LaTeX for Word Processor Users
version 1.0.2

Guido Gonzato, Ph.D. <gonzato@sci.univr.it>

Università di Verona (Italy)
Facoltà di Scienze MM. FF. NN.†
February 4, 2002

Abstract

Text processing with LaTeX offers several advantages over word processing. However, it is often hard for beginners to figure out how to perform common tasks and obtain certain features. This manual attempts to ease the transition by drawing comparisons between word processing and LaTeX typesetting. The main word processor features are listed, along with their equivalent LaTeX commands. Lots of examples are provided.

Contents

†Ca' Vignal II, Strada Le Grazie 15, 37134 Verona (Italy)

List of Tables

List of Figures

1 Introduction

First of all, let me state that this is *not* a LaTeX primer! If you're reading this, I assume that you have at least a smattering of LaTeX and its basic commands. In this document, I'll explain why LaTeX is a viable alternative to word processors. One may even think that LaTeX is *better* than any word processor… and this is definitely true, especially if you write complex documents with lots of maths.

Nevertheless, word processors are the "killer app" in modern office automation. They're perceived to be easier than LaTeX as they have a friendly WYSIWYG interface, and the average secretary will learn to use them in a relatively short time. The problem is, these beasts keep growing slow, bloated[1], buggy, expensive, virus ridden, and incompatible with each other.

LaTeX might be a good alternative. It actually is, but if you try to write casual, unstructured documents, LaTeX will rightly stand in the way. You could resort to plain TeX, but you would lose many facilities that LaTeX provides.

To sum up, sometimes you may want to use word processor-like features—but using LaTeX. It would be nice to know how to obtain some effects with LaTeX when you know how to get them with your once-favourite `:-)` word processor.

That's why I wrote this quick reference. As I said, it assumes that you already have a basic understanding of LaTeX; if not, I suggest that you go to `http://www.ctan.org/starter.html` and download "The (Not So) Short Introduction to LaTeX2e" (and possibly "A Gentle Introduction to TeX").

In the following sections, we shall go through the menus and menu items of an imaginary word processor, finding out the corresponding LaTeX way of doing the same work. If you're a LaTeX purist and this approach makes you sick... don't hurt yourself, please stop here.

1.1 Preliminaries

Many word processor features are implemented by the editor; others by standard LaTeX commands; others still are obtained using *packages*. These are sets of macros that extend LaTeX providing new commands and environments. There are lots of packages around: the only problem is knowing where they are, what they do, and how to install them. More about packages in Section 1.1.2.

Packages and other TeX-related material are available at many sites that constitute the CTAN, the Comprehensive TeX Archive Network. I already mentioned `http://www.ctan.org`; there you will find a comprehensive list of mirrors. Get LaTeX for your platform from `CTAN:/systems`.

Another obvious question is this: LaTeX will typeset the text for me, but what should I use to write it? The answer is: a good *editor*—a LaTeX-aware one, if possible. There exist editors dedicated to writing LaTeX source, and some are WYSIWYG or nearly so.

Although any editor capable of saving plain text will do (even Windows' `notepad`), most TeXnicians will recommend that you use one of the following:

- GNU Emacs (`http://www.gnu.org/software/emacs/emacs.html`) or Xemacs (`http://www.xemacs.org`);
- `vi` or one of its clones, Vim (`http://www.vim.org`) being the most popular;
- my personal choice: Jed (`http://space.mit.edu/~davis/jed/`), a light Emacs clone with WordStar/Borland keybindings too;
- LyX (`http://www.lyx.org`), a nearly-WYSIWYG text processor, less advanced as an editor but undoubtedly easier for beginners;
- GNU TeXmacs (`http://www.texmacs.org`), an impressive (if a bit slow) WYSIWYG LaTeX word processor (for Unix only);

[1]once upon a time, I wrote my thesis on a 128k, Z80-based home computer. The word processor WordStar and my thesis fit on a single CP/M-bootable 720K floppy, with lots of room to spare!

- WinEdt (http://www.winedt.com), probably the most popular LaTeX shell for Windows. Unlike all of the free software above, it's shareware;
- a comprehensive list of LaTeX shells for Windows can be found at http://home.arcor.de/itsfd/texwin.htm. I suggest that you have a look at TeXnicCenter and WinShell;
- finally, information about LaTeX on the Mac can be found at http://www.esm.psu.edu/mac-tex/.

These editors sport syntax highligting and many other helpful features that help write LaTeX source code.

1.1.1 Editor-Supported Features

It should be clear by now that LaTeX is just a formatter: things like cut and paste, search and replace etc. are delegated to the editor. Table 1 summarises the main commands of popular editors for geeks: GNU `emacs` and `vim` with their native key bindings, and `jed` configured for Borland IDE key bindings.

1.1.2 Adding Packages

The following applies to `teTeX`, which ships with most GNU/Linux distributions. Directions for MiKTeX below.

1. get the package (tipically as a gzip-compressed tar archive) from your favourite CTAN mirror;
2. find out the location of LaTeX packages; most likely, `$TEXMF/tex/latex`;
3. unpack the package under that directory;
4. If no `.sty` file exists, run the command `latex newstyle.ins` or `latex newstyle.dtx` to create it;
5. run the command `texhash` so that `teTeX` recognises the new package.

The exact location of `$TEXMF` depends on the system; in most GNU/Linux distributions, it's `/usr/share/texmf`.

To add a package to MiKTeX, create the directory `\latex\newpackage` under `C:\localtexmf\tex` and put the relevant files in there. Proceed as above, then run MiKTeX Options and click on the "Refresh now" button. Alternatively, issue the command `initexmf -u`. That's it!

1.2 The Golden Rules

Before we start, please keep this in mind:

1. get used to *structuring* your documents: think in terms of parts, chapters, sections, and so on. This holds even if you aren't writing a scientific paper.

Action	Emacs	Vim	Jed
command mode	n/a	`ESC`	n/a
insert mode	n/a	`i a o O`	n/a
line editor mode	n/a	`:`	n/a
open file	`Ctrl-X Ctrl-F`	`:e`	`Ctrl-KE`
insert file	`Ctrl-Xi`	`:r`	`Ctrl-KR`
save file	`Ctrl-X Ctrl-S`	`:w`	`Ctrl-KD`
save as	`Ctrl-X Ctrl-W name`	`:w name`	`Ctrl-KS`
close file	`Ctrl-XK`	`:q`	`Ctrl-KQ`
change buffer	`Ctrl-XB`	`bN`	`Ctrl-KN`
undo	`Ctrl-XU`	`u`	`Ctrl-U`
redo	`Ctrl-^`	`Ctrl-R`	`Ctrl-G Ctrl-U`
exit	`Ctrl-X Ctrl-C`	`:qa!`	`Ctrl-KX`
word left	`Alt-B`	`b`	`Ctrl-A`
word right	`Alt-F`	`w`	`Ctrl-F`
start of line	`Ctrl-A`	`0`	`Ctrl-QS`
end of line	`Ctrl-E`	`$`	`Ctrl-QD`
page up	`Alt-V`	`Ctrl-U`	`Ctrl-R`
page down	`Ctrl-V`	`Ctrl-D`	`Ctrl-C`
start of buffer	`Alt-<`	`1G`	`Ctrl-QR`
end of buffer	`Alt->`	`G`	`Ctrl-QC`
line n.	`Alt-G n.`	`n.G`	`Ctrl-QI`
character left	`Ctrl-H`	`X`	`BS`
character right	`Ctrl-D`	`x`	`Alt-G`
word left	`Alt-DEL`	`db`	`Alt-BS`
word right	`Alt-D`	`dw`	`Ctrl-T`
line	`Ctrl-A Ctrl-K`	`dd`	`Ctrl-Y`
search	`Ctrl-S text`	`/text`	`Ctrl-QS`
replace	`Alt-%`	`:s/old/new/g`	`Ctrl-QA`
start selection	`Ctrl-SPACE`	`v`	`Ctrl-KB`
cut	`Ctrl-W`	`D`	`Ctrl-KY`
copy	`Alt-W`	`Y`	`Ctrl-KH`
paste	`Ctrl-Y`	`P`	`Ctrl-KC`

Table 1: Common key bindings for Emacs, Vim, and Jed in IDE mode.

2. LaTeX explicitly discourages the user from messing about with formatting parameters. Don't worry too much about the *looks* of what you're writing, concentrate on the *contents*.

Convince yourself, apply the rules, and your printed material will magically look professional! Again, let me recommend that you study a real LaTeX primer.

These rules apply to most documents. However, if the stuff you are going to write is intrinsically unstructured (circulars, notices, etc.), then you asked for that: go ahead and read the rest of this guide.

2 The File Menu

Obviously, some entries of this menu have nothing to do with LaTeX: File/Open, File/Save, File/Close depend on the editor.

2.1 File/New

This is the LaTeX equivalent of a blank page:

```
\documentclass{article}
\begin{document}
% This is a comment. Write your stuff here.
\end{document}
```

As documents written in LaTeX are inherently structured, this is a more realistic example:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
\title{My Document}
\author{John Smith}
\date{London, \today}
\maketitle
\begin{abstract}
This is a very short article.
\end{abstract}
\tableofcontents
\listoftables
\listoffigures
\section{First Section}
\label{sec:start}
This is the text of the section. See \cite{Gonzato} for details.
\section{End}
\label{sec:end}
This is the end of the document. Please go to Section
\ref{sec:start} to read it again.
\begin{thebibliography}{99}
\bibitem{Gonzato} Gonzato G. \textit{\LaTeX{} for Word Processor
```

```
Users}. CTAN, 2001.
\end{thebibliography}
\end{document}
```

More document templates are shown in Appendix A.

2.2 File/Save As. . .

The following tools are handy if you want to convert LaTeX to other formats:

- `latex2html`: http://saftsack.fs.uni-bayreuth.de/~latex2ht/, CTAN://tex/support/latex2html
- `latex2rtf`: CTAN://tex/support/latex2rtf

2.3 File/Save As Template

Saving a LaTeX "template" would mean, as I see it, create a new LaTeX package. This is complex matter that goes beyond the scopes of this guide.

2.4 File/Import

These tools convert from other formats to LaTeX:

- `rtf2latex`: CTAN://tex/support/rtf2latex
- `html2latex`: CTAN://tex/support/html2latex
- `wvware` converts from MS Word to several formats including LaTeX; available at http://www.wvware.com

Other `*2latex` converters are available at the same address.

2.5 File/Page Setup

The normal method for setting paper size, orientation, and margins, is to use parameters in `\documentclass`. Paper size can be `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`; orientation is portrait by default, or optionally `landscape`. For example,

```
\documentclass[a5paper,landscape,12pt]{article}
```

Document-wide margins are set with these commands:

```
\setlength{\leftmargin}{2cm}
\setlength{\rightmargin}{2cm}
\setlength{\oddsidemargin}{2cm}
\setlength{\evensidemargin}{2cm}
\setlength{\topmargin}{-1cm}
\setlength{\textwidth}{18cm}
\setlength{\textheight}{25cm}
```

The geometry package allows for complete control of parameters such as paper size, margins width, and so on. geometry has far too many options to list them all, and you're invited to read its documentation. A fairly complete example of its usage is shown below. In this example, some parameters are not compatible with each other, and are there for purpose of argument only.

```
\geometry{paperwidth=25cm}
\geometry{paperheigth=35cm}
% or: \geometry{papersize={25cm,35cm}}
\geometry{width=20cm}  % total width
\geometry{heigth=30cm} % total heigth
% or: \geometry{total={20cm,30cm}}
\geometry{textwidth=18cm}  % width - marginpar
\geometry{textheigth=25cm} % heigth - header - footer
% or: \geometry{body={18cm,25cm}}
\geometry{left=3cm}    % left margin
\geometry{rigth=1.5cm} % right margin
% or: \geometry{hmargin={3cm,2cm}}
\geometry{top=2cm}     % top margin
\geometry{bottom=3cm}  % bottom margin
% or: \geometry{vmargin={2cm,3cm}}
\geometry{marginparwidth=2cm}
\geometry{head=1cm}    % header space
```

Options can also be set thus:

```
\usepackage[left=3cm, right=2cm]{geometry}
```

2.5.1 Page Setup/Headers and Footers

The fancyhdr package provides the new command `\pagestyle{fancy}`. This will create a header reporting the current section (chapter in book.cls) and subsection, and a footer with the page number: pretty fancy indeed. Headers and footers can obviously be customised. They are made of three parts: a left-aligned part, a centre-aligned part, and a right-aligned part. To set those, use the commands like in this example:

```
\lhead{} % empty
\chead{Hello, world!}
\rhead{Page \thepage} % page number
\lfoot{}
\cfoot{\textbf{Hello!}}
\rfoot{}
```

2.6 File/Printer Setup

This is highly OS-dependent, and definitely outside of the scopes of LaTeX. Assuming that you're using a UNIX-like system, these tips may come in handy:

- `lpr -P printername` prints to the specified printer;

- `lpr -# 10` prints 10 copies;
- `lpr -r` removes the file after printing it

Other tips will come soon.

2.7 File/Print Preview

When the LaTeX input file is ready, you have several options:

- convert it to `.dvi` (`latex file.tex`) and previewing it with `xdvi`, `yap` or other;
- convert the `.dvi` to PostScript with `dvips`, then use some variant of `Ghostview`;
- producing an Acrobat `.pdf` file with `dvipdf` or directly with `pdflatex`.

In my opinion, producing a `.pdf` file is the best option, as it makes for widest portability. Moreover, the hyperref package used together with `pdflatex` makes the resulting `.pdf` file browseable; see Section 4.15. However, using `pdflatex` you might experience problems with some packages. *Warning* More details in Section 4.12.

2.8 File/Print

Simply use `lpr file.ps` from the command line of via your previewer's menu. If you create Acrobat `.pdf` files, your file will gain more visibility, especially from the point of view of Windows and Mac users.

2.9 File/Versions

Again, this is nothing LaTeX is aware of. Under Unix and Win32, you use external tools like RCS (Revision Control System) or CVS (Concurrent Version Control System), possibly integrated with the editor. For a quick introduction to CVS and RCS, please visit `http://www.linuxdoc.org/HOWTO/CVS-RCS-HOWTO.html`.

[TODO: mention version.]

3 The Edit Menu

This menu has more to do with the editor than LaTeX features. To begin with, refer to Table 1 that shows the key bindings for the entries Edit/Cut, Edit/Copy, Edit/Paste, Edit/Find, and Edit/Replace of some common editors.

The *selection* of text is not only performed for subsequent cut and paste, but also for applying styles to the selected text. The equivalent action in LaTeX is to enclose the text either in braces, or in an environment. For example, to apply the bold attribute to a portion of text you will use one of the following:

```
this is \textbf{bold text;}\\
this is also
{\bfseries bold text;}\\
\begin{bfseries}
this is bold text, too!
\end{bfseries}
```

this is **bold text;**
this is also **bold text;**
this is bold text, too!

3.1 Edit/Autotext

We call *autotext* the feature that makes you type, say, “PS” to have the word “PostScript” automatically expanded. This is an editor feature, but there’s a rough LaTeX equivalent:

```
\def\PS    {\textsc{PostScript}}
```

which will insert the equivalent of `\textsc{PostScript}` whenever you type `\PS`. Beware, case is important.

4 The Insert Menu

4.1 Insert/Breaks

- a non-breakable space is set with ~
- to force a line break, use `\linebreak` or `\newline`; see below for details
- to start a new paragraph, use a blank line, or `\\`, or `\par`
- finally, for a page break use `\newpage` or `\clearpage`

`\linebreak` stretches the line to the margins, `\newline` does not:

```
I am stretched!\linebreak
But I am not.\newline
Ok, now you get it.
```

I am stretched!
But I am not.
Ok, now you get it.

Finally, `\clearpage` differs from `\newpage` in that it prints all pending floats (figures and tables). Floats will be explained in Section 4.12.

4.2 Insert/Enumerated List

Bulleted and enumerated lists correspond to the `itemize` and `enumerate` environments. To customize the bullet in a list environment, you can specify it as an argument to `\item`:

```
\begin{itemize}
  \item[*] with an asterisk;
  \item[-] with a dash.
\end{itemize}
```

* with an asterisk;

\- with a dash.

Another way is to redefine the style of the counters that refer to the digits in the first- to fourth-level lists[2]. There are several styles: `\arabic` for "normal" numbers, `\roman` for lower-case roman numerals (e.g. viii is 8), `\Roman` for upper-case roman numerals, `\alph` and `\Alph` for lower- and upper-case letters, and `\fnsymbol` that I'll not explain for now:

```
\begin{itemize}
\renewcommand{\labelitemi}{*}
\renewcommand{\labelitemii}{-}
  \item first level, item 1
  \item first level, item 2
  \begin{itemize}
    \item second level, item 1
    \item second level, item 2
  \end{itemize}
  \item first level, item 3
\end{itemize}
```

* first level, item 1

* first level, item 2

  - second level, item 1
  - second level, item 2

* first level, item 3

To use, say, roman numbers and uppercase letters in enumerated lists, do this:

```
\begin{enumerate}
\renewcommand{\labelenumi}
  {\Alph{enumi}}
\renewcommand{\labelenumii}
  {\roman{enumii}}
  \item first level, item 1
  \item first level, item 2
  \begin{enumerate}
    \item second level, item 1
    \item second level, item 2
  \end{enumerate}
  \item first level, item 3
\end{enumerate}
```

A first level, item 1

B first level, item 2

  i second level, item 1

  ii second level, item 2

C first level, item 3

[TODO: mention the enumerate package.]

4.3 Insert/Special Character

First of all, let's remind that some characters have a special meaning in LaTeX and must be entered either by preceding them with \, or using them in math mode, or even writing special commands: see Table 2.

If you want really weird characters, the pifont package provides you with the command `\ding` (like `ding{100}` that produces ❄), and `\dingfill`, `\dingline`, and `\dinglist`. These are equivalent to the `\fill`, `\line`, and `\list` commands and environment, but use the Dingbat code given as parameter:

```
\begin{dinglist}{43}
\item one
\item two
\item three
\end{dinglist}
```

☞ one

☞ two

☞ three

[2]These are `\labelitemi`, `\labelitemii`, `\labelitemiii` and `\labelitemiv`.

Character	**Character Sequence**
$	`\$` or `\textdollar`
&	`\&`
%	`\%`
_	`\_` or `\textunderscore`
{	`\{` or `\textbraceleft`
}	`\}` or `\textbraceright`
<	`$<$` or `\textless`
>	`$>$` or `\textgreater`
\	`\textbackslash`
|	`\textbar`
•	`\textbullet`
‡	`\textdaggerdbl`
†	`\textdagger`
¶	`\textparagraph`
§	`\textsection`
©	`\textcopyright`
ˆ	`\textasciicircum`
˜	`\textasciitilde`
$\sim$	`$\sim$`
®	`\textregistered`
™	`\texttrademark`
ª	`\textordfeminine`
º	`\textordmasculine`

Table 2: How to obtain some special characters.

❄	\Snowflake
✉	\Letter
⚡	\Lightning
📱	\Mobilefone
☕	\Coffeecup
⚽	\Football
✂	\Cutright
FAX	\FAX
📠	\Faxmachine
♡	\Heart
☺	\Smiley
🚲	\Bicycle
🚶	\Womanside
🚺	\Womanfront
♑	\Capricorn
🦇	\Bat
👩	\Womanface

Table 3: Some marvosym characters.

Another nice variant is this:

```
\begin{dingautolist}{172}
\item one
\item two
\item three
\end{dingautolist}
```

① one

② two

③ three

4.3.1 The € Sign

The marvosym package provides several fine characters, among which the Euro sign €. You obtain it with `\EUR`.

Other mouth-watering characters are shown in Table 3. Look at marvosym's documentation for a complete list of characters; my favourite is ☯ (`\Yingyang`).

[TODO: add the Dingbat table.]

4.4 Insert/Formula

LaTeX is particularly strong at doing math. To insert math symbols in the main text, you must enclose them in `$`:

```
I like math: $x^n + x^n \neq
z^n \forall n \neq 2$
is my favourite theorem.
```

I like math: $x^n + x^n \neq z^n \forall n \neq 2$ is my favourite theorem.

The environments `displaymath` and `equation` typeset formulae aside from the text. The latter adds an equation number for later reference:

```
Fermat's Last Theorem is
defined as:
\begin{equation}
x^n + x^n \neq z^n
\forall n \neq 2
\label{eq:fermat}
\end{equation}
Can you prove
Eq.~\ref{eq:fermat}?
```

Fermat's Last Theorem is defined as:

$$x^n + x^n \neq z^n \forall n \neq 2 \tag{1}$$

Can you prove Eq. 1?

4.5 Insert/Footnote

The command `\footnote[n]{footnote text}` is all you need; the optional parameter `[n]` modifies the footnote number. To use a symbol instead of a number, redefine the counter associated with `\footnote`:

```
\renewcommand{\thefootnote}{foot}
This footnote\footnote
{I mean this one.}
says it all.
```

This footnote[foot] says it all.

[foot] I mean this one.

For example, you can get footnote numbers in roman numerals, or replaced by nice symbols:

```
\renewcommand{\thefootnote}
{\Roman{footnote}}
This\footnote{the first.}
is the first footnote,
and this\footnote{the second.}
is the second.
\renewcommand{\thefootnote}
{\fnsymbol{footnote}}
The end.\footnote[8]{At last!}
```

This[I] is the first footnote, and this[II] is the second. The end.[††]

[I] the first.
[II] the second.
[††] At last!

Note the `\fnsymbol{footnote}` thing. It uses 9 symbols associated with the values 1...9 of the `footnote` counter: $* \; \dagger \; \ddagger \; \S \; \P \; \| \; ** \; \dagger\dagger \; \ddagger\ddagger$.

To make several references to the same footnote, don't write its number explicitly. Rather, do this:

```
This\footnote{the first.}
\newcounter{\myfootnote}
\setcounter{\myfootnote}
  {\value{footnote}}
and that\footnote{the second.}
are footnotes: please read note
\footnotemark
[\value{\myfootnote}] again.
```

This[1] and this[2] are footnotes: please read note[1] again.

[1] the first.
[2] the second.

4.6 Insert/Indices

Generating and inserting a table of contents, list of tables, and list of figures is a trivial task in LaTeX. All you have to do is insert these lines before the first `\section` or `\chapter` of your document:

```
\tableofcontents
\listoffigures
\listoftables
```

4.7 Insert/Vertical and Horizontal Space

This entry doesn't actually exist in any word processor I know. This is in fact a limitation that LaTeX fills in a very elegant way.

Space filling is used to center text horizontally, vertically, or both; this is a difficult task to perform with any word processor, and requires a lot of trial end error. Use a combination of `\null` or `~` to set fixed marks, followed by `\vfill` and `\hfill` like in this example:

```
one \hfill two\\
\vfill
~ \hfill three \hfill ~\\
\vfill
four \hfill five
\null
```

one		two
	three	
four		five

Normally, LaTeX won't let you insert blank spaces at your will. However, if you do want to make your document look messy, use `~` to make a non-breakable space.

Also, use the command `\hspace` like in this example:

```
This is a \hspace{2cm}
2-cm-wide hole.
```

This is a   2-cm-wide hole.

4.8 Insert/Tabs

The `tabbing` environment provides a rough equivalent to the action of the TAB key. See this example:

```
% let's set the tab positions
~ \hskip 1cm \= ~ \hskip 2cm \=
~ \hskip 3cm \= \kill % discard text
Zero \> One \> Two \> Three \\
Zero \> One \> \> Three \+ \\ % go right
Zero \> Two \> Three \- \\ % go left
Zero \> One \> Two \\
\pushtabs % save tab positions
new tab 1{\ldots} \= new tab 2 \\
new \> tab \\
\poptabs  % restore tab positions
Zero \> One \> Two \> Three \\
```

Zero	One	Two	Three
Zero	One		Three
	Zero	Two	Three
Zero	One	Two	
new tab 1…	new tab 2		
new	tab		
Zero	One	Two	Three

See also the `tabular` and `table` environments.

4.9 Insert/Cross Reference

The commands `\label`, `\ref`, and `\pageref` are all you need to insert labels in the text and do cross referencing. Labels usually have the `prefix:suffix` form, where `prefix` is one of the following: `cha` for chapters, `eq` for equations, `fig` for figures, `sec` for (sub)sections, and `tab` for tables.

References to a page (section, table, figure, etc) number are done using `\label` and `\ref` as in this example:

```
\paragraph{Example.}
\label{par:example}
This paragraph appears
in Section~\ref{par:example}
on page \pageref{par:example}.
```

Example. This paragraph appears in Section 4.9 on page 15.

Of course, you may use your own prefixes. For example, take this enumerated list:

```
\begin{enumerate}
  \item{first step: skip to
  \ref{item:end} \label{item:start}}
  \item{another step (unreferenced)}
  \item{end: go back to
  \ref{item:start} \label{item:end}}
\end{enumerate}
```

1. first step: skip to 3
2. another step (unreferenced)
3. end: go back to 1

4.10 Insert/Margin Notes

Very simple: use `\marginpar{text}`.

4.11 Insert/Frame

Let's imagine you want to typeset a poster: you'll need a way to put text and figures on fixed position on the page. Use the `textpos` package, as shown in the example listed in Figure 7 (see Appendix A).

4.12 Insert/Figure

(The reference guide for graphic inclusion in LaTeX is "Using Imported Graphics in LaTeX2e", a.k.a. `epslatex.ps`.)

A "figure" can be not only a picture, but also a portion of text, a table, etc. that you enclose in a `figure` environment. This is an example:

```
\begin{figure}[htbp]
% [htbp] specifies the
% preferred placement: here, top,
% bottom, or separate page.
  \begin{center}
  \texttt{=8-)}
  \end{center}
  \caption{A smiley representing
  the author of this guide.}
  \label{fig:mysmiley}
\end{figure}
```

`=8-)`

Figure 1: A smiley representing the author of this guide.

Please note that figures are not guaranteed to appear exactly where you write the code! In fact, the main difference with word processors is that figures don't have a fixed placement; they "float" to the optimal position that LaTeX finds for them. So, the text shouldn't refer to a figure like "the figure below" or "the figure above"; rather, use "`see Figure~\ref{fig:label}`". *Note*

Owing to this property, figures and tables are called *floats*. If you do need to position a float exactly, use the `here` package that provides an optional placement argument `H`.

Given a picture in Encapsulated PostScript (`.eps`) format, you insert it in a LaTeX source file using the `graphicx` package and commands like those shown in Figure 2.

```
\begin{figure}
\begin{center}
\fbox{\includegraphics
[width=0.5\textwidth, angle=-90]
{gnuplot.ps}}
\caption{A Gnuplot graph.}
\label{fig:gnuplot}
\end{center}
\end{figure}
```

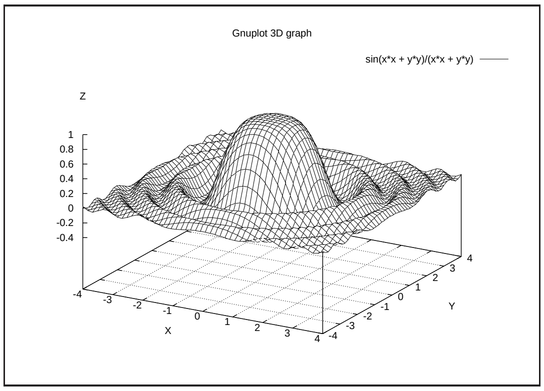

Figure 2: A Gnuplot graph.

Graphic inclusion only works with `.eps`. There are several packages that convert common graphic formats like `.jpg`, `.gif`, `.png` etc. to `.eps`: one of the finest is ImageMagik (http://www.imagemagik.org), freely available for many platforms. Besides, there's a utility that wraps `.jpg` files producing very compact `.eps` files: it's called `jpeg2ps` (http://www.pdflib.com/jpeg2ps/index.html).

If you use PDFLaTeX, you must convert your pictures to `.pdf` with the command `epstopdf` and modify the source accordingly! If you wish to make both `.pdf` and `.ps` from the same source file, include these commands: *Warning*

```
% define the variable \ifpdf
\newif\ifpdf
\ifx\pdfoutput\undefined
  \pdffalse
\else
  \pdfoutput=1
  \pdftrue
\fi
...
% include the right options
\ifpdf
  \usepackage[pdftex]{graphicx}
  \pdfcompresslevel=9
\else
  \usepackage{graphicx}
\fi
```

```
...
% include the right graphic file
\ifpdf
  \includegraphics{file.pdf}
\else
  \includegraphics{file.eps}
\fi
```

[TODO: mention wrapfig]

If you have more than 18 figures without text between them, you'll get the "Too many unprocessed floats" LaTeX error. The quickest solution to this problem[3] is to put `\clearpage` after three–four figures. *Warning*

4.13 Insert/Shapes

There is a `picture` environment whithin which you use commands like `\circle`, `\oval` and so on. In my opinion, drawing pictures without a graphical environment is just too hard, and `picture` has several limitations too. It's much better to use the epic and eepic packages (include them in that order), together with the drawing program Xfig (http://www.xfig.org). It's only available for UNIX.

Xfig looks ugly, but is very powerful. One of its greatest advantages is that it exports drawings in several formats, among which eepic macros; these, in turn, make very compact POSTSCRIPT files. Another bonus is that text objects will be rendered by LaTeX if their "special flag" field is set, letting you enter ordinary LaTeX formulae and symbols in drawings.

So, let's suppose you made a drawing called `small.fig`. Select File/Export... and choose "LaTeX picture + eepic macros" from the Language menu. You'll obtain another file, `small.eepic`. To include the drawing in a document:

```
This is a picture
drawn with Xfig:\\
\input{small.eepic}
```

This is a picture drawn with Xfig:

$e^{i\pi} = -1$

Unfortunately, eepic doesn't work with `pdflatex`. As an alternative, you should export the drawing as "Combined PS/LaTeX (both parts)". You'll get two files, `small.pstex` and `small.pstex_t`. Rename the latter to `small.tex` and edit it to insert these lines at the top:

```
\documentclass{article}
\usepackage{graphicx}
\pagestyle{empty}
\begin{document}
```

and this line at the bottom:

[3] not necessarily the best solution, though.

```
\end{document}
```

Compile it as usual, convert the `.ps` to `.eps` and include it as explained in 4.12.
[TODO: mention pstrick and give some examples.]

4.14 Insert/Line

Draw lines of any lenght and thickness with `\rule`:

```
This is a page-wide
rule:\\
\rule{\linewidth}{1pt}
but this one is shorter
and thicker:\\
\rule{2cm}{2mm}
```

This is a page-wide rule:

but this one is shorter and thicker:

Another interesting "line" is that made of dots (`\dotfill`), often used to relate things. This is how it's done:

```
Total price \dotfill \EUR~10
```

Total price .... € 10

4.15 Insert/Hyperlink

The url package lets you write URLs and have them hyphenated correctly. Used together with the package hyperref and `dvipdf` or `pdflatex`, url lets you make browseable `.pdf` documents! For example, this document uses this declaration:

```
\usepackage[colorlinks,urlcolor=blue]{hyperref}
```

4.16 Insert/Comment

This is done inserting % before each line, or by using the package comment that provides the environment of the same name.

5 The Format Menu

In general, the main format properties of a document are set with parameters in `\documentclass`: default font size (10, 11, or 12pt), paper (`a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`), and orientation (`portrait`, `landscape`). For example,

```
\documentclass[a5paper,landscape,12pt]{article}
```

Alternative font sizes can be specified as explained in Section 5.2.1.

5.1 Format/Line Spacing

The TeX command `\baselinestretch{`*amount*`}` alters the default spacing between lines. For example,

```
\baselinestretch 7 mm
```

Alternatively, use the package doublespace or setspace (they are equivalent) that provides the environments `singlespace`, `onehalfspace`, and `doublespace`.

5.2 Format/Character

Standard character properties are listed in Table 4, font sizes in Table 5.

Text attribute	**Environment form**	**Example**
`\textnormal`	`textnormal`	main document font
`\textrm`	`rmfamily`	roman
`\textit`	`itshape`	*italics*
`\emph`	n/a	*emphasis*
`\textmd`	`mdseries`	medium weight (default)
`\textbf`	`bfseries`	**boldface**
`\textup`	`upshape`	upright (default)
`\textsl`	`slshape`	*slanted*
`\textsf`	`sffamily`	sans serif
`\textsc`	`scshape`	SMALL CAPS
`\texttt`	`ttfamily`	`typewriter`
`\underline`	`underline`	$\underline{\text{underline}}$
`\textsuperscript`	n/a	this is $^{\text{superscript}}$
`\mathrm`	n/a	$\mathrm{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathbf`	n/a	$\mathbf{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathsf`	n/a	$\mathsf{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathtt`	n/a	$\mathtt{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathit`	n/a	$\mathit{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathnormal`	n/a	$\mathnormal{x^n + x^n \neq z^n \forall n \neq 2}$
`\mathcal`	n/a	$\S^{\backslash} + \S^{\backslash} \neq \ddagger^{\backslash}\forall\backslash \neq \in$

Table 4: Font attributes.

Please note the difference between italics and emphasised text. *For example, this portion of text is typeset in italics, and* these words *are emphasised in upright.* As you can see, `\emph` is a *logical* command rather than a strictly typographic one.

Also, please note that underscript is normally used in math mode only. The trick to use it in normal text is this:

```
this is
$_{\mbox{\footnotesize{subscript}}}$
```

this is $_{\text{subscript}}$

Font size	Example
tiny	sample text
scriptsize	sample text
footnotesize	sample text
small	sample text
normalsize	sample text
large	sample text
Large	sample text
LARGE	sample text
huge	sample text
Huge	sample text

Table 5: Font sizes

5.2.1 Format/Character Size

If the standard font sizes aren't enough for you, the package extsizes may be handy. It provides "extended" versions of the standard document classes, with support for sizes 8—12, 14, 17, and 20 pt.

For example, let's suppose you want to typeset an article using a 17 pt font. You'll use this document preamble:

```
\documentclass[17pt]{extarticle}
```

Another way to get big fonts is to use the package type1cm, which provides commands like the following:

```
\fontsize{72pt}{72pt}\selectfont
No Smoking
```

Parameters are font size and baseline. Yet another approach is this:

```
\resizebox{!}{1cm}{1-cm tall}
```

1-cm tall

5.2.2 Format/Character Font

LaTeX uses its own fonts, automatically generated when needed by the Metafont subsystem. This ensures portability and yields very good results. However, many of us are accustomed to other fonts: Times, Helvetica, Sans Serif...

Fortunately, LaTeX can use PostScript fonts. Try using one of the following packages: avant, avangar, bookman, chancery, charter, courier, helvet, helvetic, ncntrsbk, newcent, palatcm, palatino, pifont, times, utopia, zapfchan. Insert `\usepackage{times}` and enjoy the result, but beware: LaTeX handling of maths wasn't designed for PostScript fonts!

Family	Name
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 6: Common font families.

The packages above set the font for the whole document. To use a PostScript font for a region of text only, specify the font family like in the example below. Common font families are listed in Table 6. Beware, some font shapes may be unavailable on some systems! *Warning*

```
This is Computer Modern Roman,
{\fontfamily{phv}\selectfont
this is Helvetica!}
```

This is Computer Modern Roman, this is Helvetica!

Yet another possibility is replacing a standard LaTeX font with a PostScript one: for example, you may want to use Avantgarde whenever Computer Modern Sans Serif would appear. These commands can be renewed as in the example below:

- `\rmdefault` (roman)
- `\sfdefault` (sans serif)
- `\ttdefault` (typewriter)
- `\bfdefault` (boldface)
- `\mddefault` (medium)
- `\itdefault` (italics)
- `\sldefault` (slanted)
- `\scdefault` (small caps)
- `\updefault` (upright)

```
 % Avantgarde replaces sans serif
\renewcommand{\sfdefault}{pag}
```

5.2.3 Format/Character Colour

You can colour words using the package color and appropriate commands. Predefined colours are black, white, red, green, blue, cyan, magenta, and yellow; you can also define your own.

```
\textcolor{red}{This is red.}\\
\color{blue}
This text is blue!\\
So is this. Let's change.\\
\definecolor{mygreen}
{rgb}{0.1,1,0.1}
\color{mygreen}
This is my shade of green!\\
\color{black}
\colorbox{cyan}{A cyan box}\\
\fcolorbox{blue}{green}
{A green box in a blue frame}
```

This is red.
This text is blue!
So is this. Let's change.
This is my shade of green!
A cyan box
A green box in a blue frame

Moreover, the command `\pagecolor` lets you specify… guess what?

5.3 Format/Paragraph

Let's remind what a paragraph is according to LaTeX: a portion of text that either ends with `\\`, or is followed by a blank line.

environments are LaTeX's way of specifying properties like text alignment or font selection for a given portion of text. It's like selecting text with the mouse, then choosing the property you wish from a menu or clicking on a button. Another way is to enclose the text in brackets.

Environments have this general form:

```
\begin{environment}
...text goes here...
\end{environment}
```

For example, if you want to center a paragraph you'll use the `center` environment:

```
\begin{center}
this text is centered
\end{center}
```

this text is centered

Standard environments are listed in Table 7. In the following sections, I'll show you what to use and when.

5.3.1 Format/Paragraph Horizontal Alignment

By default, the text is justified. To get left–aligned, right–aligned or centered text, use the `flushleft`, `flushright` and `center` environments. The commands `\raggedleft`, `\raggedright`, and `\centering` are equivalent to their correspondent environments, but they do not start a new paragraph.

Environment	**Purpose**
`array`	Math arrays
`center`	Centered lines
`description`	Labelled lists
`enumerate`	Numbered lists
`eqnarray`	Sequence of aligned equations
`equation`	Displayed equation
`figure`	Floating figures
`flushleft`	Flushed left lines
`flushright`	Flushed right lines
`itemize`	Bulleted lists
`letter`	Letters
`list`	Generic list environment
`minipage`	Miniature page
`picture`	Picture with text, arrows, lines and circles
`quotation`	Indented environment with paragraph indentation
`quote`	Indented environment with no paragraph indentation
`tabbing`	Align text arbitrarily
`table`	Floating tables
`tabular`	Align text in columns
`thebibliography`	Bibliography or reference list
`theorem`	Theorems, lemmas, etc
`titlepage`	For hand crafted title pages
`verbatim`	Simulating typed input
`verse`	For poetry and other things

Table 7: Standard LaTeX environments.

5.3.2 Format/Paragraph Vertical Alignment

The way paragraphs are separated is often puzzling to word processor users. *Empty lines and multiple spaces are treated like a single empty line or space.* The commands `\smallskip`, `\medskip`, and `\bigskip` provide a bit of space between paragraphs. Warning

If you need more space, use the command `\vskip`{*parameter*} like in this example:

```
These paragraphs will be
separated by 1.3 cm:\\
\vskip 1.3cm
there is a 1.3 cm gap above me.
```

These paragraphs will be separated by 1.3 cm:

there is a 1.3 cm gap above me.

Note that `\vskip` only works between paragraphs. What if you wanted to start a page after an additional margin of, say, 1.5 cm? You'll have to set a "mark" in the text:

```
\null
\vskip 1.3 cm
This text comes after 1.3 cm...
```

This text comes after 1.3 cm...

Finally, the command `\vfill` is used to add empty lines between two paragraphs so that the second paragraph goes exactly to the bottom of the page. For example,

```
This appears at the top of
the page{\ldots}
\vfill
{\ldots}and this at the bottom.
```

This appears at the top of the page…

…and this at the bottom.

5.3.3 Format/Paragraph Margins

Normally, the margins are set for the whole document as seen in Section 2.5. Redefining them for a section of text will not work: if you want to set a paragraph's margins, you'll have to create a new environment like in the following example:

```
\newenvironment{margins}[2]
{
  \begin{list}{} {
    \setlength{\leftmargin}{#1}
    \setlength{\rightmargin}{#2}
  } \item }
{\end{list}}
```

Then you will use the new environment:

```
As you can see, this paragraph
has normal margins.
\begin{margins}{0.5cm}{1cm}
 But please note that this
 paragraph has custom margins.
\end{margins}
```

As you can see, this paragraph has normal margins.

But please note that this paragraph has custom margins.

5.3.4 Format/Paragraph Indentation

To set the amount of indentation of the first line of a paragraph, use the command `\parindent`. In the following example, we set a 1-cm indentation:

```
\setlength{\parindent}{1cm}
```

The commands `\indent` and `\noindent` allow/disallow indentation on the following paragraph. Finally, the distance between paragraphs is set by `\parskip`:

```
\setlength{\parskip}{3pt}
```

5.4 Format/Paragraph Border and Shade

To get framed (bordered) paragraphs or words, you have the choice of using the framed package or the `\parbox` command. The package calc is required in the latter case.

This is the simplest method, using framed:

```
\setlength{\FrameRule}{2pt}
\setlength{\FrameSep}{5pt}
\begin{framed}
  this is a framed paragraph!
\end{framed}
\definecolor{shadecolor}{rgb}
{0.9,0.8,1}
\begin{shaded}
  this is a shaded paragraph,
  do you like it?
\end{shaded}
```

this is a framed paragraph!

this is a shaded paragraph, do you like it?

Equivalently, use the boxedminipage package and the equally named environment. For those who want to know more: the commands

```
\framebox{
  \begin{minipage}[c]{\linewidth}
  text to be framed
  \end{minipage}
}
```

are functionally equivalent to the `boxedminipage` environment.

This example uses `\parbox`:

```
\noindent
\fbox{
  \parbox{\linewidth
    -2 \fboxsep -2 \fboxrule}
  {again, a framed paragraph!}
}
```

again, a framed paragraph!

`\linewidth` sets the width of the minipage equal to that ot the remaining text. Obviously, you can specify the width as you like.

Finally, to frame something adapting the frame to the width of the text:

```
this is a
\framebox[\width]{framed}
word
```

this is a framed word

Modifying the parameter, you can adjust the frame width:

```
this is another
\framebox[2\width][r]{framed}
word
```

this is another framed word

Note that the second optional parameter specifies the alignment (to the right in this example).

5.5 Format/Colour

Now that you have a bordered paragraph, you'll want to set its colour too. Do this:

```
\colorbox{yellow}{
  \begin{minipage}
  {0.8\linewidth}
  I am a minipage, my colour
  is yellow!
  \end{minipage}
}
```

I am a minipage, my colour is yellow!

More about colours in Section 5.2.3.

5.6 Format/Columns

The commands `\twocolumn` and `\onecolumn` start a new page and set the number of columns; they can also be used as parameters in `\documentclass`. If this is not enough for you, the package multicols provides an environment of the same name. I could have set this section in two columns with these commands:

```
\columnseprule=1pt
\begin{multicols}{2}[\subsection{\entry{Format}{Columns}}]
The commands \cmd{twocolumn} ...
\end{multicols}
```

The space between columns is controlled by the parameter `\columnsep`, and the thickness of the rule between columns by `\columnseprule`. The text given as optional parameter in brackets is excluded from the environment.

6 The Table Menu

Quite a complex subject... This is the general format of a table:

```
\begin{table}[htbp] % placement: here, top, bottom, separate page
% \begin{small}     % sets the table font
\begin{center}      % optional
% 4-column table; alignment is left, centered, right, fixed width
\begin{tabular}{|l|c|r|p{4cm}|}
\hline              % horizontal line
\textbf{Left} & \textbf{Centre} & \textbf{Right} & \textbf{4 cm} \\
\hline
row 1, col 1 & row 1, col 2 & row 1, col 3 & row 1, col 4 \\
\cline{1-2}         % horizontal line spanning columns 1-2
row 2, col 1 & row 2, col 2 & row 2, col 3 & row 2, col 4 \\
\cline{1-2}
\multicolumn{2}{|c|}{spanning two columns} & row 3, col 3 &
row 3, col 4 \\
\cline{1-3}
row 4, col 1 & row 4, col 2 & row 4, col 3 & ~ \hfill right \\
% force a space with "\ "
row 5, col 1 & row 5, col 2 & row 5, col 3 & left \hfill ~ \\
row 5, col 1 & row 5, col 2 & row 5, col 3 &
~ \hfill centre \hfill ~ \\
\hline
\end{tabular}
\caption{A sample table.}
% labels are used for cross references;
% for example, "see Table~\ref{tab:sampletab}"
\label{tab:sampletab}
\end{center}
% \end{small}
\end{table}
```

Table 8 shows the results. Let's remind that tables are floats, as explained in Section 4.12.

Left	**Centre**	**Right**	**4 cm**
row 1, col 1	row 1, col 2	row 1, col 3	row 1, col 4
row 2, col 1	row 2, col 2	row 2, col 3	row 2, col 4
spanning two columns		row 3, col 3	row 3, col 4
row 4, col 1	row 4, col 2	row 4, col 3	right
row 5, col 1	row 5, col 2	row 5, col 3	left
row 5, col 1	row 5, col 2	row 5, col 3	centre

Table 8: A sample table.

Sometimes, a table is too wide and won't fit on the page. In that case, the rotating package provides the new environment `sidewaystable`. Moreover, the longtable package provides support for tables that span more than one page.

[TODO: provide a few examples.]

To enable colours in tables, you use the colortbl package:

```
Colour by row:\\\vskip 2mm
\begin{tabular}{|l|c|r|}
\hline
\rowcolor{cyan}
one & two & three\\
\rowcolor{green}
one & two & three\\
\rowcolor{yellow}
one & two & three\\
\hline
\end{tabular}
```

Colour by row:

one	two	three
one	two	three
one	two	three

```
Colour by column:\\\vskip 2mm
\begin{tabular}
{|>{\columncolor{cyan}}l|
>{\color{red}
\columncolor{green}}c|
>{\columncolor{yellow}}r|}
\hline
one & two & three\\
one & two & three\\
one & two & three\\
\hline
\end{tabular}
```

Colour by column:

one	two	three
one	two	three
one	two	three

6.0.1 Aligning Numbers

A special case of a tabular environment is when we want to align numbers with respect to the decimal positions. You use the dcolumn package, which adds the D column specifier. D has three arguments: the separator to use in the LaTeX source and in output (usually the same, '.'), and the number of digits to the right of the decimal place indicator. Optionally, the third argument can specify the number of digits to the left and to the right of the decimal place indicator, separated by a dot. Lastly, if the third argument is -1, the material of the column is centered around the separator.

All material in the table is typeset in math mode. To insert headings, you'll have to put the text in an \mbox.

```
\begin{tabular}{|D{.}{,}{4.2}|%
D{.}{.}{5}|D{.}{.}{-1}|}
\hline
\mbox{One} & \mbox{Two} &
\mbox{Three} \\
10.33 & 10.33 & 10.33 \\
1000 & 1000 & 1000 \\
5.1 & 5.1 & 5.1 \\
3.14 & 3.14159 & 3.14159 \\
\hline
\end{tabular}
```

One	Two	Three
10,33	10.33	10.33
1000	1000	1000
5,1	5.1	5.1
3,14	3.14159	3.14159

[TODO: mention \newcolumntype, \multicolumn{1}{c}{stuff}, the floatflt package.]

7 The Tools Menu

7.1 Tools/Mail Merges

This useful and time-saving tool is implemented in LaTeX by the textmerg package. Let's consider a simple document, in which the name, surname, and title of people we're writing to may vary. The remaining text does not change.

We'll define three *fields*, which are the variable part of the text: `\Name`, `\Surname`, and `\Title`. Their values will be gathered from an external file, `data.dat`.

```
\documentclass{article}
\usepackage{textmerg}
\begin{document}
% let's declare the variable fields:
% \Void is for empty lines
\Fields{\Name\Surname\Title-\Void}
\Merge{data.dat}{%
Dear \Title{} \Surname,\\
may I call you \Name?\\
Yours,\\
\hspace{3cm}Guido\clearpage}
\end{document}
```

The fourth field, `\Void`, isn't really necessary and it's there for purpose of argument. It's preceded by a minus sign, which indicates that it can be empty in the data file. Simply put, we want to separate the records using empty lines.

The file `data.dat` reads:

```
Guido
Gonzato
Dr.

Francesco
Mulargia
Prof.

Carol
John-Paul II
Pope
```

That's it: the resulting output will contain the merged text, one page for each recipient.

7.2 Tools/Labels

If making mail merges was easy, making labels is even trivial. Let's suppose you want to make 20 equal labels on a 3×8 peel–off label sheet. The package to use, predictably, is called labels. In this example, we'll make 10 plain labels and 10 boxed labels:

```
\documentclass[a4paper,12pt]{article}
\usepackage{labels}
\LabelCols=3      % n. of columns of labels
\LabelRows=8      % n. of rows of labels
\LeftBorder=8mm   % borders of each label
\RightBorder=8mm
\TopBorder=5mm
\BottomBorder=5mm
\LabelGridtrue      % show the grid
\numberoflabels=10  % number of labels of each type to print
% the text of the label is specified by
% the \addresslabel[]{} macro:
\begin{document}
  \addresslabel[\large] % optional arguments
  {\textbf{Guido Gonzato}, Ph.D.\\
  \textsl{Linux system manager}}
  % now on to the boxed labels
  \boxedaddresslabel[\fboxsep=4mm\fboxrule=1mm]
  {\textbf{Guido Gonzato}, Ph.D.\\
  \textsl{Linux system manager}}
\end{document}
```

To make labels containing different addresses, you may use either an external file or insert the addresses in the main file:

```
\documentclass[a4paper,12pt]{article}
\usepackage{labels}
\LabelCols=3
\LabelRows=8
\LeftBorder=3mm
\RightBorder=3mm
\TopBorder=8mm
\BottomBorder=8mm
\LabelGridtrue
\begin{document}
% use either this environment:
\begin{labels}
  1$^{st}$ name
  1$^{st}$ address
  1$^{st}$ city, state, zipcode

  2$^{nd}$ name
  2$^{nd}$ address
  2$^{nd}$ city, state, zipcode

  3$^{rd}$ name
  3$^{rd}$ address
  3$^{rd}$ city, state, zipcode
\end{labels}
% or an external file containing exactly the same text:
% \labelfile{addresses.dat}
\end{document}
```

It is left to you to combine textmerg and labels!

7.3 Tools/Default Language

LaTeX default language is English, but other languages are supported. By language support I mean the translation of terms like “Chapter” or “Index”, correct hyphenation, and the possibility of inserting characters like ‘ç’ or ‘é’ directly via your keyboard. (The normal way being typing `\c c` and `\'e`.)
If you are not a native English speaker, you'll want to use the package babel like in the following example:

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

In addition, to type accented letters and in general non-standard ASCII characters[4] you may want to use the package isolatin1. However, this is not a recommended practice, because it reduces the legibility and portability of your file: you had better stick to the TeX way.
If you just can't stand typying all those three- or four-letter clusters, then you could configure your editor to type those for you. For example, I set up my editor of choice (`jed`) to have it type `\'e` whenever I type ‘é’. This is what I included in my `.jedrc`:

```
define latex_mode_hook ()
{
  set_abbrev_mode (1);
  if ( () = abbrev_table_p ("LaTeX") )
    use_abbrev_table ("LaTeX");
  local_setkey (" \\`a", "à");
  local_setkey (" \\`e", "é");
  local_setkey (" \\'e", "è");
  local_setkey (" \\`\\i{}", "ì");
  local_setkey (" \\`o", "ò");
  local_setkey (" \\`u", "ù");
}
```

Please consult your editor's documentation.

7.4 Tools/Hyphenation

Although LaTeX does a good job at hyphenating words, sometimes manual intervention may yield better results. Manual hyphenations are specified inserting `\-` where we want the word to be broken. A better way is to declare hyphenation rules:

```
\hyphenation{ge-o-phy-sics, ge-o-lo-gy, earth}
```

The above declaration instructs LaTeX not to hyphen the word “earth”. Another way to prevent a word to be hyphenated is to enclose it in `\mbox`:

```
Do not hyphen \mbox{internationalisation}.
```

[4] in computer jargon, “standard ASCII characters” are those characters whose code is included between 32 (space) and 126 (tilde).

7.5 Tools/Spell Check

Spell checking the text is not something LaTeX is aware of; it is another tool's work. Under UNIX, you can use `ispell` this way:

```
shell> ispell -t mydocument.tex
```

The `-t` switch instructs `ispell` to ignore TeX and LaTeX commands. If your language is different than English, specify the appropriate dictionary with the `-d` switch:

```
shell> ispell -d italiano -t mydocument.tex
```

8 The Help Menu

There are many ways of getting help with LaTeX, both online and offline. The best place to start is the CTAN site, `http://www.ctan.org/tex-archive/info/`.

- `info latex` (UNIX systems) gives a concise but very complete on-line summary of commands and concepts;
- `http://www.giss.nasa.gov/latex/` is a comprehensive on-line reference. Lots of useful links!
- `http://www.ctan.org/tex-archive/info/LatexHelpBook/` is a very fine help system for LaTeX, fully integrated with Windows.
- don't forget the `news:comp.text.latex` newsgroup.

Many GNU/Linux distributions ship with `teTeX`, one of the most complete TeX/LaTeX systems. A lot of documentation is provided, and is usually found in `TEXMF/doc`. Point your browser there, and be happy.

9 The End

 I really hope you'll find this guide useful. For any suggestions or comments, please feel free to contact me.

A Document Templates

A template for the class `article` was presented in Section 2.1. More examples are shown in the following figures.

```
\documentclass[twoside,11pt]{book}
\begin{document}
\frontmatter
\begin{titlepage}
\title{The Book of Mine}
\end{titlepage}
\author{John B. Smith}
\maketitle
\tableofcontents
\mainmatter
\part{The Beginning}
\chapter{Introduction}
\section{Let's Start}
The book starts here.
\part{The End}
\backmatter
Thank you for reading this book.
\end{document}
```

Figure 3: Book template.

```
\documentclass[twoside,12pt]{report}
% tables and figures at the end:
\usepackage{endfloat}
\begin{document}
\frontmatter
\title{Final Report}
\author{John B. Smith}
\date{London, \today}
\maketitle
\begin{abstract}
This is the final report.
\end{abstract}
\tableofcontents
\listoftables
\listoffigures
\mainmatter
\part{Start}
\section{Introduction}
The report starts here.
\end{document}
```

Figure 4: Report template.

```
\documentclass[12pt]{letter}
\begin{document}
\address{My address}
\signature{Guido}
\begin{letter}{John's address}
\opening{Dear John,}
Thank you for being my friend.
\closing{Hope to see you soon,}
\ps{P.S. Say hello to granny!}
\encl{My son's photographs!}
\end{letter}
\end{document}
```

Figure 5: Letter template.

```
\documentclass[a4paper]{article}
\usepackage{type1cm}
\usepackage{times}
\usepackage{color}
\usepackage{rotating}
\pagestyle{empty}
\begin{document}
\begin{sidewaysfigure}
  \fontsize{2.5cm}{2.5cm}\selectfont
  \centerline{\textcolor{blue}{\textbf{Please:}}}
  \vskip 1cm
  \fontsize{4cm}{3cm}\selectfont
  \centerline{\textcolor{red}{DO NOT}}
  \centerline{\textcolor{red}{SMOKE}}
  \centerline{\textcolor{red}{HERE!}}
  \vskip 1cm
  \fontsize{2cm}{2cm}\selectfont
  \centerline{\textcolor{magenta}{If you do,}}
  \centerline{\textcolor{magenta}{you'll be \emph{deboned!}}}
\end{sidewaysfigure}
\end{document}
```

Figure 6: How to write a notice.

```
\documentclass{article}
\usepackage[absolute,showboxes]{textpos}
\usepackage{color}
\usepackage{framed}
\usepackage{graphicx}
\setlength{\TPHorizModule}{10mm} % standard unit of length
\setlength{\TPVertModule}{\TPHorizModule}
\setlength{\TPboxrulesize}{1pt}  % box line width
% start everything near the top-left corner
\textblockorigin{0mm}{0mm}

\begin{document}
\setlength{\parindent}{0pt}
\definecolor{shadecolor}{rgb}{0.9,1,1}
\begin{textblock}{5}(0,0)
% this block is 5 modules wide; heigth is
% automatically determined
\begin{center}
  \begin{minipage}[c]{0.8 \linewidth}
  \begin{shaded}
  This block is placed with its top left corner at the ‘origin’
  on the page, which has been set to (0mm,0mm). The internal
  margin and she shading are provided by the \texttt{minipage}
  and \texttt{shaded} environments.
  \end{shaded}
  \end{minipage}
\end{center}
\end{textblock}
\begin{textblock}{6}(10,1)
  \includegraphics[width=6cm,angle=-90]{gnuplot.ps}
  This picture is at (10,1). Note that rotating it
  by -90 makes it overflow the margin.
\end{textblock}
\begin{textblock}{5}[0.5,0.5](2.5,8)
This block is at position (2.5,8), but because the optional
argument [0.5,0.5] has been given, it is the centre of the block
which is located at that point, rather than the top-left corner.
\end{textblock}
\begin{textblock}{3,4}(6,4)
The dimensions of this block are 3$\times$4 cm.
Its origin is position (6,7) on the page. Note that the text
overflows the margin in some cases; you’ll want to
use the \texttt{minipage} environment to prevent that.
\end{textblock}
\end{document}
```

Figure 7: How to write a poster.