione di ActionScript conforme allo standard ECMA-262. La differenza principale tra le azioni *with* e *tellTarget* consiste nel fatto che *with* richiede come parametro il riferimento a un clip filmato o a un altro oggetto, mentre *tellTarget* richiede una stringa di percorso target che identifica il clip filmato e il suo parametro e non può agire su oggetti.

Esempio

L'esempio seguente imposta le proprietà *x* e *y* dell'istanza *someOtherMovieClip*, quindi ordina a *someOtherMovieClip* di passare al fotogramma 3 e di arrestarsi in tale posizione.

```
with (someOtherMovieClip) {  
    _x = 50;  
    _y = 100;  
    gotoAndStop(3);  
}
```

Il frammento di codice seguente rappresenta un'alternativa a quello precedente, senza l'uso dell'azione *with*.

```
someOtherMovieClip._x = 50;  
someOtherMovieClip._y = 100;  
someOtherMovieClip.gotoAndStop(3);
```

È altresì possibile scrivere questo codice usando l'azione *tellTarget*. Tuttavia, qualora *someOtherMovieClip* non fosse un clip filmato bensì un oggetto, non sarà possibile usare l'azione *with*.

```
tellTarget ("someOtherMovieClip") {  
    _x = 50;  
    _y = 100;  
    gotoAndStop(3);  
}
```

L'azione `with` è utile per accedere contemporaneamente a più elementi all'interno di una catena di area di validità. Nell'esempio seguente, l'oggetto incorporato `Math` è posto all'inizio della catena di area di validità. L'impostazione di `Math` come oggetto predefinito risolve gli identificatori `cos`, `sin` e `PI` rispettivamente a `Math.cos`, `Math.sin` e `Math.PI`. Gli identificatori `a`, `x`, `y` e `r` non sono metodi o proprietà dell'oggetto `Math`, ma poiché esistono nell'area di validità dell'oggetto di attivazione per la funzione `polar` vengono risolti come le variabili locali corrispondenti.

```
function polar(r) {
    var a, x, y;
    with (Math) {
        a = PI * r * r;
        x = r * cos(PI);
        y = r * sin(PI/2);
    }
    trace("area = " + a);
    trace("x = " + x);
    trace("y = " + y);
}
```

È possibile usare azioni `with` annidate per accedere a informazioni in più aree di validità. Nell'esempio seguente, l'istanza `fresno` e l'istanza `salinas` sono istanze secondarie dell'istanza `california`. L'istruzione imposta i valori `_alpha` di `fresno` e `salinas` senza modificare il valore `_alpha` di `california`.

```
with (california){
    with (fresno){
        _alpha = 20;
    }
    with (salinas){
        _alpha = 40;
    }
}
```

Vedere anche

`tellTarget`

XML (oggetto)

I metodi e le proprietà dell'oggetto XML consentono di caricare, analizzare sintatticamente, inviare, costruire e manipolare alberi di documenti XML. In Flash MX, l'oggetto XML è diventato un oggetto nativo. Questa modifica ha consentito di migliorarne notevolmente le prestazioni.

Prima di chiamare i metodi dell'oggetto XML è necessario crearne un'istanza usando la funzione di costruzione `new XML()`.

XML è supportato in Flash Player 5 e Flash Player 6.

Riepilogo dei metodi validi per l'oggetto XML

Metodo	Descrizione
<code>XML.appendChild</code>	Aggiunge un nodo alla fine dell'elenco dei nodi secondari dell'oggetto specificato.
<code>XML.cloneNode</code>	Duplica il nodo specificato e, se richiesto, duplica ricorsivamente tutti i nodi secondari.
<code>XML.createElement</code>	Crea un nuovo elemento XML.
<code>XML.createTextNode</code>	Crea un nuovo nodo di testo XML.
<code>XML.getBytesLoaded</code>	Restituisce il numero di byte caricati per il documento XML specificato.
<code>XML.getBytesTotal</code>	Restituisce la dimensione del documento XML, in byte.
<code>XML.hasChildNodes</code>	Restituisce <code>true</code> se il nodo specificato possiede nodi secondari; altrimenti restituisce <code>false</code> .
<code>XML.insertBefore</code>	Inserisce un nodo davanti a uno esistente nell'elenco dei nodi secondari del nodo specificato.
<code>XML.load</code>	Carica un documento (specificato dall'oggetto XML) da un URL.
<code>XML.parseXML</code>	Analizza sintatticamente un documento XML nell'albero dell'oggetto XML specificato.
<code>XML.removeNode</code>	Rimuove il nodo specificato dal relativo nodo principale.
<code>XML.send</code>	Invia l'oggetto XML specificato a un URL.
<code>XML.sendAndLoad</code>	Invia l'oggetto XML specificato a un URL e carica la risposta del server in un altro oggetto XML.
<code>XML.toString</code>	Converte il nodo specificato e ogni nodo secondario in testo XML.

Riepilogo delle proprietà valide per l'oggetto XML

Proprietà	Descrizione
<code>XML.contentType</code>	Indica il tipo MIME trasmesso al server.
<code>XML.docTypeDecl</code>	Imposta e restituisce informazioni sulla dichiarazione <code>DOCTYPE</code> di un documento XML.
<code>XML.firstChild</code>	Fa riferimento al primo nodo secondario nell'elenco per il nodo specificato.
<code>XML.ignoreWhite</code>	Se impostata su <code>true</code> , i nodi di testo che contengono solo spazi bianchi vengono eliminati durante il processo di analisi sintattica.
<code>XML.lastChild</code>	Fa riferimento all'ultimo nodo secondario nell'elenco per il nodo specificato.
<code>XML.load</code>	Verifica se l'oggetto XML specificato è stato caricato.
<code>XML.nextSibling</code>	Fa riferimento al nodo successivo nell'elenco dei nodi secondari del nodo principale.
<code>XML.nodeName</code>	Restituisce il nome del tag di un elemento XML.
<code>XML.nodeType</code>	Restituisce il tipo del nodo specificato (elemento XML o nodo di testo).
<code>XML.nodeValue</code>	Restituisce il testo del nodo specificato se il nodo è un nodo di testo.
<code>XML.parentNode</code>	Fa riferimento al nodo principale del nodo specificato.
<code>XML.previousSibling</code>	Fa riferimento al nodo precedente nell'elenco dei nodi secondari del nodo principale.
<code>XML.status</code>	Restituisce un codice numerico di stato che indica l'esito dell'operazione di analisi sintattica di un documento XML.
<code>XML.xmlDecl</code>	Imposta e restituisce informazioni sulla dichiarazione del documento di un documento XML.

Riepilogo delle collezioni valide per l'oggetto XML

Metodo	Descrizione
<code>XML.attributes</code>	Restituisce una matrice associativa contenente tutti gli attributi del nodo specificato.
<code>XML.childNodes</code>	Restituisce una matrice contenente riferimenti ai nodi secondari del nodo specificato.

Riepilogo dei gestori di evento validi per l'oggetto XML

Metodo	Descrizione
<code>XML.onData</code>	Funzione di callback invocata quando viene completato lo scaricamento di testo XML dal server o quando, durante lo scaricamento, si verifica un errore.
<code>XML.onLoad</code>	Funzione di callback per <code>load</code> e <code>sendAndLoad</code> .

Funzione di costruzione dell'oggetto XML

Disponibilità

Flash Player 5.

Uso

```
new XML([origine])
```

Parametri

origine Il testo XML analizzato sintatticamente per creare il nuovo oggetto XML.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un nuovo oggetto XML. Prima di chiamare i metodi dell'oggetto XML è necessario crearne un'istanza usando il metodo di costruzione `new XML()`.

Note: i metodi `createElement` e `createTextNode` sono i metodi di 'costruzione' per creare gli elementi e i nodi di testo in un albero di documento XML.

Esempio

Uso 1: nell'esempio seguente, viene creato un nuovo oggetto XML vuoto.

```
myXML = new XML();
```

Uso 2: nell'esempio seguente, viene creato un oggetto XML mediante l'analisi sintattica del testo XML specificato nel parametro *origine*; gli elementi dell'albero XML risultante vengono quindi inseriti nel nuovo oggetto XML.

```
anyOtherXML = new XML("<state>California<city>san francisco</city></state>");
```

Vedere anche

`XML.createElement`, `XML.createTextNode`

XML.appendChild

Disponibilità

Flash Player 5.

Uso

myXML.appendChild(nodoSecondario)

Parametri

nodoSecondario Il nodo secondario da aggiungere all'elenco dei nodi secondari dell'oggetto XML specificato.

Valori restituiti

Nessuno.

Descrizione

Metodo; aggiunge il nodo secondario specificato all'elenco dei nodi secondari dell'oggetto XML. Il nodo secondario aggiunto viene collocato nella struttura ad albero una volta rimosso dal relativo nodo principale eventualmente esistente.

Esempio

L'esempio seguente duplica l'ultimo nodo di *doc1* e lo aggiunge a *doc2*.

```
doc1 = new XML(src1);  
doc2 = new XML();  
node = doc1.lastChild.cloneNode(true);  
doc2.appendChild(node);
```

XML.attributes

Disponibilità

Flash Player 5.

Uso

myXML.attributes

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Collezione (lettura-scrittura); restituisce una matrice associativa contenente tutti gli attributi dell'oggetto XML specificato.

Esempio

L'esempio seguente visualizza i nomi degli attributi XML nella finestra Output.

```
str = "<mytag name=\"Valore\"> intem </mytag>";  
doc = new XML(str);  
y = doc.firstChild.attributes.name;  
    trace (y);  
doc.firstChild.attributes.order = "primo";  
z = doc.firstChild.attributes.order  
    trace (z);
```

I valori seguenti vengono visualizzati nella finestra Output:

```
Valore  
primo
```

XML.childNodes

Disponibilità

Flash Player 5.

Uso

myXML.childNodes

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Collezione (sola lettura); restituisce una matrice dei nodi secondari dell'oggetto XML specificato. Ogni elemento nella matrice è un riferimento a un oggetto XML rappresentante un nodo secondario. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

Questa collezione non è definita per i nodi di testo (`nodeType == 3`).

Vedere anche

`XML.nodeType`

XML.cloneNode

Disponibilità

Flash Player 5.

Uso

myXML.cloneNode(profondità)

Parametri

profondità Valore booleano che specifica se i nodi secondari dell'oggetto XML specificato vengono ricorsivamente duplicati.

Valori restituiti

Nessuno.

Descrizione

Metodo; costruisce e restituisce un nuovo nodo XML con tipo, nome, valore e attributi uguali a quelli dell'oggetto XML specificato. Se *profondità* è impostato su *true*, tutti i nodi secondari vengono ricorsivamente duplicati; il risultato è una copia esatta dell'albero di documento dell'oggetto originale.

Il nodo originale del nodo duplicato restituito non è più associato all'albero dell'elemento duplicato. Ne consegue che *nextSibling*, *parentNode* e *previousSibling* hanno tutti valore *null*. Se non viene effettuata la copia di un clip filmato, anche *firstChild* e *lastChild* risultano *null*.

XML.contentType

Disponibilità

Flash Player 6.

Uso

myXML.contentType

Descrizione

Proprietà; il tipo MIME inviato al server quando si chiama il metodo *XML.send* o *XML.sendAndLoad*. Il tipo predefinito è *application/x-www-urlform encoded*.

Vedere anche

XML.send, *XML.sendAndLoad*

XML.createElement

Disponibilità

Flash Player 5.

Uso

myXML.createElement(nome)

Parametri

nome Il nome del tag dell'elemento XML creato.

Valori restituiti

Nessuno.

Descrizione

Metodo; crea un nuovo elemento XML usando il nome specificato nel parametro. In un primo tempo, il nuovo elemento non ha nodi né principali, né secondari, né allo stesso livello. Il metodo restituisce un riferimento all'oggetto XML appena creato che rappresenta l'elemento. Questo metodo e *createTextNode* sono i metodi di costruzione per la creazione di nodi per un oggetto XML.

XML.createTextNode

Disponibilità

Flash Player 5.

Uso

```
myXML.createTextNode(testo)
```

Parametri

testo Il testo usato per creare il nuovo nodo di testo.

Valori restituiti

Nessuno.

Descrizione

Metodo; crea un nuovo nodo di testo XML con il testo specificato. In un primo tempo il nuovo nodo non ha nodi principali e, in quanto nodo di testo, non ha nodi né secondari, né allo stesso livello. Questo metodo restituisce un riferimento all'oggetto XML che rappresenta il nuovo nodo di testo. Questo metodo e `createElement` sono i metodi di costruzione per la creazione di nodi per un oggetto XML.

XML.docTypeDecl

Disponibilità

Flash Player 5.

Uso

```
my.XML.XMLdocTypeDecl
```

Descrizione

Proprietà; imposta e restituisce informazioni sulla dichiarazione DOCTYPE di un documento XML. Dopo la trasformazione del testo XML in oggetto XML tramite l'analisi sintattica, la proprietà `XML.docTypeDecl` dell'oggetto XML viene impostata sul testo della dichiarazione DOCTYPE del documento XML. Ad esempio, `<!DOCTYPE greeting SYSTEM "hello.dtd">`. Questa proprietà viene impostata usando il formato stringa della dichiarazione DOCTYPE, non un oggetto nodo XML.

La funzione di analisi sintattica XML di ActionScript non è una funzione di convalida. La dichiarazione DOCTYPE viene letta dalla funzione di analisi sintattica e memorizzata nella proprietà `docTypeDecl`, ma non viene effettuata alcuna convalida in base al DTD.

Se l'operazione di analisi sintattica non incontra alcuna dichiarazione DOCTYPE, `XML.docTypeDecl` viene impostato su un valore non definito. `XML.toString` genera il contenuto di `XML.docTypeDecl` immediatamente dopo la memorizzazione della dichiarazione XML in `XML.xmlDecl` e prima di qualsiasi altro testo nell'oggetto XML. Se `XML.docTypeDecl` non è definito, non viene generata alcuna dichiarazione DOCTYPE.

Esempio

Nell'esempio seguente, viene usato `XML.docTypeDecl` per impostare la dichiarazione DOCTYPE per un oggetto XML:

```
OggettoXML.docTypeDecl = "<!DOCTYPE greeting SYSTEM \"hello.dtd\">";
```

Vedere anche

`XML.toString`, `XML.xmlDecl`

XML.firstChild

Disponibilità

Flash Player 5.

Uso

```
myXML.firstChild
```

Descrizione

Proprietà (sola lettura); valuta l'oggetto XML specificato e fa riferimento al primo nodo secondario nell'elenco dei nodi secondari del nodo principale. Questa proprietà è `null` se il nodo non ha nodi secondari. Questa proprietà non è definita se il nodo è un nodo di testo. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

Vedere anche

`XML.appendChild`, `XML.insertBefore`, `XML.removeNode`

XML.getBytesLoaded

Disponibilità

Flash Player 6.

Uso

```
XML.getBytesLoaded()
```

Parametri

Nessuno.

Valori restituiti

Un numero intero che indica il numero dei byte caricati.

Descrizione

Metodo; restituisce il numero di byte caricati (di cui si è effettuato lo streaming) per il documento XML. Il confronto tra il valore restituito da `getBytesLoaded` e il valore restituito da `getBytesTotal` fornisce la percentuale di un documento XML caricata.

Vedere anche

`XML.getBytesTotal`

XML.getBytesTotal

Disponibilità

Flash Player 6.

Uso

```
XML.getBytesTotal()
```

Parametri

Nessuno.

Valori restituiti

Un numero integrale.

Descrizione

Metodo; restituisce la dimensione, in byte, del documento XML.

Vedere anche

`XML.getBytesLoaded`

XML.hasChildNodes

Disponibilità

Flash Player 5.

Uso

```
myXML.hasChildNodes()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; restituisce `true` se il nodo specificato possiede nodi secondari; altrimenti restituisce `false`.

Esempio

Nell'esempio seguente, vengono usate le informazioni dall'oggetto XML in una funzione definita dall'utente.

```
if (rootNode.hasChildNodes()) {  
    myfunc (rootNode.firstChild);  
}
```

XML.ignoreWhite

Disponibilità

Flash Player 5.

Uso

```
myXML.ignoreWhite = booleano
```

```
XML.prototype.ignoreWhite = booleano
```

Parametri

booleano Valore booleano `true` o `false`.

Descrizione

Proprietà; l'impostazione predefinita è `false`. Se impostata su `true`, i nodi di testo che contengono solo spazi bianchi vengono eliminati durante il processo di analisi sintattica. I nodi di testo con interlinea o spazi bianchi finali rimangono invariati.

Uso 1: è possibile impostare la proprietà `ignoreWhite` per i singoli oggetti XML, come nel codice seguente:

```
myXML.ignoreWhite = true
```

XML.insertBefore

Disponibilità

Flash Player 5.

Uso

```
myXML.insertBefore(nodoSecondario, nodoPrecedente)
```

Parametri

nodoSecondario Il nodo da inserire.

nodoPrecedente Il nodo prima del punto di inserzione per *nodoSecondario*.

Valori restituiti

Nessuno.

Descrizione

Metodo; inserisce un nuovo nodo secondario nell'elenco dei nodi secondari dell'oggetto XML, prima di *nodoPrecedente*. Se il parametro *nodoPrecedente* non è definito o è `null`, il nodo viene aggiunto usando `appendChild`. Se *nodoPrecedente* non è un nodo secondario di *myXML*, l'inserimento non può essere eseguito.

XML.lastChild

Disponibilità

Flash Player 5.

Uso

```
myXML.lastChild
```

Descrizione

Proprietà (sola lettura); valuta l'oggetto XML e fa riferimento all'ultimo nodo secondario nell'elenco dei nodi secondari del nodo principale. Questo metodo restituisce `null` se il nodo non ha nodi secondari. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

Vedere anche

`XML.appendChild`, `XML.insertBefore`, `XML.removeNode`

XML.load

Disponibilità

Flash Player 5.

Uso

```
myXML.load(url)
```

Parametri

url L'URL in cui si trova il documento XML da caricare. Tale URL deve trovarsi nello stesso sottodominio dell'URL che contiene il filmato.

Valori restituiti

Nessuno.

Descrizione

Metodo; carica un documento XML dall'URL specificato e sostituisce il contenuto dell'oggetto XML specificato con i dati XML scaricati. Il processo di caricamento è asincrono; non termina subito dopo l'esecuzione del metodo `load`. Quando viene eseguita l'azione `load`, la proprietà `loaded` dell'oggetto XML viene impostata su `false`. Al termine dello scaricamento dei dati XML, la proprietà `loaded` viene impostata su `true` e viene chiamato il metodo `onLoad`. I dati XML non vengono analizzati sintatticamente fino alla fine dello scaricamento. Gli eventuali alberi XML contenuti nell'oggetto XML vengono rimossi.

È possibile specificare le proprie funzioni callback al posto del metodo `onLoad`.

Esempio

L'esempio seguente illustra un semplice uso di `XML.load`.

```
doc = new XML();  
doc.load ("theFile.xml");
```

Vedere anche

`XML.loaded`, `XML.onLoad`

XML.loaded

Disponibilità

Flash Player 5.

Uso

myXML.loaded

Descrizione

Proprietà (sola lettura); indica se il processo di caricamento del documento avviato dall'attivazione di `XML.load` è completato. Se il processo è completato, il metodo restituisce `true`; altrimenti restituisce `false`.

Esempio

L'esempio seguente illustra l'uso di `XML.loaded` in un semplice script.

```
if (doc.loaded) {  
    gotoAndPlay(4);  
}
```

XML.nextSibling

Disponibilità

Flash Player 5.

Uso

myXML.nextSibling

Descrizione

Proprietà (sola lettura); valuta l'oggetto XML e fa riferimento al nodo successivo nell'elenco dei nodi secondari del nodo principale. Questo metodo restituisce `null` se il nodo non ha un nodo successivo allo stesso livello. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

Vedere anche

`XML.appendChild`, `XML.insertBefore`, `XML.removeNode`

XML.nodeName

Disponibilità

Flash Player 5.

Uso

`myXML.nodeName`

Descrizione

Proprietà; estrae o restituisce il nome del nodo dell'oggetto XML. Se l'oggetto XML è un elemento XML (`nodeType == 1`), `nodeName` è il nome del tag rappresentante il nodo nel file XML. Ad esempio, `TITLE` è il `nodeName` di un tag `HTML TITLE`. Se l'oggetto XML è un nodo di testo (`nodeType == 3`), `nodeName` è `null`.

Vedere anche

`XML.nodeType`

XML.nodeType

Disponibilità

Flash Player 5.

Uso

`myXML.nodeType`

Descrizione

Proprietà (sola lettura); estrae o restituisce un valore `nodeType`, dove 1 rappresenta un elemento XML e 3 un nodo di testo.

Vedere anche

`XML.nodeValue`

XML.nodeValue

Disponibilità

Flash Player 5.

Uso

`myXML.nodeValue`

Descrizione

Proprietà; restituisce il valore del nodo dell'oggetto XML. Se l'oggetto XML è un nodo di testo, `nodeType` è 3, `nodeValue` è il testo del nodo. Se l'oggetto XML è un elemento XML (tipo di nodo 1), il valore di `nodeValue` è `null` ed è di sola lettura.

Vedere anche

`XML.nodeType`

XML.onData

Disponibilità

Flash Player 5

Uso

`myXML.onData()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando viene completato lo scaricamento di testo XML dal server o quando, durante lo scaricamento, si verifica un errore. Questo gestore viene invocato prima che il testo XML sia analizzato e può dunque essere usato per chiamare una routine di analisi personalizzata invece di usare la funzione di analisi sintattica XML di Flash. Il metodo `XML.onData` restituisce il valore `undefined` oppure una stringa contenente il testo XML scaricato dal server. Se il valore restituito è `undefined`, si è verificato un errore durante lo scaricamento del testo XML dal server.

Per impostazione predefinita, il metodo `XML.onData` invoca il metodo `XML.onLoad`. È possibile sovrascrivere il metodo `XML.onData` con un'azione personalizzata; tuttavia, `XML.onLoad` non verrà più richiamato salvo che durante l'implementazione del metodo `XML.onData` personalizzato.

Esempio

L'esempio seguente mostra i valori predefiniti del metodo `onData`:

```
XML.prototype.onData = function (src) {
    if (src == undefined) {
        this.onLoad(false);
    }
    else {
        this.parseXML(src);
        this.loaded = true;
        this.onLoad(true);
    }
}
```

Il metodo `XML.onData` può essere sovrascritto per poter riconoscere il testo XML senza ricorrere all'analisi sintattica.

XML.onLoad

Disponibilità

Flash Player 5.

Uso

```
myXML.onLoad(successo)
```

Parametri

successo Valore booleano che indica se l'oggetto XML è stato caricato correttamente tramite l'azione `XML.load` o `XML.sendAndLoad`.

Valori restituiti

Nessuno.

Descrizione

Metodo; chiamato da Flash Player quando un documento XML viene ricevuto dal server. Se il documento XML viene ricevuto correttamente, il parametro *successo* risulta `true`. Se il documento non è stato ricevuto o se si è verificato un errore nel ricevimento della risposta dal server, il parametro *successo* risulta `false`. L'implementazione predefinita di questo metodo non implica alcuna azione. Per ridefinire tale implementazione, è necessario assegnare una funzione contenente le proprie azioni.

Esempio

L'esempio seguente crea un semplice filmato Flash per una semplice applicazione di vetrina e-commerce. Il metodo `sendAndLoad` viene usato per passare un elemento XML contenente il nome e la password dell'utente e installare il gestore `onLoad` per la gestione della risposta dal server.

```
function myOnLoad(success) {
    if (success) {
        if (e.firstChild.nodeName == "LOGINREPLY" &&
            e.firstChild.attributes.status == "OK") {
            gotoAndPlay("loggedIn")
        } else
            gotoAndStop("loginFailed")
    }
    else
        gotoAndStop("connectionFailed")
}
var myLoginReply = new XML();
myLoginReply.onLoad = myOnLoad;
myXML.sendAndLoad("http://www.samplestore.com/login.cgi",
    myLoginReply);
```

Vedere anche

`function`, `XML.load`, `XML.sendAndLoad`

XML.parentNode

Disponibilità

Flash Player 5.

Uso

myXML.parentNode

Descrizione

Proprietà (sola lettura); fa riferimento al nodo principale dell'oggetto XML specificato o restituisce il valore `null` se il nodo non ha nodo principale. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

XML.parseXML

Disponibilità

Flash Player 5.

Uso

myML.parseXML(*origine*)

Parametri

origine Il testo XML di cui eseguire l'analisi sintattica e da passare all'oggetto XML specificato.

Valori restituiti

Nessuno.

Descrizione

Metodo; esegue l'analisi sintattica del testo XML specificato nel parametro *origine*, quindi inserisce gli elementi dell'albero XML risultante nell'oggetto XML specificato. Gli eventuali alberi XML contenuti nell'oggetto XML vengono rimossi.

XML.previousSibling

Disponibilità

Flash Player 5.

Uso

myXML.previousSibling

Descrizione

Proprietà (sola lettura); restituisce un riferimento al precedente nodo dello stesso livello nell'elenco dei nodi secondari del nodo principale. Restituisce `null` se il nodo non ha un nodo precedente allo stesso livello. Questa proprietà è di sola lettura e non può essere usata per manipolare nodi secondari. Usare i metodi `appendChild`, `insertBefore` e `removeNode` per manipolare i nodi secondari.

XML.removeNode

Disponibilità

Flash Player 5.

Uso

```
myXML.childNodes[1].removeNode()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; rimuove l'oggetto XML specificato dal relativo nodo principale. Tutti i relativi sottonodi vengono eliminati.

XML.send

Disponibilità

Flash Player 5.

Uso

```
myXML.send(url, [finestra])
```

Parametri

url L'URL di destinazione dell'oggetto XML specificato.

finestra La finestra del browser che visualizza i dati restituiti dal server: *_self* specifica il frame corrente nella finestra corrente, *_blank* specifica una nuova finestra, *_parent* specifica il frame principale del frame corrente e *_top* specifica il frame di primo livello nella finestra corrente. Questo parametro è opzionale; è possibile non specificare il parametro *finestra*, con esito analogo a quello che si ottiene specificando *_self*.

Valori restituiti

Nessuno.

Descrizione

Metodo; codifica l'oggetto XML specificato e lo trasforma in un documento XML, per poi inviarlo all'URL specificato tramite il metodo POST.

XML.sendAndLoad

Disponibilità

Flash Player 5.

Uso

```
mtXML.sendAndLoad(url, oggettoXMLtarget)
```

Parametri

url L'URL di destinazione dell'oggetto XML specificato. L'URL deve trovarsi nello stesso sottodominio dell'URL dal quale è stato scaricato il filmato.

oggettoXMLtarget Oggetto XML creato con la relativa funzione di costruzione che riceverà le informazioni restituite dal server.

Valori restituiti

Nessuno.

Descrizione

Metodo; codifica l'oggetto XML specificato e lo trasforma in un documento XML, lo invia all'URL specificato tramite il metodo `POST`, scarica la risposta del server e quindi la carica nell'oggetto *oggettoXMLtarget* specificato nei parametri. La procedura per il caricamento della risposta del server è analoga a quella usata dal metodo `load`.

Vedere anche

`XML.load`

XML.status

Disponibilità

Flash Player 5.

Uso

myXML.status

Descrizione

Proprietà; imposta e restituisce automaticamente un valore numerico se un documento XML è stato analizzato correttamente e trasformato in un oggetto XML. I codici numerici di stato e la relativa descrizione sono elencati di seguito:

- 0 Nessun errore; l'analisi sintattica è stata completata correttamente.
- -2 Una sezione di CDATA non è stata terminata correttamente.
- -3 La dichiarazione XML non è stata terminata correttamente.
- -4 La dichiarazione DOCTYPE non è stata terminata correttamente.
- -5 Un commento non è stato terminato correttamente.
- -6 È presente un elemento XML non strutturalmente corretto.
- -7 Memoria esaurita.
- -8 Un valore attributo non è stato terminato correttamente.
- -9 È presente un tag di apertura senza tag di chiusura corrispondente.
- -10 È presente un tag di chiusura senza tag di apertura corrispondente.

XML.toString

Disponibilità

Flash Player 5.

Uso

```
myXML.toString()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; valuta l'oggetto XML specificato, costruisce una rappresentazione testuale della struttura XML, includendo in essa il nodo, i nodi secondari e gli attributi, quindi restituisce il risultato in formato stringa.

Nel caso di oggetti XML di primo livello, ovvero creati con la funzione di costruzione, l'output del metodo `XML.toString` sarà costituito dalla dichiarazione XML del documento (memorizzata nella proprietà `XML.xmlDecl`), seguita dalla dichiarazione DOCTYPE del documento (memorizzata nella proprietà `XML.docTypeDecl`), quindi dalla rappresentazione testuale di tutti i nodi XML nell'oggetto. La dichiarazione XML non viene inclusa nell'output se la proprietà `XML.xmlDecl` non è definita. La dichiarazione DOCTYPE non viene inclusa nell'output se la proprietà `XML.docTypeDecl` non è definita.

Esempio

Il codice seguente illustra l'uso del metodo `XML.toString` per inviare `<h1>test</h1>` alla finestra Output.

```
node = new XML("<h1>test</h1>");  
trace(node.toString());
```

Vedere anche

`XML.docTypeDecl`, `XML.xmlDecl`

XML.xmlDecl

Disponibilità

Flash Player 5.

Uso

```
myXML.xmlDecl
```

Descrizione

Proprietà; imposta e restituisce informazioni sulla dichiarazione XML di un documento. Dopo aver eseguito l'analisi sintattica di un documento XML e averlo trasformato in oggetto XML, questa proprietà viene assegnata al testo della dichiarazione XML del documento. Questa proprietà viene impostata usando il formato stringa della dichiarazione XML, non un oggetto nodo XML. Se l'operazione di analisi sintattica non incontra alcuna dichiarazione XML, la proprietà viene impostata su `undefined.XML`. Il metodo `toString` genera i contenuti di `XML.xmlDecl` prima di qualsiasi altro testo presente nell'oggetto XML. Se `XML.xmlDecl` contiene il tipo `undefined`, l'output non conterrà la dichiarazione XML.

Esempio

Nell'esempio seguente, viene usata la proprietà `XML.xmlDecl` per impostare la dichiarazione XML del documento per un oggetto XML.

```
OggettoXML.xmlDecl = "<?xml version=\"1.0\" ?>";
```

L'esempio seguente illustra una dichiarazione XML.

```
<?xml version="1.0" ?>
```

Vedere anche

`XML.docTypeDecl`, `XML.toString`

XMLSocket (oggetto)

L'oggetto `XMLSocket` implementa dei socket lato client che consentono al computer che esegue Flash Player di comunicare con un computer server identificato da un indirizzo IP o da un nome di dominio.

Uso dell'oggetto XMLSocket

Per usare l'oggetto `XMLSocket`, è necessario che sul computer server sia in esecuzione un daemon compatibile con il protocollo usato dall'oggetto stesso. Tale protocollo è il seguente:

- I messaggi XML vengono inviati attraverso una connessione socket in streaming TCP/IP full duplex.
- Ciascun messaggio XML è un documento XML completo, che termina con un byte a zero.
- È possibile inviare e ricevere un numero illimitato di messaggi XML attraverso una singola connessione `XMLSocket`.

L'oggetto `XMLSocket` è utile per le applicazione client server che richiedono un intervallo di latenza basso, quali i sistemi di conversazione in tempo reale. In genere il funzionamento dei sistemi di conversazione tradizionali basati sul protocollo HTTP consiste nell'interrogare il server e quindi scaricare i nuovi messaggi tramite una richiesta HTTP. Il sistema di conversazione `XMLSocket`, al contrario, mantiene la connessione al server aperta in modo da consentire a quest'ultimo di inviare subito i messaggi in arrivo senza aspettare la richiesta da parte del client.

La configurazione di un server che comunichi con l'oggetto `XMLSocket` può risultare impegnativa. Se l'applicazione non richiede un'interazione in tempo reale, usare l'azione `loadVariables` o i metodi di connettività al server dell'oggetto XML basati sul protocollo HTTP disponibili in Flash (`XML.load`, `XML.sendAndLoad`, `XML.send`), anziché i metodi dell'oggetto `XMLSocket`.

Per poter usare i metodi dell'oggetto `XMLSocket`, è necessario innanzitutto creare un nuovo oggetto `XMLSocket` tramite la funzione di costruzione `new XMLSocket`.

Oggetto XMLSocket e sicurezza

Poiché l'oggetto `XMLSocket` stabilisce una connessione al server e la mantiene aperta, sono state definite restrizioni specifiche per ragioni di sicurezza.

- Il metodo `XMLSocket.connect` consente di effettuare la connessione solo ai numeri di porta TCP maggiori o uguali a 1024. Ne consegue, fra l'altro, che ai daemon server che comunicano con l'oggetto `XMLSocket` deve essere assegnato un numero di porta maggiore o uguale a 1024. I numeri di porta minori vengono spesso usati dai servizi di sistema quali FTP, Telnet e HTTP e questa restrizione esclude l'oggetto `XMLSocket` da tali porte. In tal modo viene ridotta la possibilità di accesso non autorizzato e di abuso di tali risorse.
- Il metodo `XMLSocket.connect` consente di effettuare la connessione solo ai computer all'interno dello stesso sottodominio in cui risiede il file SWF (il filmato). Questa restrizione non si applica ai filmati in esecuzione da un disco locale e corrisponde esattamente alle regole di sicurezza relative all'azione `loadVariables` e ai metodi `XML.sendAndLoad` e `XML.load`.

Riepilogo dei metodi validi per l'oggetto XMLSocket

Metodo	Descrizione
<code>XMLSocket.close</code>	Chiude una connessione socket aperta.
<code>XMLSocket.connect</code>	Stabilisce una connessione al server specificato.
<code>XMLSocket.close</code>	Invia un oggetto XML al server.

Riepilogo dei gestori di evento validi per l'oggetto XMLSocket

Metodo	Descrizione
<code>XMLSocket.onClose</code>	Funzione di callback invocata quando viene chiusa una connessione <code>XMLSocket</code> .
<code>XMLSocket.onConnect</code>	Funzione di callback invocata quando viene stabilita una connessione <code>XMLSocket</code> .
<code>XMLSocket.onData</code>	Funzione di callback invocata quando è stato scaricato dal server un messaggio XML.
<code>XMLSocket.onXML</code>	Funzione di callback invocata quando arriva dal server un oggetto XML.

Funzione di costruzione per l'oggetto XMLSocket

Disponibilità

Flash Player 5.

Uso

```
new XMLSocket()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Funzione di costruzione; crea un nuovo oggetto `XMLSocket`. L'oggetto `XMLSocket` inizialmente non è collegato a un server. Ciò avviene quando viene richiamato il metodo `XMLSocket.connect`.

Esempio

```
myXMLSocket = new XMLSocket();
```

Vedere anche

`XMLSocket.connect`

XMLSocket.close

Disponibilità

Flash Player 5.

Uso

```
myXMLSocket.close()
```

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; chiude la connessione specificata dall'oggetto XMLSocket.

Vedere anche

XMLSocket.connect

XMLSocket.connect

Disponibilità

Flash Player 5.

Uso

```
myXMLSocket.connect(host, porta)
```

Parametri

host Il nome di dominio DNS completo o l'indirizzo IP nel formato *aaa.bbb.ccc.ddd*. È inoltre possibile specificare la parola chiave `null` per stabilire la connessione al server *host* su cui risiede il filmato.

porta Il numero di porta TCP sull'*host* tramite la quale si stabilisce la connessione. Il numero di porta deve essere maggiore o uguale a 1024.

Valori restituiti

Nessuno.

Descrizione

Metodo; stabilisce una connessione all'*host* Internet indicato tramite la porta TCP specificata (numero maggiore o uguale a 1024), quindi restituisce `true` o `false` a seconda del risultato del tentativo di connessione. Se non si conosce il numero di porta del computer *host* per Internet, rivolgersi all'amministratore di rete. Se è in uso il plug-in Netscape di Flash o il controllo ActiveX, il sottodominio dell'*host* specificato nel relativo parametro e quello dell'*host* da cui si è scaricato il filmato devono corrispondere.

Se si specifica la parola chiave `null` come valore del parametro *host*, verrà contattato l'*host* su cui risiede il filmato che chiama il metodo XMLSocket.connect. Se, ad esempio, il filmato è stato scaricato dal sito <http://www.sito.com> e si è specificato il valore `null` per il parametro *host*, ciò equivale ad aver immesso l'indirizzo IP del sito www.sito.com.

Se il metodo `XMLSocket.connect` restituisce il valore `true`, significa che la fase iniziale del processo di connessione è stata completata correttamente. In un secondo momento, viene invocato il metodo `XMLSocket.onConnect` al fine di determinare se la connessione finale è stata stabilita correttamente o se non è riuscita. Se il metodo `XMLSocket.connect` restituisce il valore `false`, significa che non è stato possibile stabilire una connessione.

Esempio

L'esempio seguente usa il metodo `XMLSocket.connect` per effettuare una connessione all'host su cui risiede il filmato, quindi usa l'azione `trace` per visualizzare il valore restituito che indica se la connessione è stata stabilita correttamente.

```
function myOnConnect(success) {
    if (success) {
        trace ("Connessione riuscita!");
    } else {
        trace ("Connessione non riuscita!");
    }
}
socket = new XMLSocket()
socket.onConnect = myOnConnect
if (!socket.connect(null, 2000)) {
    trace ("Connessione non riuscita!")
}
```

Vedere anche

`function`, `XMLSocket.onConnect`

XMLSocket.onClose

Disponibilità

Flash Player 5.

Uso

myXMLSocket.onClose()

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Metodo; funzione di callback invocata solo quando una connessione aperta viene chiusa dal server. L'implementazione predefinita di questo metodo non implica l'esecuzione di azioni. Per ridefinire tale implementazione, è necessario assegnare una funzione contenente le proprie azioni.

Vedere anche

`function`, `XMLSocket.onConnect`

XMLSocket.onConnect

Disponibilità

Flash Player 5.

Uso

```
myXMLSocket.onConnect(successo)
```

Parametri

successo Valore booleano che indica se la connessione socket è stata stabilita correttamente (true o false).

Valori restituiti

Nessuno.

Descrizione

Metodo; una funzione di callback invocata da Flash Player quando una richiesta di connessione effettuata tramite il metodo `XMLSocket.connect` è stata stabilita o non è riuscita. Se la connessione è stata stabilita, il valore del parametro *successo* è true; se non è riuscita, il valore del parametro *successo* è false.

L'implementazione predefinita di questo metodo non implica l'esecuzione di azioni. Per ridefinire tale implementazione, è necessario assegnare una funzione contenente le proprie azioni.

Esempio

Nell'esempio seguente viene specificata una funzione che sostituisce il metodo `onConnect` in una semplice applicazione di conversazione in linea.

La funzione controlla la finestra alla quale avrà accesso l'utente, a seconda che la connessione sia stata stabilita correttamente o meno. Se la connessione è stata stabilita correttamente, verrà visualizzata la finestra principale di conversazione nel fotogramma etichettato `startChat`. Se la connessione non è riuscita, verrà visualizzata la finestra contenente le informazioni per la risoluzione dei problemi nel fotogramma etichettato `connectionFailed`.

```
function myOnConnect(success) {  
    if (success) {  
        gotoAndPlay("startChat")  
    }  
    else {  
        gotoAndStop("connectionFailed")  
    }  
}
```

Dopo aver creato un oggetto `XMLSocket` tramite la funzione di costruzione, lo script inserisce il metodo `onConnect` tramite l'operatore di assegnazione:

```
socket = new XMLSocket()  
socket.onConnect = myOnConnect
```

Infine, viene avviato il processo di connessione. Se `connect` restituisce `false`, il filmato viene inviato direttamente al fotogramma etichettato `connectionFailed` e il metodo `onConnect` non viene invocato. Se `connect` restituisce `true`, il filmato passa al fotogramma etichettato `waitForConnection`, ossia a una finestra di attesa. Il filmato rimane nel fotogramma etichettato `waitForConnection` fino a quando viene invocato il gestore `onConnect` in un arco di tempo che dipende dalla latenza di rete.

```
if (!socket.connect(null, 2000)) {
    gotoAndStop("connectionFailed")
} else {
    gotoAndStop("waitForConnection")
}
```

Vedere anche

`function, XMLSocket.connect`

XMLSocket.onData

Disponibilità

Flash Player 5.

Uso

`XMLSocket.onData()`

Parametri

Nessuno.

Valori restituiti

Nessuno.

Descrizione

Gestore di evento; invocato quando un messaggio XML che termina con un byte a zero è stato scaricato dal server.

Per impostazione predefinita, il metodo `XMLSocket.onData` invoca il metodo `XMLSocket.onLoad`. Se si sovrascrive il metodo `XMLSocket.onData` con un'azione personalizzata, `XMLSocket.onLoad` non verrà più salvo che durante l'implementazione del metodo `XMLSocket.onData` personalizzato.

```
XMLSocket.prototype.onData = function (src) {
    this.onXML(new XML(src));
}
```

Nell'esempio precedente, il parametro `src` è una stringa contenente testo XML scaricato dal server. Il byte a zero finale non è incluso nella stringa.

`XMLSocket.onData` può essere sovrascritto per poter riconoscere il testo XML senza ricorrere all'analisi sintattica.

XMLSocket.onXML

Disponibilità

Flash Player 5.

Uso

myXMLSocket.onXML(oggetto)

Parametro

oggetto Un'istanza dell'oggetto XML che contiene un documento XML di cui è stata eseguita l'analisi sintattica, proveniente da un server.

Valori restituiti

Nessuno.

Descrizione

Metodo; una funzione di callback invocata da Flash Player quando l'oggetto XML specificato che contiene un documento XML viene ricevuto tramite una connessione XMLSocket aperta. La connessione XMLSocket consente di trasferire tra il client e il server un numero illimitato di documenti XML. Tutti i documenti terminano con un byte a zero. Nel momento in cui riceve il byte a zero, Flash Player esegue l'analisi sintattica di tutti i documenti XML ricevuti nell'intervallo trascorso dall'ultimo byte con valore zero oppure, se si tratta del primo messaggio ricevuto, dal momento in cui è stata stabilita la connessione. Ogni blocco di documenti XML sottoposto ad analisi sintattica viene gestito come un singolo documento XML e viene passato al metodo `onXML`.

L'implementazione predefinita di questo metodo non implica l'esecuzione di azioni. Per ridefinire tale implementazione, è necessario assegnare una funzione contenente le proprie azioni.

Esempio

La funzione seguente ridefinisce l'implementazione predefinita del metodo `onXML` in una semplice applicazione di conversazione in linea. La funzione `myOnXML` indica all'applicazione di conversazione in linea come riconoscere un singolo elemento XML, MESSAGE, nel formato seguente:

```
<MESSAGE USER="Giovanni" TEXT="Ciao, il mio nome è Giovanni!" />.
```

È necessario aver inserito il gestore `onXML` nell'oggetto XMLSocket nel modo seguente:

```
socket.onXML = myOnXML;
```

La funzione `displayMessage` rappresenta una funzione definita dall'utente per visualizzare il messaggio.

```
function myOnXML(doc) {  
    var e = doc.firstChild;  
    if (e != null && e.nodeName == "MESSAGGIO") {  
        displayMessage(e.attributes.user, e.attributes.text);  
    }  
}
```

Vedere anche

`function`

XMLSocket.close

Disponibilità

Flash Player 5.

Uso

```
myXMLSocket.send(oggetto)
```

Parametri

oggetto Oggetto XML o altri dati da trasmettere al server.

Valori restituiti

Nessuno.

Descrizione

Metodo; converte l'oggetto XML o i dati specificati nel parametro *oggetto* in una stringa che viene quindi trasmessa al server, seguita da un byte a zero. Se l'argomento *oggetto* contiene un oggetto XML, la stringa sarà la rappresentazione testuale XML dell'oggetto stesso. Questa operazione è asincrona, ossia, anche se restituisce subito un valore, i dati potrebbero essere trasmessi in un secondo momento. Il metodo `XMLSocket.send` non restituisce un valore che indica se i dati sono stati trasmessi correttamente.

Se l'oggetto *OggettoXMLSocket* non è collegato al server (tramite il metodo `XMLSocket.connect`), il metodo `XMLSocket.send` avrà esito negativo.

Esempio

L'esempio seguente illustra il codice per specificare un nome utente e una password per l'invio dell'oggetto XML `myXML` al server.

```
var myXML = new XML();
var myLogin = myXML.createElement("login");
myLogin.attributes.username = usernameTextField;
myLogin.attributes.password = passwordTextField;
myXML.appendChild(myLogin);
myXMLSocket.send(myXML);
```

Vedere anche

`XMLSocket.connect`

