

My \LaTeX Notes

The Messy Notes of a \LaTeX User

Samuele Carcagno

Welcome! Some clarifications first: Though the layout of the first pages (with a table of contents etc..) make this look like a true \LaTeX guide, it isn't!! These are just notes I'm writing down in the process of learning \LaTeX , so they just address some specific problems I'm running into during this process. With these notes I want to gather in one single place, in a form that is sensible for me, all the information I need to use \LaTeX , information that comes from different places, like tutorials on the internet and the documentation of some \LaTeX packages. I'm putting *some* effort into making these notes clear also for other people that may read them, so that perhaps some other people can benefit from them. This is work in progress, so these notes are very sketchy and messy. These notes come with absolutely NO WARRANTY whatsoever. As I said above I'm in the process of learning \LaTeX and therefore I'm not an expert. However, I think they may be useful because there are solution to problems that I, as a beginner, spent hours to get solved looking for some information in the internet or reading some obscure package documentation. Also forgive my poor English, it's not my first language and what's more, I rarely check two times what I've written... Enjoy!!

Contents

1	Introduction	1
1.1	The Structure of a \LaTeX Document: The Preamble	1
2	Headings	3
2.1	Section numbering	3
3	Fonts	5
3.1	Font Size	5
3.2	Font Shape	5
3.3	Font Family	6
3.4	Font type	6
4	Special Characters	10
4.1	Verbatim	10
4.2	Commenting out text	10
5	Document Layout	11
5.1	Page Layout	11
5.1.1	Multiple columns in the same page	11
5.2	Paragraph Layout	11
5.2.1	Indent	11
5.2.2	Doublespace	12
5.3	Spacing	12
6	Tables	13
6.1	Formatting Tables	13
6.1.1	Cells that Span Several Columns	13
6.1.2	Referencing Tables	15
6.1.3	Starting a new line inside a cell	16
6.1.4	Adjusting the spacing between rows	16
6.2	Tables that span several pages	16
6.2.1	The longtable package	16
6.2.2	The xtab package	20

6.3	Publication Quality Tables: The booktabs Package	22
7	Typesetting Math	25
8	Writing with Style	26
9	pdfL^AT_EX	27
9.1	Adding hyperlinks to your document	27
10	hyperref	28
10.0.1	Hyperlinked pdf Documents from dvi Files	28
11	Colours	30
11.1	Adding colours to your documents	30
11.1.1	Colour Models	30
11.1.2	Other Stuff You Can Colour	34
12	Graphics	35
12.1	Imported Graphics	35
13	Language Selection	36
13.1	Windows: MiK _T E _X users	36
13.2	GNU/Linux: Debian Based Distributions	36
13.3	Using Babel	37
13.4	Accents	37
14	Adding a Bibliography	38
14.1	Bib _T E _X	38
14.1.1	Special Characters in Bib _T E _X	38
14.2	Apacite	39
15	Adding an Index	40
16	Adding a Glossary	41
16.1	The Glossary Package	41
A	Info and Resources for L^AT_EX	43
	References	44

Chapter 1

Introduction

1.1 The Structure of a \LaTeX Document: The Preamble

Every \LaTeX document starts with a declaration about the type of document you're writing, or better, its *class*. The class of a \LaTeX document can be `article`, `report` or `book`. After declaring the class of the document you can add other options or directly start your document. Let's see an example:

```
\documentclass[a4paper,12pt]{article}
```

```
\begin{document}
```

Here you actually write what goes inside your document.

```
\end{document}
```

In the example above we declared the class of the document (`article`), and then we directly started the document with the command `\begin{document}`, we wrote something to appear in the document and we ended the document with the command `\end{document}`. The space between the `documentclass` command and the `begin` command is called the **preamble**, and that's the place where you call other packages you want to use for your document. For example, if you want to include a picture in your document, you need to call the `graphicx` package in the preamble of your document first. Then inside your document you'll have the possibility to use the commands for including your photo with the `graphicx` package. Below there's an example:

```
\documentclass[a4paper,12pt]{article}
```

```
\usepackage{graphicx}
```

```
\usepackage[italian]{babel}
```

```
\begin{document}
```

Here you actually write what goes inside your document.

```
\end{document}
```

in this example we have called two packages, the `graphicx` package and then the `babel` package which can be used to load the hyphenation patterns for languages other than English¹. In this case we've used an **option** to tell \LaTeX to use the hyphenation patterns of the Italian language. Options go in square brackets, virtually every package has one or more options you can use when calling it. Other two options we declared go with `documentclass` command, and they are the format of the paper we want to use, in our case `a4` and the size of the font, `12pt` in the example. As you can see you can declare more than one option at a time when calling a given package, by separating them with a comma.

¹You actually may have to edit a configuration file to load the hyphenation patterns for your language, see ahead the chapter **Language Selection** for information on how to do this

Chapter 2

Headings

With \LaTeX you use sectioning commands that automatically define the way chapter titles, section titles, subsection titles and so on, are positioned and appear in your document. This easily allows you to define the way different titles look like in your document in a consistent way, which would be more difficult if you had to manually change each time the font size or the position of a title, as you do with some word processors. If you don't like the way \LaTeX defines your titles by default, you can change it anyway.

Some sectioning commands in descending order are:

```
\chapter{Title of your chapter}
```

The chapter command is available only in the report and book classes.

```
\section{Title of your section}
```

```
\subsection{Title of your subsection}
```

```
\subsubsection{Title of your subsubsection}
```

For the chapter command, by default \LaTeX will insert the heading **Chapter** with the number of the chapter, before the title you give to your chapter. If you want to change the heading for chapter, use the command:

```
\renewcommand{\chaptername}{your chaptername}
```

if you want to have the number of the chapter, yet without the associated heading, just leave the second argument blank.

2.1 Section numbering

For all the other sectioning commands except subsubsection, \LaTeX will just insert the sectional numbering. To get rid of sectional numbering ¹ you can use a **starred** version of the sectioning commands, for example:

¹This applies also for chapters

```
\section*{A section without the associated number}
```

will yield the title of your section without the associated numbering.

Using the starred version of the the commands defining the section headings, you get rid of the numbering, but in addition the starred sectional heading won't appear in the table of contents. To get the section headings without numbering, but inside the table of content write:

```
\setcounter{secnumdepth}{-1}
```

be careful however, this will get the name of the section or chapter in the table of contents, but there won't be any numbering there either!

Chapter 3

Fonts

3.1 Font Size

The following control sequences change the size of your font from the place you declare them onwards. To return to the default size use the `\normalsize` control sequence.

```
\tiny tiny  
\scriptsize scriptsize  
\footnotesize footnotesize  
\small small  
\normalsize normalsize  
\large large  
\LARGE LARGE  
\huge huge  
\Huge Huge
```

3.2 Font Shape

You can change the font shape with one of the *commands* on the left column below, enclosing the text whose shape you want to change within braces. Otherwise you can use one of the *declarations* on the right column below, to change the shape of your text from the point you give the declaration onward. In the latter case you don't enclose your text within braces.

Command

```
\textit{your text} italic
```

Declarations

```
\itshape
```

<code>\emph{your text}</code>	<i>emphasised</i>	
<code>\textup{your text}</code>	upright shape	<code>\upshape</code>
<code>\textsl{your text}</code>	<i>slanted shape</i>	<code>\slshape</code>
<code>\textmd{your text}</code>	medium series	<code>\mdseries</code>
<code>\textbf{your text}</code>	boldface	<code>\bfseries</code>
<code>\textsc{your text}</code>	SMALL CAPITAL	<code>\scshape</code>

3.3 Font Family

You can change the font family in the very same way you change the font shape (see description above)

Command		Declarations
<code>\textrm{your text}</code>	roman family	<code>\rmfamily</code>
<code>\textsf{your text}</code>	sans serif family	<code>\ssfamilly</code>
<code>\texttt{your text}</code>	typewriter family	<code>\ttfamily</code>
<code>\textnormal{your text}</code>	default font	

3.4 Font type

To change font type in the preamble type:

```
\usepackage{yourfont}
```

for example, to get palatino write:

```
\usepackage{palatino}
```

Table 3.3: Font packages

Package name	Description
ae	
aecompl	
avant	
bera	
bookman	
calligra	
chancery	Quite gothic!!
concrete	
courier	
fourier	Like utopia and supports math as well
helvet	
lmodern	Latin Modern
lucida	Lucida bright fonts
luximono	A family of monospaced typewriter fonts

Package name	Description
mathpazo, bm	get palatino in both text and math
mathptmx	Get times in both text and math
newcent	
palatino	
pandora	
pifont	
punk	Donald Knuth's punk font. Metafont
times	
txfonts	TX fonts
universal	The universal font. Metafont
utopia	

You can also re-size fonts, for example the helvetica font looks better if scaled down a little bit, to do this:

```
\usepackage[scaled=0.92]{helvet}
```

To change the font type only for a portion of the document you can use the following command:

```
{\fontfamily{phv}\selectfont your text in the new font!} old font again
```

here the font family phv indicates the Helvetica font, table 3.4 lists the names for other fonts:

cmr	Computer Modern Roman
cmss	Computer Modern Sans Serif
cmtt	Computer Modern Typewriter
pag	Avantgarde
pbk	Bookman
phv	Helvetica
pnc	New Century Schoolbook
ppl	Palatino
ptm	Times
pcr	Courier

Table 3.4: Font Families for Changing the Font on the Fly!

Chapter 4

Special Characters

4.1 Verbatim

To print out special characters, you can use a verbatim output through:

```
\verb+yourspecialtext+
```

To produce text, exactly as you typeset it including spaces and some special characters, use the verbatim environment:

```
\begin{verbatim}
```

```
\end{verbatim}
```

4.2 Commenting out text

To insert comments you can use the % special character, anything written after it in a line will be considered as comment and will not appear in the output.

For example

```
The number of subjects is % INSERT NUMBER OF SUBJ. HERE!!  
The subjects were all healthy...
```

Chapter 5

Document Layout

5.1 Page Layout

5.1.1 Multiple columns in the same page

If you want to have multiple columns in the same page you can use the **multicols** environment. To use it you need first to declare its package in the preamble:

```
\usepackage{multicol}
```

then inside your document, where you want to have your multicolumn output, start the multicolumn environment with:

```
\begin{multicols}{number_of_columns}
```

```
....
```

```
\end{multicols}
```

Please notice that the label used to declare the package and the one used to start the environment are different, as the latter ends with a ‘s’ in addition.

5.2 Paragraph Layout

5.2.1 Indent

In \LaTeX all paragraphs but the first one of a section are indented by default, if you want to change the width of the indentation you can use:

```
\setlength{\parindent}{2pt}
```

If you don’t want to have paragraph indentation altogether, using instead a wider space between paragraphs, you can use the above commands to set the length of

paragraph indent to 0, and in addition the following command:

```
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

to get a wider spacing between paragraphs. Beware that the latter command, if put in the preamble will also change the spacing between lines of the table of contents. In order to avoid this problem you can declare it directly in your document, after the table of contents.

Check package indentfirst

5.2.2 Doublespace

To get double spaced text, put in the preamble:

```
\usepackage{setspace}
```

Then in the text, wherever you want doublespace use the command:

```
\doublespace
```

and that's it.

5.3 Spacing

You can define horizontal and vertical spaces with the `\hspace{}` and `\vspace{}` commands respectively, giving as an argument the amount of space you want in whatever unit of measure you want to, from points to centimetres.

There are other options, for horizontal spacing:

```
# \thinspace (1/12em), which is used for fine adjustments
# \enspace (1/2em)
# \quad (1 em)
# \qquad (2em)
```

For vertical spacing:

```
\smallskip
\medskip
\bigskip
```

these can be used only after a paragraph break, that's also true of `\vspace`. Differently from `\vspace`, these commands will automatically compress or expand a bit, depending on the demands of the rest of the page.

Chapter 6

Tables

6.1 Formatting Tables

```
\begin{table}[ht]
\begin{tabular}[]{|c|l|c|r|}
\hline
first cell (R11) & second cell (R12)
\hline
(R21) & (R22)
\end{tabular}
\end{table}
```

The optional argument `h` in the first line

```
\begin{table}[h]
```

puts the table ‘right here’, meaning right there where you put it, this has to do with the problems of floating bodies, and other optional arguments are available to manage them apart from `h`.

To get centred tables you can either use the `ctab` package or use the `\centered` environment, before starting the tabular environment but *after* declaring a table.

6.1.1 Cells that Span Several Columns

You can use the `\multicol` command to format tables with cells that span several columns. Below there is an example, it uses the `booktabs` environments which is described later on. You can see the results in table 6.1

```
\begin{table}[htbp]

\caption{multicolumn example}
\centering
```

```

\begin{tabular}{cccc}

\toprule

\multicolumn{2}{c}{\textbf{Drug}}& \multicolumn{2}{c}{\textbf{No-Drug}} \\

\cmidrule(1{.75em}r{.75em}){1-2}
\cmidrule(1{.75em}r{.75em}){3-4}

\textbf{Alcohol}&\textbf{No-Alcohol}&\textbf{Alcohol}&\textbf{No-Alcohol} \\

\midrule

7& 6& 6 & 4 \\
5& 4& 5 & 2 \\
8& 7& 7 & 4 \\

\bottomrule

\end{tabular}

\label{tab:multicolumn}

\end{table}

```

Table 6.1: multicolumn example

Drug		No-Drug	
Alcohol	No-Alcohol	Alcohol	No-Alcohol
7	6	6	4
5	4	5	2
8	7	7	4

6.1.2 Referencing Tables

Often you need to make a reference by number to a table in your text. This is very easy with \LaTeX in fact you don't have to keep track of the table number, which would be a problem if you add or delete another table before the one you're referencing. You just give the table a label with the `label` command, and make a reference to it with the `ref` command. For example we will label the next table as `\label{tab:example}`:

```
\caption{Referencing Example Table}
\centering
\begin{table}[htbp]

\begin{tabular}{c c}
\hline
Even Cells & Odd Cells \\
\hline
Cell 1 & Cell 2 \\
Cell 3 & Cell 4 \\
\hline
\end{tabular}

\label{tab:example}

\end{table}
```

and then we can reference to our table with :

```
\ref{tab:example}
```

Which would let me say, "You can see our referencing example in table 6.2". If you have the `hyperref` package also loaded, as I have, it will also automatically generate a link to the table, try clicking on my reference to the table on the text!

Table 6.2: Referencing Example Table

Even Cells	Odd Cells
Cell 1	Cell 2
Cell 3	Cell 4

6.1.3 Starting a new line inside a cell

In order to have cells that span several lines, you need to specify the column formatting as in the following example

```
\begin{tabular}{llp{20em}}
```

the first two cells are left aligned, and the line inside these cells will expand, but not break to a new line, the last cell however, called with `p{measure}` is given a specific width of 20em, if a line does not fit within this length it will be broken and a new line started.

6.1.4 Adjusting the spacing between rows

If you want to add blank space between rows you can use the command:

```
\renewcommand{\arraystretch}{yourvalue}
```

The default value for the `arraystretch` is 1, so you may set it to 1.5 or 2 to get more space between rows.

6.2 Tables that span several pages

6.2.1 The `longtable` package

To create tables that span several pages you cannot use the `tabular` environment, you can use instead the `longtable` environment. Its usage is basically the same of the `tabular` environment, there are however some differences. First, you cannot put the `longtable` environment inside the `table` environment. You don't have to worry about that however, because with the `longtable` environment, your table can still have a caption and get listed in the list of tables with its progressive number, as it would do with the `table` environment.

Another difference is that with the `longtable` package you can define how the header (first row) and the foot (last row) of your table should look like for each new page that the table spans, and if they should be different from the first header, or the last foot. Everything will become more clear with an example, the code is below and the result is table 6.3:

```
\begin{longtable}[c]{|p{4cm}|p{4cm}|}
```

```
\caption{Cities to visit in Sicily} \label{tab:longtable} \\
```

```
\hline
```

```
\textbf{City} & \textbf{Province} \\
\hline
\endfirsthead % these first three lines define the first header
```

```
\hline
\textbf{City} (continued) & \textbf{Province} (continued) \\
\hline
\endhead % this is the header for the other pages
```

```
continues below & continues below \\
\hline
\endfoot % this is the foot for every page but the last
```

```
\hline
End & End \\
\hline
\endlastfoot % this is the last foot
```

```
Catania& CT \\ % here begins the table
\hline
Taormina& ME \\
\hline
Isole Eolie & ME \\
\hline
Agrigento & AG \\
\hline
Palermo & PA \\
\hline
Siracusa & SR \\
\hline
Marina di Ragusa & RG\\
\hline
Noto & RG \\
\hline
Caltagirone & CT \\
\hline
Trapani & TP \\
\hline
Lipari & AG \\
\hline
```

```

Mineo & CT \\
\hline
Piazza Armerina & EN \\
\hline
Adrano & CT \\
\hline
Marsala & AG \\
\hline
Messina & ME \\
\hline
Acireale & CT \\
\hline
Buccheri & SR \\
\hline
Comiso & RG \\
\hline
Enna & EN \\
\hline
Caltanissetta & CL \\
\end{longtable}

```

We start the `longtable` environment with the `[c]` option, which tells \LaTeX to centre the table, we could have used the `[l]` or the `[r]` option instead, to tell \LaTeX to flush the table left or right respectively. Then, after having defined the **caption** and the **label** for the table in a line that we end with a **double backslash**, we start the header that will appear only on the first page that we end with the `\endfirsthead` command, in the same way we define the header that will appear on all other pages and we end it with the `\endhead` command. Then we define the tailfoot for each page but the last, ending it with the `\endfoot` command, and finally we define the foot for the last page and we end it with guess what...the `\endlastfoot` command. And that's it.

Table 6.3: Cities to visit in Sicily

City	Province
Catania	CT
Taormina	ME
continues below	continues below

City (continued)	Province (continued)
Isole Eolie	ME
Agrigento	AG
Palermo	PA
Siracusa	SR
Marina di Ragusa	RG
Noto	RG
Caltagirone	CT
Trapani	TP
Lipari	AG
Mineo	CT
Piazza Armerina	EN
Adrano	CT
Marsala	AG
Messina	ME
Acireale	CT
Buccheri	SR
Comiso	RG
Enna	EN
Caltanissetta	CL
End	End

6.2.2 The xtab package

Another way to create tables that span several pages is to use the `xtab` package. This package gives you options and results similar to the `longtable`, just the commands are different. Well, as usual you call the package in the preamble of your document:

```
\usepackage{xtab}
```

then be careful, because to start a new table you use:

```
\begin{xtabular}
```

I said, be careful, because in this case you don't use the same name that you use to call the package, and for this very reason I indeed got a series of disheartening error messages when I first tried to use the `xtab` package.

With `xtab` you can use the following commands:

```
\tablecaption{your caption}
```

refers to the caption of the table

```
\tablefirsthead{your main header}
```

 this is the header that will appear in very first row of your table if you want one

```
\tablehead{the headers for subsequent pages}
```

this is the header that will appear in the first row of each new page that the table spans, so it can be different from the first main header of the table, or you can also not use it altogether.

```
\tabletail{how your table finishes}
```

this is the tail of your table, that will appear each time the table ends a page or the table ends itself. As above you can customise it as you like. Below there is one example of usage for `xtabular`, please notice that when using the `xtabular` package, you don't use the `tabular` environment. Also notice that you decide the appearance of the headers, enclosing the header with a line at the top or not and choosing the appropriate font, it may not seem straightforward at first, but with some little experimenting everything will become clear :)

```
\tablecaption{Sicilian Provinces}
```

```

\tablefirsthead{\hline \textbf{City} & \textbf{Abbreviation} \\\}
\tablehead{\hline \textbf{City} & \textbf{Abbreviation} \\\}
\tabletail{\hline}
\begin{xtabular}{|p{5cm}|p{5cm}|}
\hline
Palermo & PA \\\}
\hline
Catania & CT \\\}
\hline
Messina & ME \\\}
\hline
Agrigento & AG \\\}
\hline
Siracusa & SR \\\}
\hline
Ragusa & RG \\\}
\hline
Trapani & TP \\\}
\hline
Caltanissetta & CL \\\}
Enna & EN \\\}
\hline
\end{xtabular}

```

Table 6.4: Sicilian Provinces

City	Abbreviation
Palermo	PA
Catania	CT
Messina	ME
Agrigento	AG

City	Abbreviation
Siracusa	SR
Ragusa	RG
Trapani	TP
Caltanissetta	CL
Enna	EN

6.3 Publication Quality Tables: The booktabs Package

The booktabs package lets you format tables that conform to typographical standards, such as those for scientific publishing. The major differences between these and other table formats you can produce in \LaTeX is the proper spacing of the horizontal rules, and the absence of vertical rules. You can pick up any scientific publication and see yourself to which standard they conform and how much clear and easier to read they are.

Well, as for the use of the booktabs package, you first call it in the preamble of your document, then you begin the tabular environment as usual, the only difference is that instead of using `\hline` for horizontal rules, you use `\toprule` for the first horizontal rule, `\midrule` for any horizontal rule between the top and the bottom rules, and `\bottomrule` for the rule at the bottom of the table. Additionally you can use `\cmidrule` for horizontal rules that extend only through some of the columns of the table. For the `\cmidrule` command, you additionally the columns range through which the rule should span, for example if you have a five columns table, and you want a rule to span only over the last two columns, you would use:

```
\cmidrule{4-5}
```

additionally you can trim a `\cmidrule` both on its left and/or right side by giving an option in parentheses as follows:

```
\cmidrule(lr){4-5}
```

with the above we've trimmed the `\cmidrule` both on its left (l) and on its right

(r) side of a default value, which is 0.5 em. You can change this value, for example, if you want the right side of the rule to be trimmed of 0.75 em you would write:

```
\midrule(lr{0.75em}){4-5}
```

Moreover, you can seamlessly integrate the `booktabs` package with the `longtable` package to get professionally formatted tables that span several pages.

Here is a complete example of a table formatted with the `booktabs` package:

```
\begin{table}[htp]
\caption{Grades for the \LaTeX course}\
\begin{tabular}{llll}

\toprule

Pupil & A & B & B \\\

\midrule

Jack & A & B & B \\\

Jim & B & C & C \\\

Karen & A & B & B \\\

Matt & A & A & B \\\

\bottomrule
\end{tabular}
\end{table}
```

Table 6.5: Grades for the L^AT_EXcourse

Pupil	Fonts	Page Layout	Tables
Jack	A	B	B
Jim	B	C	C
Karen	A	B	B
Matt	A	A	B

Chapter 7

Typesetting Math

`$$\bar{a}$$` or `$$\overline{a}$$`

Chapter 8

Writing with Style

If you want to get ellipsis, that is ‘...’, and actually type three dots, you get the wrong spacing. You should use instead the `\ldots` command, which gives the ellipsis with the correct formatting.

Chapter 9

pdfL^AT_EX

Instead of using L^AT_EX to process your document you can choose pdfL^AT_EX which allows you to create directly pdf files from your L^AT_EX source. From the point of view of the user apart from this fact there are only very few differences, mainly regarding the inclusion of graphic files. If you're a t_EX user, to process your L^AT_EX source file with pdfL^AT_EX at the shell prompt just use the command:

```
pdflatex your_file.tex
```

9.1 Adding hyperlinks to your document

To get hyperlinks and cross-references, just use the hyperref package, sometimes however you may want to add some links that the hyperref package hasn't added. To link to a U.R.L.:

```
\href{http://www.example.com}{Example}
```

for an e-mail address:

```
\href{mailto:example@example.com}{example}
```

Chapter 10

hyperref

The `hyperref` package automatically creates hyperlinks for your cross-references, for example for your table of contents, figures tables etc. Always load the `hyperref` package as the last one, otherwise it may cause problems!

In addition with the `hypersetup` command, which goes in the preamble, just after calling the `hyperref` package, you can customise the behaviour of the `hyperref` package to choose for example the colour of the links or the way the resulting pdf file (if you're using `pdftex`) is displayed. Below there's a setup example:

```
\usepackage{hyperref}

\hypersetup{pdftitle=My LaTeX Notes,
            pdfauthor=Samuele Carcagno,
            pdfpagemode=none,
            pdfstartview=FitH,
            pdfkeywords={LaTeX: document typesetting system},
            colorlinks=true,
            linkcolor=black,
            citecolor=black,
            filecolor=black,
            urlcolor=black
}
```

10.0.1 Hyperlinked pdf Documents from dvi Files

`hyperref` will produce directly a hyperlinked pdf document with `pdfTeX` however for different reasons sometimes you may want to first produce a dvi file with standard `LATEX` rather than `pdfTeX` and then convert it into a pdf file. There are different possibilities to convert a dvi file into a pdf.

dvipdfm

The dvipdfm program allows you to convert a dvi file into a hyperlinked pdf. First add the option dvipdfm' when calling the hyperref package in the preamble:

```
\usepackage[dvipdfm]{hyperref}
```

you can then specify the other options in the hypersetup as usual.

Next, you run the file through L^AT_EX producing the dvi file, and then you convert it to pdf using dvipdfm:

```
latex foo
dvipdfm foo.dvi
```

dvips and ghostscript

Another option is to use the dvips driver to first convert the dvi into a postscript file, and then convert the latter to pdf through ghostscript. To get hyperref right, you first need to tell it which driver you're going to use later for conversion, this is done when calling hyperref:

```
\usepackage[dvips]{hyperref}
```

Next you run the file through L^AT_EX producing the dvi file. The dvi file is then converted to postscript using dvips, and finally you can convert the postscript to pdf calling ghostscript through the ps2pdf utility:

```
latex foo
dvips foo.dvi
ps2pdf foo.ps
```

Some systems have a script, dvipdf, that invokes dvips and then ghostscript on its output, providing a shortcut for the conversion process:

```
latex foo
dvipdf foo.dvi
```

Chapter 11

Colours

11.1 Adding colours to your documents

To change the colour of your text, the colour of your page and do many other marvellous things, you can use the `xcolor` package. First you need to call it in the preamble of your document:

```
\usepackage{xcolor}
```

Now let's see the `xcolor` package at work, say you want to draw your reader's attention on a word by making it red, that's very simple:

```
\textcolor{red}{BEWARE!!}
```

which yields **BEWARE!!**

There is a set of predefined colours that you can use in this fashion with the `xcolor` package, they are listed in table 11.1, if you want other colours you'll have to do some extra effort working with **col or schemes**.

11.1.1 Colour Models















With `xcolor` you can directly change colours using colour models such as **HTML**, **rgb** or **cmyk**. With these colour models you have to use the appropriate code for a colour to use it. You can easily find these colour codes on the internet. One of the easiest is the **HTML** colour model, for example the HTML colour code for the colour gold is FFE4C4, so if you want your text gold:

```
\textcolor[HTML]{FFE4C4}{ gold text}
```

and this would be your resulting **gold text**. You can find more HTML colour codes in table 11.2

You can do the same with the **rgb** or the **cmyk** models, just you need to find the proper colour codes. For example:

Table 11.1: Predefined colours you can use with `xcolor`

 red	 yellow	 black
 green	 orange	 darkgray
 blue	 violet	 gray
 cyan	 purple	 lightgray
 magenta	 brown	white

```
\textcolor[rgb]{0.6,0.0,0.1}{rgb defined colour}
```

```
\textcolor[cmypk]{0.6,1.0,0.2,0.5}{cmypk defined colour}
```

rgb defined colour

cmypk defined colour

With the **rgb** model you can (almost) easily customise your colours, each colour is defined by three values that range from 0 to 1. The first parameter defines the amount of red, the second the amount of green and the third the amount of blue that will be present in your colour. If all the values are set to 0 you get black, if all the values are set to 1 you get white. By changing the value of the red, green, and blue parameters you can get all other colours. You can see some `rgb` codes on table 11.3.

Table 11.2: HTML Colour Codes























































 FAEBD7	 8B0000	 3CB371
 000000	 483D8B	 7B68EE
 0000FF	 B22222	 FFDEAD
 8A2BE2	 228B22	 000080
 A52A2A	 FFD700	 808000
 DEB887	 DAA520	 FFA500
 FF7F50	 CD5C5C	 DA70D6
 FFF8DC	 4B0082	 DDA0DD
 DC143C	 F08080	 800080
 00008B	 D3D3D3	 FF0000
 A9A9A9	 8470FF	 4169E1
 8B008B	 B0C4DE	 F4A460
 FF8C00	 0000CD	 C0C0C0
 9932CC	 9370D8	 6A5ACD

Table 11.3: RGB Colour Codes

 0 0 0	 0.0,0.6,0.8	 0.8,0.3,0.6
 1 1 1	 0.0,0.3,0.8	 0.1,0.3,0.9
 0.0,0.5,0.0	 0.5,0.0,0.0	 0.9,0.5,0.3
 0.0,0.9,0.0	 0.6,0.9,0.1	 0.4,0.2,0.1
 0.0,0.6,0.5	 0.9,0.7,0.3	 0.7,0.3,0.9

11.1.2 Other Stuff You Can Colour

As I said above you can do many more things with the `xcolor` package than just colouring text, for example you can change the background colour of your page with this command:

```
\pagecolor [HTML] {FFDEAD}
```

Chapter 12

Graphics

12.1 Imported Graphics

Of course latex can also include pictures or graphic files in various formats in your document. There are different ways of doing this, and as usual with latex hundreds of options. I'm not going to explain the gory details, also because I don't know them :)...Well the first simplifying assumption I will make is that you want to get your graphics in a pdf file and you are working directly with pdflatex, rather than creating first the dvi file and then transforming this in a pdf. The reason is that inserting graphics in dvi files is a quite different procedure than inserting them in pdf files, and since I'm a fan of pdf, I'm not going to explain the former. Said this, the first thing you need to do to do is to call a package in the preamble that handles your graphics, my advice is to use the graphicx package, so:

```
\usepackage{graphicx}
```

Then, let's say you want to insert in your document a picture called sicilian-summer.jpg, in your latex file you would write:

```
\includegraphics{sicilian-summer.jpg}
```

and that's it. However you probably want to change something, for example the position of the picture, or its size. You can do this by adding options, in square brackets, to the command. For example, if the picture is too big and you want to scale it down a little bit, you could give the following option:

```
\includegraphics[scale=0.75]{sicilian-summer.jpg}
```

and you would get the picture scaled down to 75% of its original size.

Chapter 13

Language Selection

The `babel` package allows to set the default language for the document. This takes care of a series of things, like providing the headings for sections, table of contents, figures, tables etc. . . in the language defined, as well as using the hyphenation patterns that decide how words can break across lines, for that language. Depending on the operating system and \LaTeX distribution you're using, you might need to install some additional software to get full support for your language. The following sections will provide some platform specific information.

13.1 Windows: MiKTeX users

Warning: I have not used \LaTeX on Windows for a while, this section was written around 2004-2005 and may now be obsolete.

In order to load the hyphenation patterns for another language with MikTeX, go to the directory:

```
C:\texmf\tex\generic\config\
```

you'll find a file called `language.dat`, open it with notepad and uncomment the line for your language, by deleting the `%` sign. Then open the miktex options wizard, choose languages and select the language you want, finally return to the tab general, again in the wizard and push the button 'update now' under the section format files.

If that's not enough you could also try to open the MS-DOS command line and type:

```
makefmt latex
```

13.2 GNU/Linux: Debian Based Distributions

On Debian and Debian based GNU/Linux distributions (Ubuntu and friends), the \LaTeX distribution of choice is \TeX Live, in order to get support for your language,

you need to install the T_EXLive language support package for that language, for example for Italian:

```
$ apt-get install texlive-lang-italian
```

13.3 Using Babel

The `babel` package is loaded as any other package, with an option that defines the language to use, so in order to get support for the Italian language you would write in the preamble of your document:

```
\usepackage[italian]{babel}
```

13.4 Accents

L^AT_EX provides a series of control sequences for typesetting accented characters, for example

`\'e` produces é (acute accent)

`\`e` produces è (grave accent)

you can google it to find out more, however if you're writing in a language like Italian or French that constantly require the use of accented characters, a better way to produce them is to use the `inputenc` package. I'll give information only related to the Italian language, because it's the only one I have experience with. In the preamble of your document load the `inputenc` package with

```
\usepackage[ansinew]{inputenc}
```

now you can input the accented characters directly, as they appear in your native Italian keyboard, without needing any control sequence.

Chapter 14

Adding a Bibliography

14.1 BibTeX

The names can be entered in the following format:

Surname, Forenames

if there is something like a 'James von der Vanderbilt' use:

von der Surname, Forenames

Multiple authors should be separated with 'and'

14.1.1 Special Characters in BibTeX

Imagine you have one book in your reference list which contains the word \LaTeX and you want to render it properly as a logo. If in the title section of your BibTeX file, you just write \LaTeX , you're likely to get an irreversible error. This is because, as far as I've understood, some bibliographic styles require that the title contains all lowercase letters, so what your \LaTeX will be transformed to \latex in the bbl file, a control sequence which, as a matter of facts, has no meaning and will cause an error when it will be latexed. To prevent uppercase letters from being transformed into lowercase letters, you need to put them between curly braces, writing for example: $\{America\}$ or $\{A\}merica$. You may use this strategy to use other special characters in BibTeX entry fields. Beware, some special characters may be different under BibTeX one case being just the \LaTeX command, you not only need to put it between curly braces, but also eliminate the second backslash, so to get it right type:

```
{\LaTeX}
```

It has taken me half a day to discover all the above!!

14.2 Apacite

To use the apacite package you need to both load it in the preamble and declare it in the `bibliographystyle{apacite}`. After this to get both author and year, between brackets, for example (Cartwright, 2000), write:

```
\cite{cartwright2000}
```

To get the author(s) in the text and the year between brackets, for example Cartwright (2000) has stated..., use a modified version of the cite command:

```
\citeA{cartwright2000}
```

The command:

```
\citeauthor{cartwright2000}
```

prints only the name of the author, and may be used when you want to refer to an author whose work has been fully referenced previously, in the same paragraph. You may need the citeauthor command along with the citeyear command in other few cases, for example when you have to get something like this Cartwright's (2003) theory.

In the citations you may add prefixes and postfixes with the `<prefix>` and `[postfix]` commands, for instance:

```
\cite<see>[for a different view]{cartwright2000}
```

```
\citeyear{cartwright2000}
```

to get only the year between brackets, for example Cartwright (2000) has stated...

Chapter 15

Adding an Index

To get multiple indexes, for example one index for relevant entries and an author index, you can use the package **index**. For generating an index use the command `\newindex` which has 4 required arguments, for example in order to get an index for relevant entries, in the preamble of the document, use the command:

```
\newindex{default}{idx}{ind}{Name of the Index}
```

the first argument is a tag to identify the index, and the last argument is the title which will appear at the head of the index. The tag default is special: If the commands `\index` and `\printindex` are used without specifying a tag, default will be used.

Then, where you want your index to appear in your document, use the command:

```
\printindex[default]
```

the optional argument contains the index tag and specifies which index should be printed.

An essential step is adding the index entries in your document for the words you want to get indexed. To do this, use the command `\index`. This command works as in the `makeindex` package. So if you want the word `saccades` to get indexed, after it appears in your text write:

```
\index{saccades}
```

Let's say you want to generate a sub-entry for `express saccades`, then you can use the following command:

```
\index{saccades!express}
```

Finally

```
$ latex foo.tex  
$ makeindex foo.idx  
$ latex foo.tex
```

Chapter 16

Adding a Glossary

16.1 The Glossary Package

Installation

Installation of the `glossary` package is a little different than that of other packages, because, besides the usual `.sty` file, a perl script needs to be put in place and made executable. The perl script, that requires of course a functioning Perl installation, is just a script that calls `makeindex` with the appropriate command line switches, and if you like, you can do without it altogether by calling `makeindex` directly.

After having downloaded and unpacked the `glossary` package, you need to run \LaTeX on the `.ins` file:

```
latex glossary.ins
```

and you'll get the following files:

```
glossary.sty
makeglos.pl
glossary.perl
```

the file `glossary.sty` needs to be put in your \LaTeX path:

```
texmf/tex/latex/glossary/glossary.sty
```

the file `makeglos.pl` is the perl script mentioned above, and needs to be put somewhere in your path, for example, for a GNU/Linux installation it can be put in the user's bin directory:

```
/home/user/bin/makeglos.pl
```

where `user` is your username. Further, it needs to be made executable:

```
chmod a+x makeglos.pl
```

The `glossary.perl` is a \LaTeX 2HTML style file and should be placed in the \LaTeX 2HTML styles directory. In my Ubuntu GNU/Linux installation this directory is located in

```
/usr/share/latex2html/styles
```

Setting up the Glossary

The `glossary` package needs to be called in the preamble of your document, followed by a `\makeglossary` command. Finally you need to place the `\printglossary` command where you want the glossary to appear in your document. Below there is an example:

```
\documentclass[a4paper,11pt]{report}
\usepackage[number=none,toc=true,style=list]{glossary}
\makeglossary

\begin{document}
...
\printglossary
\end{document}
```

as you can see the call to the `glossary` package accepts a number of options, which are detailed in the manual.

Entries can be added anywhere in your document through the `\glossary` command, using a list of key-value pairs. An example will make this clear:

```
\glossary{name={LaTeX},
description={a computer typesetting system}}
```

here the list is composed of the `name` key, whose value is the name of the entry, and the `description` key. Other possible key values are `sort`, and `format`. Be careful because no characters, including spaces, other than a newline character may precede a key-value pair, so the following examples would give you an error:

```
%%These will give you errors!!!
\glossary{ name={LaTeX},description={a computer typesetting system}}
%%notice the space before the name key
\glossary{name={LaTeX}, description={a computer typesetting system}}
%%notice the space before the description key
\glossary{name ={LaTeX},description={a computer typesetting system}}
%%notice the space between the name key and the equal sign
```

Once the entries are properly inserted into the document, it can be processed as follows:

```
latex foo.tex
makeglos foo.glo
latex foo.tex
```

Appendix A

Info and Resources for L^AT_EX

- **T_EXMed find Pubmed bibliographic citations directly in BibT_EX format**
<http://www.sbg.bio.ic.ac.uk/~mueller/TeXMed/>

Bibliography

- [1] D. Carlisle. *The longtable package*, 2004. Comes with the longtable package, available from CTAN.
- [2] L. Caucci and M. Spadaccini. *Gestione di figure e tabelle con \LaTeX* , 2005.
- [3] D. B. Dahl. *The xtable package*, 2004. Comes with the xtable package, available from CTAN.
- [4] P. W. Daly. *Graphics and colour with \LaTeX* , 2004.
- [5] S. Fear. *Publication quality tables in \LaTeX* , 2003. Comes with the booktabs package, available from CTAN.
- [6] T. Feuerstack. Introduction to pdftex. *TUGboat*, 23(3/4):329–334, 2002.
- [7] D. M. Jones. *A new implementation of \LaTeX 's indexing commands*, 1995. Comes with the index package, available from CTAN.
- [8] U. Kern. *Extending \LaTeX 's color facilities: the xcolor package*, 2004. Comes with the xcolor package, available from CTAN.
- [9] E. Meijer. *The apacite package*, 2004. Comes with the apacite package, available from CTAN.
- [10] T. Morrison. *Using the Glossary \LaTeX Package. (A Quick-Start)*, 2005. Retrieved from the Internet.
- [11] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *The not so short introduction to $\LaTeX 2\epsilon$* . Version 4.16.
- [12] N. Talbot. *Using \LaTeX to Write a PhD Thesis*, 2004. (Available: <http://theoval.cmp.uea.ac.uk/nlct/latex/thesis/thesis.html>).
- [13] N. L. C. Talbot. *glossary.sty v 2.19: $\LaTeX 2\epsilon$ Package to Assist Generating Glossaries*, 2005. Comes with the glossary package, available from CTAN.
- [14] D. R. Wilkins. *Getting Started with \LaTeX* . 2nd edition.