Abstract

Text processing with \LaTeX offers several advantages over word processing. However, beginners may find it hard 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 capabilities are listed, along with their equivalent \LaTeX commands. Many examples are provided.
List of Tables

1 Useful key bindings for Emacs, Vim, and Jed in IDE mode. ..................................... 3
2 How to obtain some special characters. ................................................................. 12
3 Font attributes. ........................................................................................................ 23
4 Font sizes ................................................................................................................ 24
5 Actual font size in pt ............................................................................................... 24
6 Common font families. ............................................................................................. 26
7 Standard \LaTeX\ environments. ............................................................................... 28
8 A sample table. ........................................................................................................ 32

List of Figures

1 A smiley representing the author of this guide. ....................................................... 17
2 A Gnuplot graph. .................................................................................................... 18
3 Inserting a \LaTeX\ formula. .................................................................................... 19
4 A \LaTeX\ object can be edited as desired. ............................................................... 20
5 Book template. ........................................................................................................ 40
6 Report template. ..................................................................................................... 41
7 Letter template. ...................................................................................................... 41
8 How to write a notice. ............................................................................................ 42
9 How to write a poster. ............................................................................................ 43

1 Introduction

First of all, let me state that this is not a \LaTeX\ primer! If you’re reading this document, I
assume that you have at least a basic understanding of \LaTeX\ and of its basic commands.
In this guide, I’ll explain how to replace a word processor effectively using \LaTeX. 

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, buggy, crash-prone, expensive, virus ridden, and incompatible with each
other. Not to talk about their default output quality.

\LaTeX\ is an excellent alternative (in some cases, it is the only viable alternative); but it’s
not much intuitive for those accustomed to WYSIWYG.

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 some basic \LaTeX\ knowledge;
if it’s not the case, I suggest that you go to http://www.ctan.org/starter.html and download
The (Not So) Short Introduction to \LaTeX\2e. Another good primer is
http://en.wikibooks.org/wiki/LaTeX/.

\footnote{\textsuperscript{1}once upon a time, I wrote my thesis on a 128k RAM, 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!}
In the following sections, we shall navigate through the menus and menu items of an imaginary word processor, finding out the corresponding \LaTeX{} way of doing the same work.

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 \url{http://www.ctan.org}; this site has a wide list of mirrors. From now on, CTAN means ‘your favourite CTAN mirror here, starting from the \TeX{} directory’. For instance, you can get \LaTeX{} for your platform from CTAN://systems (e.g., \url{http://www.tex.ac.uk/tex-archive/systems/}).

To write your documents, you will need a good text editor. A better choice for beginners is a \TeX{} \textit{IDE}: an editor dedicated to writing \LaTeX{} source, with preview and many facilities.

I suggest that you install one of the programs listed below; all of them are Free/Open Source software.

- Texmaker (multiplatform): \url{http://www.xm1math.net/texmaker/index.html}
- TeXstudio (multiplatform): \url{http://texstudio.sourceforge.net/}
- TeXworks (multiplatform): \url{http://tug.org/texworks/}
- TeXShop (Mac OS X): \url{http://www.uoregon.edu/~koch/texshop/}
- TeXnicCenter (Windows): \url{http://www.texniccenter.org/}

Information about \LaTeX{} on the Mac can be found at \url{http://www.esm.psu.edu/mac-tex/}.

1.1.1 Editor-Supported Features

\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 \texttt{emacs} and \texttt{vim} with their native key bindings, and \texttt{jed} configured for Borland IDE key bindings.

1.1.2 Adding Packages

The following information applies to \TeX{} Live, which ships with most GNU/Linux distributions. It may also work on Mac\TeX{}, but I have no first-hand experience. Instructions for MiK\TeX{} (probably the most popular implementation for Windows) will follow below.
<table>
<thead>
<tr>
<th><strong>Action</strong></th>
<th><strong>Emacs</strong></th>
<th><strong>Vim</strong></th>
<th><strong>Jed</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>command mode</td>
<td>Alt-X</td>
<td>ESC</td>
<td>Alt-X</td>
</tr>
<tr>
<td>insert mode</td>
<td>n/a</td>
<td>i a o O</td>
<td>n/a</td>
</tr>
<tr>
<td>line editor mode</td>
<td>n/a</td>
<td>:</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### File Operations

<table>
<thead>
<tr>
<th></th>
<th><strong>open file</strong></th>
<th><strong>insert file</strong></th>
<th><strong>save file</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ctrl-X Ctrl-F</td>
<td>Ctrl-Xi</td>
<td>Ctrl-X Ctrl-S</td>
</tr>
<tr>
<td></td>
<td>:e</td>
<td>:r</td>
<td>:w</td>
</tr>
<tr>
<td></td>
<td>Ctrl-KE</td>
<td>Ctrl-KR</td>
<td>Ctrl-KD</td>
</tr>
<tr>
<td>save as</td>
<td>Ctrl-X Ctrl-W name</td>
<td>:w name</td>
<td>Ctrl-KS</td>
</tr>
<tr>
<td>close file</td>
<td>Ctrl-XK</td>
<td>:q</td>
<td>Ctrl-KQ</td>
</tr>
<tr>
<td>change buffer</td>
<td>Ctrl-XB</td>
<td>bN</td>
<td>Ctrl-KN</td>
</tr>
<tr>
<td>undo</td>
<td>Ctrl-XU</td>
<td>u</td>
<td>Ctrl-U</td>
</tr>
<tr>
<td>redo</td>
<td>Ctrl-&lt;</td>
<td>Ctrl-R</td>
<td>Ctrl-G Ctrl-U</td>
</tr>
<tr>
<td>exit</td>
<td>Ctrl-X Ctrl-C</td>
<td>:qa!</td>
<td>Ctrl-KX</td>
</tr>
</tbody>
</table>

### Moving Around

<table>
<thead>
<tr>
<th></th>
<th><strong>word left</strong></th>
<th><strong>word right</strong></th>
<th><strong>start of line</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt-B b</td>
<td>Alt-F w</td>
<td>Ctrl-A</td>
</tr>
<tr>
<td></td>
<td>Ctrl-A 0</td>
<td>Ctrl-QS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ctrl-E $</td>
<td>Ctrl-QD</td>
<td></td>
</tr>
<tr>
<td>page up</td>
<td>Alt-V Ctrl-U</td>
<td>Ctrl-R</td>
<td></td>
</tr>
<tr>
<td>page down</td>
<td>Ctrl-V Ctrl-D</td>
<td>Ctrl-C</td>
<td></td>
</tr>
<tr>
<td>start of buffer</td>
<td>Alt-&lt; 1G</td>
<td>Ctrl-QR</td>
<td></td>
</tr>
<tr>
<td>end of buffer</td>
<td>Alt-&gt; G</td>
<td>Ctrl-QC</td>
<td></td>
</tr>
<tr>
<td>line n.</td>
<td>Alt-G n. n.G</td>
<td>Ctrl-QI</td>
<td></td>
</tr>
</tbody>
</table>

### Deleting

<table>
<thead>
<tr>
<th></th>
<th><strong>character left</strong></th>
<th><strong>character right</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ctrl-H X</td>
<td>Ctrl-D x</td>
</tr>
<tr>
<td></td>
<td>Ctrl-G Alt-G</td>
<td></td>
</tr>
<tr>
<td>word left</td>
<td>Alt-DEL db</td>
<td>Alt-BS</td>
</tr>
<tr>
<td>word right</td>
<td>Alt-D dw</td>
<td>Ctrl-T</td>
</tr>
<tr>
<td>end of line</td>
<td>Ctrl-K d$</td>
<td>Ctrl-QY</td>
</tr>
<tr>
<td>line</td>
<td>Ctrl-A Ctrl-K dd</td>
<td>Ctrl-Y</td>
</tr>
</tbody>
</table>

### Search & Replace

<table>
<thead>
<tr>
<th></th>
<th><strong>search</strong></th>
<th><strong>replace</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ctrl-S text</td>
<td>Alt-% :s/old/new/g</td>
</tr>
<tr>
<td></td>
<td>/text</td>
<td>Ctrl-QS</td>
</tr>
<tr>
<td></td>
<td>Ctrl-QS</td>
<td>Ctrl-QA</td>
</tr>
</tbody>
</table>

### Blocks

<table>
<thead>
<tr>
<th></th>
<th><strong>start selection</strong></th>
<th><strong>cut</strong></th>
<th><strong>copy</strong></th>
<th><strong>paste</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ctrl-SPACE v</td>
<td>Ctrl-W D</td>
<td>Alt-W Y</td>
<td>Ctrl-Y P</td>
</tr>
<tr>
<td></td>
<td>Ctrl-KB</td>
<td>Ctrl-KY</td>
<td>Ctrl-KH</td>
<td>Ctrl-KC</td>
</tr>
</tbody>
</table>

Table 1: Useful key bindings for Emacs, Vim, and Jed in IDE mode.
A large collection of \LaTeX\ packages is natively supported; for instance, Ubuntu provides lots of `texlive-*` packages. Please note the ambiguity: we’re talking of `.deb packages` containing \LaTeX\ packages.

If you need to install an unsupported package, do the following.

1. create this directory structure:

   ```
   $ mkdir -p ~/texmf/tex/latex
   ```

   Under this directory, new packages will be installed.

2. get the package (typically as a zip-compressed directory) from your favourite CTAN mirror; let’s call it `foo.zip`

3. unpack it in the right place:

   ```
   $ mkdir ~/texmf/tex/latex/foo
   $ mv foo.zip ~/texmf/tex/latex/foo
   $ cd ~/texmf/tex/latex/foo ; unzip foo.zip
   ```

4. If no `.sty` file exists, run the command `latex foo.ins` or `latex foo.dtx` to create it;

5. run the command `texhash ~/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!

Once a package is installed, you make it available in your documents adding a line under the `documentclass` declaration:

```latex
\usepackage{foo}
```

### 1.1.3 Adding the Info Page

‘Man’ and ‘Info’ pages are command-line sources of documentation for software, widely employed in Unix and Linux systems. If your \LaTeX\ distributions lacks the `latex2e.info` info page, do the following:

1. download it from `http://tug.ctan.org/info/latex2e-help-texinfo/latex2e.info;`

2. run these commands:

   ```
   $ sudo cp latex2e.info /usr/share/info/
   $ sudo ginstall-info latex2e.info dir
   ```

Now, `info latex` or `info latex2e` is available.
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 true even if you’re not writing a scientific paper.

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*.

Applying these simple rules, your printed material will magically look professional.

That said, this guide (which breaks the second rule somehow!) will help you overcome the second rule. So, you will also be able to compose unstructured material such as circulars, notices, poster, etc.

2 The File Menu

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

2.1 File/New

This is the \LaTeX{} equivalent of a blank page:

\begin{verbatim}
\documentclass{article}
\thispagestyle{empty} % no page number
\begin{document}
% This is a comment. Write your stuff here.
\end{document}
\end{verbatim}

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

\begin{verbatim}
\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}
\end{document}
\end{verbatim}
This is the end of the document. Please go to Section \ref{sec:start} to read it again.

More document templates are listed in Appendix A.

2.2 File/Save As...

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

- \TeX{}4ht is probably the best \LaTeX{} to HTML converter:
  http://tug.org/tex4ht

- latex2html, another converter to HTML:
  http://saftsack.fs.uni-bayreuth.de/~latex2html/, 
  CTAN://support/latex2html

- latex2rtf, a converter to Rich Text Format:
  CTAN://support/latex2rtf

- detex (a command line converter) removes all \LaTeX{} tags and outputs plain text:
  http://www.cs.purdue.edu/homes/trinkle/detex/, 
  CTAN://support/detex/

See also Section 2.7 for details on PDF creation.

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://support/rtf2latex

- html2latex: CTAN://support/html2latex

- wvware is a set of tools that convert from MS Word to several formats including \LaTeX{}: http://wvware.sourceforge.net

- the free word processor Abiword, http://www.abiword.org, imports MS Word and can export \LaTeX{}.

- txt2tex: CTAN://support/txt2tex does a fairly good job at converting plain text files into \LaTeX{.}
Other LaTeX 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 \texttt{a4paper}, \texttt{a5paper}, \texttt{b5paper}, \texttt{letterpaper}, \texttt{legalpaper}, \texttt{executivepaper}; orientation is \texttt{portrait} by default, or optionally \texttt{landscape}. For example,

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

Document-wide margins are set with the \texttt{\setlength} command, which is used to change the value of variables and counters:

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

The \texttt{geometry} package allows for complete control of parameters such as paper size, margins width, and so on. \texttt{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.

\usepackage{geometry} % top of document
...
\geometry{paperwidth=25cm}
\geometry{paperheight=35cm}
% or: \geometry{papersize={25cm,35cm}}
\geometry{width=20cm} % total width
\geometry{height=30cm} % total heigth
% or: \geometry{total={20cm,30cm}}
\geometry{textwidth=18cm} % width - marginpar
\geometry{textheight=25cm} % heigth - header - footer
% or: \geometry{body={18cm,25cm}}
\geometry{left=3cm} % left margin
\geometry{right=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 this way:

\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:

\begin{verbatim}
\usepackage{fancyhdr}
...
\lhead{} % empty
\chead{Hello, world!}
\rhead{Page \thepage} % page number
\lfoot{}
\cfoot{\textbf{Hello!}}
\rfoot{}
\end{verbatim}

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

See below for more tips.

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 DVI previewer;
- convert the `.dvi` to PostScript with `dvips`, then use some variant of `Ghostview`;
- producing a `.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. While `dvipdf` is just a script that converts the `.dvi` to `.ps` then to `.pdf`, using `pdflatex` is more interesting. In fact, some packages like `hyperref` and `url` make the resulting `.pdf` file browseable; see Section 4.15. However, using `pdflatex` you might experience problems with other packages. More details in Section 4.12.

2.8 File/Print

Simply use `lpr file.ps` from the command line, or via the File/Print menu entry of your previewer.
2.9 File/Versions

Although the version package provides basic facilities for versioning \LaTeX sources, using external tools is usually a better choice.

There are lots of programs for revision control and collaborative writing. While old-style geeks may want to use a single-user tool like RCS (I do), more powerful and multi-user tools like Subversion, Git, Mercurial etc. may be preferable. Some of them are integrated with the editor.

An introduction to \LaTeX and Subversion is here:

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.

Selecting 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 between braces, or in an environment. For example, to apply the bold attribute to a portion of text you will use one of the following:

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

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:

\begin{verbatim}
\def \PS {\textsc{PostScript}}
\end{verbatim}

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 ` (tilde)
- to force a line break, use \linebreak or \newline; see below for details
- to start a new paragraph use a blank line, or ` ` (possibly followed by an optional space, as in ` `\[1cm\]), or \par
- finally, for a page break use \newpage or \clearpage
Unlike \newline, \linebreak stretches the line to the margins:

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

Besides, \clearpage differs from \newpage in that it prints all pending floats, that is 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 customise 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;
  \item[.] with a dot.
\end{itemize}

Another way is to redefine the style of the counters\footnote{each text element numbered by \LaTeX (sections, lists, figures, etc.) has a counter associated with it.} hat refer to the digits in the first-to fourth-level lists. These counters are \labelitemi, \labelitemii, \labelitemiii and \labelitemiv for itemised lists; \labelenumi, \labelenumii, \labelenumiii and \labelenumiv for enumerated lists.

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 will not explain for now:

\begin{itemize}
  \item first level, item 1
  \item first level, item 2
  \item second level, item 1
  \item second level, item 2
  \item first level, item 3
\end{itemize}
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}

Alternatively, use the \texttt{enumerate} package. It redefines the \texttt{enumerate} environment with the ability to specify an optional argument. Any occurrence of one of the characters A a I i 1 will produce the value of the counter, using (respectively) \texttt{Alph}, \texttt{alph}, \texttt{Roman}, \texttt{roman}, or \texttt{arabic}. To include some text, enclose it in braces:

\begin{enumerate}[]
\item [{Example I.}] First example.
\item Second example.
\item Last example. Go to Item ~\ref{item:first}.
\end{enumerate}

To change the number of an enumerated item, just redefine its counter:

\begin{enumerate}
\setcounter{enumi}{2}
\item Example 3.
\item Example 4.
\setcounter{enumi}{5}
\item Example 6.
\end{enumerate}

Lists within paragraphs are implemented by the \texttt{paralist} package, which provides the \texttt{inparaenum} environment:

I’ll throw in a list of items:
\begin{inparaenum}
\item apples,
\item pears, and
\item oranges.
\end{inparaenum}
The same list can be labelled with letters:
\begin{inparaenum}[a)]
\item apples, \label{first}
\item pears, and
\item oranges. The first item is \ref{first}.
\end{inparaenum}

I’ll throw in a list of items: 1. apples, 2. pears, and 3. oranges. The same list can be labelled with letters: a) apples, b) pears, and c) oranges. The first item is a.

As seen above, the characters A a I i 1 can be used to modify the counter. \texttt{paralist} can do much more, and I suggest that you read its documentation.
### Table 2: How to obtain some special characters.

<table>
<thead>
<tr>
<th>Character</th>
<th>\LaTeX{} Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>$ or \textdollar</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>_</td>
<td>_ or \textunderscore</td>
</tr>
<tr>
<td>{</td>
<td>{ or \textbraceleft</td>
</tr>
<tr>
<td>}</td>
<td>} or \textbraceright</td>
</tr>
<tr>
<td>&lt;</td>
<td>$&lt;$ or \textless</td>
</tr>
<tr>
<td>&gt;</td>
<td>$&gt;$ or \textgreater</td>
</tr>
<tr>
<td>\</td>
<td>\textbackslash</td>
</tr>
<tr>
<td>•</td>
<td>\textbullet</td>
</tr>
<tr>
<td>‡</td>
<td>\textdaggerdbl</td>
</tr>
<tr>
<td>†</td>
<td>\textdagger</td>
</tr>
<tr>
<td>¶</td>
<td>\textparagraph</td>
</tr>
<tr>
<td>§</td>
<td>\textsection</td>
</tr>
<tr>
<td>®</td>
<td>\textcopyright</td>
</tr>
<tr>
<td>^</td>
<td>\textasciicircum</td>
</tr>
<tr>
<td>~</td>
<td>\textasciitilde or ~{}</td>
</tr>
<tr>
<td>≈</td>
<td>$\sim$</td>
</tr>
<tr>
<td>™</td>
<td>\textregistered</td>
</tr>
<tr>
<td>a</td>
<td>\textordfeminine</td>
</tr>
<tr>
<td>o</td>
<td>\textordmasculine</td>
</tr>
</tbody>
</table>

#### 4.3 Insert/Special Character

First of all, let’s remind that some characters have a special meaning in \LaTeX{}. They must be entered either with a leading \, or using them in math mode, or even writing special commands: see Table 2.

Another way to enter special characters is using their ASCII code and the \texttt{\char} command. For example, you could insert the characters $\&^\sim$ entering \texttt{\char36 \char38 \char94 \char126}.

There are packages that provide thousands of unusual characters and symbols. For instance, \texttt{pifont} provides the commands \texttt{\ding}, \texttt{\dingfill}, \texttt{\dingline}, and \texttt{\dinglist}. The first command produces the Dingbat character of the specified code. The other commands are equivalent to the \texttt{\fill}, \texttt{\line}, and \texttt{\list} commands and environment, but use the Dingbat code given as parameter:

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

12
Another nice variant is this:

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

There are far too many symbols to mention them all in this guide. Rather, I’ll point you to ‘The Comprehensive \LaTeX Symbol List’ at CTAN://info/symbols/comprehensive.

4.3.1 The € Sign

The official Euro sign is provided by the \texttt{eurosym} package, which can be used in two ways:

\begin{verbatim}
\usepackage[gen]{eurosym}
\usepackage[official]{eurosym}
\end{verbatim}

Both provide the \texttt{\euro} command, which produces €. The actual shape of the symbol depends on the declaration: the \texttt{[gen]} option provides a symbol that works with all font styles; the second declaration produces €. Please note the difference. The latter shape can always be obtained with \texttt{\officialeuro}.

Another package that provides the Euro sign is \texttt{marvosym}, which also provides many more fine characters. You get € with \texttt{\EUR}.

4.4 Insert/Formula

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

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

The environments \texttt{displaymath} and \texttt{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}
\forall n \neq 2 
\begin{align*}
x^n + y^n &\neq z^n \\
\label{eq:fermat}
\end{align*}
\end{equation}

Can you prove Eq. \ref{eq:fermat}?

\begin{equation}
\forall n \neq 2 
\begin{align*}
x^n + y^n &\neq z^n \\
\end{align*}
\end{equation}

4.5 Insert/Footnote

The command \texttt{\footnote[n]{footnote text}} is all you need; the optional parameter \texttt{[n]} modifies the footnote number. The \texttt{\footnote} command should be placed after commas, full stops, and other punctuation signs that follow the word.
To use a symbol or arbitrary text instead of a number, redefine the counter associated with \footnote:

\renewcommand\{\thefootnote\}{read me!}
Thisfootnote\footnote{I mean this one.}
says it all.

Using this method, 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!}

Note the \fnsymbol{footnote} thing. It uses 9 symbols associated with the values 1...9 of the footnote counter: * † ‡ § ¶ ∥ ∗∗ †† ‡‡.

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.

Warning: minipages use their own counters, mpfootnote and thempfootnote.

4.5.1 Footnotes at End of Document

The endnotes package lets you move all footnotes at the end of the document. You’ll have to add this line to the preamble:

\let\footnote=\endnote

and these lines as the last thing in your document:

\newpage
\begingroup
\parindent 0pt
\parskip 2ex
\def\enotesize{\normalsize}
\theseendnotes
\endgroup

Other commands are available, please read the endnotes.sty source.
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 \texttt{section} or \texttt{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 am aware of. 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 \texttt{\null} or \texttt{~} to set fixed marks, followed by \texttt{vfill} and \texttt{hfill} like in this example:

\begin{verbatim}
one \hfill two\\
vfill
- \hfill three \hfill ~\\
vfill
four \hfill five
\null
\end{verbatim}

Normally, \LaTeX won’t let you insert blank spaces at your will: two or more are considered a single space. However, if you do want to make your document look messy, use ~ to make a non-breakable space.

Also, use the command \texttt{hspace} like in this example:

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

4.8 Insert/Tabs

The \texttt{tabbing} environment provides a rough equivalent to the action of the TAB key, and it’s used to align text in columns. These are the most commonly used commands:

\begin{tabular}{|l|l|}
\hline
Command & Action \\
\hline
\= & Sets a tab stop \\
\> & Advances to the next tab stop \\
\+ & Sets the left margin one tab stop to the right \\
\- & Sets the left margin one tab stop to the left \\
\\ & Ends a line \\
\pushtabs & Saves all tab stop positions \\
\poptabs & Restores previously saved tab stop positions \\
\hline
\end{tabular}
This example shows some of the available commands:

\begin{tabbing}
\% let's set the tab positions
- \hspace{1cm} \= \hspace{2cm} \= \hspace{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 \\ 
\poptabs \% restore tab positions
Zero \> One \> Two \> Three \\
\end{tabbing}

See also the \texttt{tabular} and \texttt{table} environments.

\section*{4.9 Insert/Cross Reference}

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

References to a page (section, table, figure, etc.) number can be obtained using \texttt{\label} and \texttt{\ref} as in this example:

\paragraph{Example.} \label{par:example}
\begin{quote}
This paragraph appears in Section\ref{par:example} on page\pageref{par:example}.
\end{quote}

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}

\section*{4.10 Insert/Margin Notes}

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

\section*{4.11 Insert/Text Frame}

A text frame is used to define a part of text that does not break across pages. To place a text frame on a fixed position on the page, use the \texttt{textpos} package, as shown in the example listed in fig. 9 (see Appendix A).
If you don’t need to specify a fixed position, then use the \texttt{minipage} (miniature page) environment. This text is enclosed in a minipage...

...and this is some other text enclosed in a \texttt{boxedminipage} environment, provided by the package of same name.

I remind you what a \texttt{minipage} declaration looks like:

\begin{minipage}[position]{width}...
\end{minipage}

In a \texttt{boxedminipage}, the space between the frame and the text is set with this command:

\setlength{\fboxsep}{5mm}

4.12 Insert/Figure

(The reference guide for graphic inclusion in \LaTeX is ‘Using Imported Graphics in \LaTeX\texttt{2e}, a.k.a. \texttt{epslatex.ps}.)

A ‘figure’ can be not only a picture, but also a portion of text, a table, etc. that you put in a \texttt{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}

\begin{figure}
\begin{center}
\texttt{=8-)}
\end{center}
\caption{A smiley representing the author of this guide.}
\end{figure}

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’; use ‘\texttt{see fig.}\texttt{\ref{fig:label}}’ instead.

Owing to this property, figures and tables are called floats. If you do need to position a float exactly, use the \texttt{here} package that provides an optional placement argument \texttt{H} (meaning, “right HERE!”).

Given a picture in Encapsulated \texttt{POSTSCRIPT} (.\texttt{eps}) format, you insert it in a \LaTeX source file using the \texttt{graphicx} package and commands like those shown in fig. 2.

When you typeset your document with \texttt{\LaTeX} then \texttt{dvips}, graphic inclusion only works with \texttt{EPS} files; \texttt{pdflatex} accepts JPG, PNG, and of course PDF files. The latter choice might be preferable for most users.
There are several packages that convert common graphic formats like `.jpg`, `.gif`, `.png` etc. to `.eps` and/or `.pdf`; for example, ImageMagik (http://www.imagemagik.org) and The GIMP (http://www.gimp.org). However, these applications produce huge PostScript files.

Best results are obtained using applications that wrap the bitmap, turning it into a compact PostScript file. You’ll want to use `jpeg2ps` (http://www.pdflib.com/jpeg2ps/index.html) or `bmeps` (CTAN://support/bmeps). The former is often the best choice for wrapping `.jpg` files, but the latter handles more graphics formats.

If you wish to make both `.pdf` and `.ps` from the same source file, include these commands:

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

Figure 2: A Gnuplot graph.

\begin{verbatim}
% include the right options
\ifpdf
  \usepackage[pdftex]{graphicx}
  \pdfcompresslevel=9
\else
  \usepackage{graphicx}
\fi

% include the right graphic file
\ifpdf
  \includegraphics{file.png}
\else
  \includegraphics{file.eps}
\fi
\end{verbatim}

If you have more than 18 figures without text between them, you’ll get the ‘Too many unprocessed floats’ \LaTeX error. The quickest way to solve this problem is to put `\clearpage` after three or four figures.
4.12.1 Wrapping Floats

For a magazine-like layout, use the `wrapfig` package:

If you meet this guy, give him some money.

\begin{wrapfigure}[4]{l}[5pt]{2cm}
\Huge
\texttt{=8-)}
\end{wrapfigure}

The reason may not be apparent to you, but I can assure that your money will end up in good hands. I say again, if you meet this guy, give him some money: he knows how to use it properly. OK?

The parameters are the number of lines to be narrowed, the figure placement, the overhang, and the figure width.

4.13 Insert/Shapes

\LaTeX{} provides a `picture` environment within which you use commands like \texttt{\textbackslash circle}, \texttt{\textbackslash 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 a couple of great programs, both free and open source: the vector drawing program Inkscape, https://inkscape.org/, along with Pstoedit, http://www.pstoedit.net/.

Start Inkscape and draw any shape you wish using its tools. To insert text rendered by \LaTeX{}, select Extensions/Render/LaTeX formula..., insert your text as in fig. 3, then click on Apply.

Your \LaTeX{}-rendered text will be included as a graphics object, and you’ll be able to edit it as you wish. The resulting picture can be exported to several formats supported by \LaTeX{}, such as PDF, PNG, and many others. More information here: http://www.ctan.org/tex-archive/info/svg-inkscape.

If you wish to do real magic, then check out the following packages/programs:

- `pgf` is a \TeX{} macro package for generating graphics:
  http://sourceforge.net/projects/pgf/
Figure 4: A \LaTeX{} object can be edited as desired.

- GLE (Graphics Layout Engine) is a graphics scripting language designed for creating publication quality graphs, plots, diagrams, figures and slides:
  www.gle-graphics.org

- Asymptote is a powerful descriptive vector graphics language that provides a natural coordinate-based framework for technical drawing:
  http://asymptote.sourceforge.net/

- ePiX, a collection of batch utilities for GNU/Linux and similar platforms, creates mathematically accurate figures, plots, and movies using easy-to-learn syntax:
  http://mathcs.holycross.edu/~ahwang/current/ePiX.html;

- pstricks is a set of macros that allow the inclusion of PostScript drawings in \LaTeX{} documents:
  http://tug.org/PSTricks/main.cgi/

These packages let you make publication-quality PostScript drawings in \LaTeX{}. Many more are available; search the web for “\LaTeX{} vector graphics”.

Many more kinds of ‘shapes’ can also be inserted. To whet your appetite, please visit the \TeX{} Showcase page, http://www.tug.org/texshowcase/.
4.14 Insert/Line

Draw lines of any length and thickness with \texttt{\textbackslash rule}:

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

Another interesting ‘line’ is that made of dots (\texttt{\textbackslash dotfill}), often used to relate things. This is how it’s done:

Total price \texttt{\textbackslash dotfill \text{"euro}{10}}

4.15 Insert/Hyperlink

The \texttt{hyperref} package lets you write URLs and other external references. When used together with \texttt{dvipdf} or \texttt{pdflatex}, \texttt{hyperref} lets you make browseable .pdf documents!

For instance, this document uses this declaration:

\begin{verbatim}
\usepackage[colorlinks, urlcolor=blue, filecolor=magenta, linkcolor=darkred, hyperfootnotes=false]{hyperref}
\end{verbatim}

Let’s see an example:

The \texttt{\textbackslash hypertarget{\texttt{ctan}}{CTAN}} main site is \texttt{\textbackslash url\{\url{http://www.ctan.org}\}, a.k.a \texttt{\textbackslash href\{\url{http://www.ctan.org}\}{CTAN://}}.\texttt{\textbackslash hypertarget{\texttt{ctan}}{CTAN}} main site is \texttt{\textbackslash url\{\url{http://www.ctan.org}\}, a.k.a \texttt{\textbackslash href\{\url{http://www.ctan.org}\}{CTAN://}}.

Listen to \texttt{\textbackslash href\{\texttt{run:midifile.mid}\} \{this MIDI file\}}.

Click \texttt{\textbackslash hyperlink[\texttt{ctan}]\{here\} to go back to the top.}

As you can see, the \texttt{\textbackslash url} command typesets its contents using a monospace font. To use the same font as the remaining text, use the command:

\begin{verbatim}
\urlstyle{same}
\end{verbatim}

after the \texttt{\textbackslash hyperref} declaration.

The \texttt{\textbackslash hypertarget} and \texttt{\textbackslash hyperlink} commands provide internal links, just like HTML; \texttt{\textbackslash href} creates links to URLs or external files. Note the \texttt{run:} parameter: you can run external programs like multimedia players, office applications, whatever. As far as I know, this feature only works in Adobe Reader, Okular, and Evince.

On Linux and possibly other UNIX variants, you’ll have to instruct your favourite PDF reader what to run when an external file is referenced. Insert lines like the following in your .mailcap or /etc/mailcap:

\begin{verbatim}
21
audio/midi;/usr/bin/timidity %s
audio/*; xmms %s
video/*; xine -pfhq %s

Please read hyperref’s documentation for further examples and possibilities.

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 12 pt), 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 5.2.3.

5.1 Format/Line Spacing

The package setspace provide the environments singlespace, onehalfspace, and doublespace. In addition, the environment/command \spacing{amount} will set the spacing to the specified amount:

\begin{spacing}{2.5}
These two lines \ \ are crazily spaced!
\end{spacing}
\begin{spacing}{1}
Much better, these lines \ \ have a pretty space.
\end{spacing}

These two lines are crazily spaced!
Much better, these lines have a pretty space.

To apply line spacing to the whole document, use the \linespread{factor} command in the preamble. Default value of factor is 1; larger values give larger line spacing (1.6 is roughly double line spacing).

5.2 Format/Character

Standard character properties are listed in Table 3, font sizes in Table 4. Please note that actual font size depends on the default size defined in documentclass (10, 11, or 12 pt); see Table 5.

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 rather than typographic command.
Table 3: Font attributes.

<table>
<thead>
<tr>
<th>Text attribute</th>
<th>Environment form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textnormal</td>
<td>textnormal</td>
<td>main document font</td>
</tr>
<tr>
<td>\textit</td>
<td>itshape</td>
<td>italics</td>
</tr>
<tr>
<td>\texttt</td>
<td>sffamily</td>
<td>sans serif</td>
</tr>
<tr>
<td>\underline</td>
<td>underline</td>
<td>underline</td>
</tr>
<tr>
<td>\textsuperscript</td>
<td>n/a</td>
<td>this is superscript</td>
</tr>
<tr>
<td>\mathbf</td>
<td>n/a</td>
<td>x^n + y^n \neq z^n \forall n \neq 2</td>
</tr>
<tr>
<td>\mathcal</td>
<td>n/a</td>
<td>$\text{H}_2\text{O} + \text{CO}_2 \rightarrow \text{H}_2\text{CO}_3$</td>
</tr>
<tr>
<td>\mathit</td>
<td>n/a</td>
<td>$\text{CaCO}_3 \rightarrow \text{Ca}^{2+} + \text{CO}_3^{2-}$</td>
</tr>
</tbody>
</table>

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

\$$_{\text{subscript}}$$

\[ \text{this is superscript} \]

\$_{\text{\footnotesize{subscript}}} \$

5.2.1 Superscript and Subscript in Chemical Formulae

Most chemical formulae could be entered as math formulae, using ^ and _ to obtain superscript and subscript. The \texttt{mhchem} package provides a simpler command, though. Digits are printed as subscripts by default, as superscript when preceded by ^. Formulae must be enclosed in the \texttt{\ce} command:

\begin{align*}
\text{\ce{H2O + CO2 \rightarrow H2CO3}} & \quad \text{H}_2\text{O} + \text{CO}_2 \rightarrow \text{H}_2\text{CO}_3 \\
\text{\ce{CaCO3 \rightarrow Ca^{2+} + CO3^{2-}}} & \quad \text{CaCO}_3 \rightarrow \text{Ca}^{2+} + \text{CO}_3^{2-} \\
\text{\ce{CO3^{2-} + H2CO3 \rightarrow 2 HC03^{-}}} & \quad \text{CO}_3^{2-} + \text{H}_2\text{CO}_3 \rightarrow 2\text{HCO}_3^{-} \\
\text{\ce{CaCO3 + H2CO3 \rightarrow Ca^{2+} + 2 HC03^{-}}}} & \quad \text{CaCO}_3 + \text{H}_2\text{CO}_3 \rightarrow \text{Ca}^{2+} + 2\text{HCO}_3^{-}
\end{align*}

5.2.2 Underline styles

Normally, underline is not used. It’s just a relic of the old teletype era, and it doesn’t look really good. If you still want to use underline, the \texttt{ulem} package provides some fancy
<table>
<thead>
<tr>
<th>Font size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>tiny</td>
<td>sample text</td>
</tr>
<tr>
<td>scriptsize</td>
<td>sample text</td>
</tr>
<tr>
<td>footnotesize</td>
<td>sample text</td>
</tr>
<tr>
<td>small</td>
<td>sample text</td>
</tr>
<tr>
<td>normalsize</td>
<td>sample text</td>
</tr>
<tr>
<td>large</td>
<td>sample text</td>
</tr>
<tr>
<td>Large</td>
<td>sample text</td>
</tr>
<tr>
<td>LARGE</td>
<td>sample text</td>
</tr>
<tr>
<td>huge</td>
<td>sample text</td>
</tr>
<tr>
<td>Huge</td>
<td>sample text</td>
</tr>
</tbody>
</table>

Table 4: Font sizes

<table>
<thead>
<tr>
<th>Default font size</th>
<th>10pt</th>
<th>11pt</th>
<th>12pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>tiny</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>scriptsize</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>footnotesize</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>small</td>
<td>9</td>
<td>10</td>
<td>10.95</td>
</tr>
<tr>
<td>normalsize</td>
<td>10</td>
<td>10.95</td>
<td>12</td>
</tr>
<tr>
<td>large</td>
<td>12</td>
<td>12</td>
<td>14.4</td>
</tr>
<tr>
<td>Large</td>
<td>14.4</td>
<td>14.4</td>
<td>17.28</td>
</tr>
<tr>
<td>LARGE</td>
<td>17.2</td>
<td>17.28</td>
<td>20.74</td>
</tr>
<tr>
<td>huge</td>
<td>20.7</td>
<td>20.74</td>
<td>24.88</td>
</tr>
<tr>
<td>Huge</td>
<td>24.8</td>
<td>24.88</td>
<td>24.88</td>
</tr>
</tbody>
</table>

Table 5: Actual font size in pt

styles:
\uline{important}
\uuline{urgent}
\uwave{boat}
\sout{wrong}
\xout{removed}

Beware: ulem redefines the \emph command, which will be replaced by underline. To avoid this behaviour, use this declaration:

\usepackage[normalem]{ulem}

5.2.3 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:
Another way to get big fonts is to use the package `type1cm`, which provides commands like the following:

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

(The example above is way too large to fit on this page...) Parameters are font size and baseline. Yet another approach is this:

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

\textsc{D}ropped capitals at the start of a paragraph can be obtained using the \texttt{lettrine} package, which provides a fully customisable \texttt{\lettrine} command. This paragraph uses the default behaviour:

\texttt{\lettrine{D}{ropped} capitals at the start...}

5.2.4 Format/Character Font

\LaTeX{} uses its own fonts (Computer Modern), automatically generated when needed by the \texttt{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 \texttt{POSTSCRIPT} fonts. Try using one of the following packages: \texttt{avant}, \texttt{avangard}, \texttt{bookman}, \texttt{chancery}, \texttt{charter}, \texttt{courier}, \texttt{helvet}, \texttt{helvetica}, \texttt{ncntrsbk}, \texttt{newcent}, \texttt{palatcm}, \texttt{palatino}, \texttt{pifont}, \texttt{times}, \texttt{utopia}, \texttt{zapfchan}. Insert \texttt{\usepackage{times}} and enjoy the results. The only caveat is that \LaTeX{} handles maths at its best only with Computer Modern fonts: using \texttt{POSTSCRIPT} fonts might render your formulas slightly less appealing.

The packages above set the font for the whole document. To use a \texttt{POSTSCRIPT} font for a region of text only, specify the font family as in the example below. Common font families are listed in Table 6.

\text{\textbullet} Beware, some font shapes may be unavailable on some systems!

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

Yet another possibility is replacing a standard \LaTeX{} font with a \texttt{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:

- \texttt{\rmdefault} (roman)
- \texttt{\sfdefault} (sans serif)
- \texttt{\ttdefault} (typewriter)
<table>
<thead>
<tr>
<th>Family</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmr</td>
<td>Computer Modern Roman</td>
</tr>
<tr>
<td>cmss</td>
<td>Computer Modern Sans Serif</td>
</tr>
<tr>
<td>cmtt</td>
<td>Computer Modern Typewriter</td>
</tr>
<tr>
<td>pag</td>
<td>Avantgarde</td>
</tr>
<tr>
<td>pbk</td>
<td>Bookman</td>
</tr>
<tr>
<td>phv</td>
<td>Helvetica</td>
</tr>
<tr>
<td>pnc</td>
<td>New Century Schoolbook</td>
</tr>
<tr>
<td>ppl</td>
<td>Palatino</td>
</tr>
<tr>
<td>ptm</td>
<td>Times</td>
</tr>
<tr>
<td>pcr</td>
<td>Courier</td>
</tr>
</tbody>
</table>

Table 6: Common font families.

- \bfdefault (boldface)
- \mddefault (medium)
- \itdefault (italics)
- \sldefault (slanted)
- \scdefault (small caps)
- \updefault (upright)

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

5.2.5 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}

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 between 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}

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

5.3.1 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 \raggedright, \raggedleft, and \centering are equivalent to their correspondent environments, but they do not start a new paragraph.

5.3.2 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. This means that you can’t get more space between paragraphs inserting more empty lines. The commands \smallskip, \medskip, and \bigskip provide some space between paragraphs.

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

These paragraphs will be separated by 1.3 cm:\
\vskip 1.3cm
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 use \null, which sets a ‘mark’ in the text:

\null
\vskip 1.3 cm
This text comes after 1.3 cm...
<table>
<thead>
<tr>
<th>Environment</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract</td>
<td>abstract</td>
</tr>
<tr>
<td>array</td>
<td>Math arrays</td>
</tr>
<tr>
<td>center</td>
<td>Centered lines</td>
</tr>
<tr>
<td>description</td>
<td>Labelled lists</td>
</tr>
<tr>
<td>displaymath</td>
<td>Formulas on their own line</td>
</tr>
<tr>
<td>document</td>
<td>Encloses the whole document</td>
</tr>
<tr>
<td>enumerate</td>
<td>Numbered lists</td>
</tr>
<tr>
<td>eqnarray</td>
<td>Sequence of aligned equations</td>
</tr>
<tr>
<td>equation</td>
<td>Displayed equation</td>
</tr>
<tr>
<td>figure</td>
<td>Floating figures</td>
</tr>
<tr>
<td>flushleft</td>
<td>Flushed left lines</td>
</tr>
<tr>
<td>flushright</td>
<td>Flushed right lines</td>
</tr>
<tr>
<td>itemize</td>
<td>Bulleted lists</td>
</tr>
<tr>
<td>letter</td>
<td>Letters</td>
</tr>
<tr>
<td>list</td>
<td>Generic list environment</td>
</tr>
<tr>
<td>math</td>
<td>In-line math</td>
</tr>
<tr>
<td>minipage</td>
<td>Miniature page</td>
</tr>
<tr>
<td>picture</td>
<td>Picture with text, arrows, lines and circles</td>
</tr>
<tr>
<td>quotation</td>
<td>Indented environment with paragraph indentation</td>
</tr>
<tr>
<td>quote</td>
<td>Indented environment with no paragraph indentation</td>
</tr>
<tr>
<td>tabbing</td>
<td>Align text arbitrarily</td>
</tr>
<tr>
<td>table</td>
<td>Floating tables</td>
</tr>
<tr>
<td>tabular</td>
<td>Align text in columns</td>
</tr>
<tr>
<td>thebibliography</td>
<td>Bibliography or reference list</td>
</tr>
<tr>
<td>theorem</td>
<td>Theorems, lemmas, etc</td>
</tr>
<tr>
<td>titlepage</td>
<td>For hand crafted title pages</td>
</tr>
<tr>
<td>verbatim</td>
<td>Simulating typed input</td>
</tr>
<tr>
<td>verse</td>
<td>For poetry and other things</td>
</tr>
</tbody>
</table>

Table 7: Standard \LaTeX environments.
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,

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

5.3.3 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:

\begin{quote}
\newenvironment{margins}{\begin{list}{}{\setlength{\leftmargin}{#1} \setlength{\rightmargin}{#2}} \item }{\end{list}}
\begin{margins}{0.5cm}{1cm}
As you can see, this paragraph 
has normal margins. 
But please note that this 
paragraph has custom margins.
\end{margins}
\end{quote}

Then you will use the new environment:

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

5.3.4 Paragraph/Indentation

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

\begin{quote}
\setlength{\parindent}{1cm}
\end{quote}

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

\begin{quote}
\setlength{\parskip}{3pt}
\end{quote}

5.3.5 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`:
this is a framed paragraph!

this is a framed paragraph!

this is a shaded paragraph, do you like it?

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. \width sets the width of the minipage equal to that of 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
```

Modifying the parameter, you can adjust the frame width:

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

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

### 5.3.6 Paragraph/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}
}
```

Just as an example, we set the minipage colour for only the 80% of its width. More about colours in Section 5.2.5.
5.3.7 Format/Columns

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

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

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

6 The Table Menu

Quite a complex subject... A table is a float (as explained in Section 4.12) that must fit on one page. It usually contains a \texttt{tabular} environment, even though other possibilities exist. By default, a table adjusts its width to match the width of its contents.

Let me stress that the \texttt{table} environment is a float, but \texttt{tabular} is not. Keep this in mind if you want to write informal tabular material, i.e. without label and caption.

This is the general format of a table:

\begin{verbatim}
\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|rp{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 & " \texttt{hfill right}\"
% \hfill right\
% & force a space with "
% row 5, col 1 & row 5, col 2 & row 5, col 3 & left \texttt{hfill }\"
% & \texttt{hfill centre}\"\"
% & \texttt{hfill}\"\
% \hline
% \caption{A sample table.}
% \label{tab:sampletab}
% \end{tabular}
% \end{center}
% \end{table}
\end{verbatim}

31
Table 8 shows the result.

Sometimes, a table is too wide and won’t fit on the page. In that case, the `rotating` package provides the new environment `sidewaystable`. Also, `rotating` makes it possible to rotate the contents of a cell by a specified angle. Finally, the `tabularx` package lets one specify tables of fixed width: the X column specifier indicates that a column can be spread as needed.

Here’s an example:

```
\begin{sidewaystable}
\begin{tabularx}{7.5cm}{|l|X|X|}
\hline
 normal & tilted & wider \\
\hline
 normal & \rotatebox{30}{I’m tilted!} & I’m wider \\
\hline
\end{tabularx}
\end{sidewaystable}
```

The standard `tabular` environment cannot span more than one page! There are some packages that overcome this limitation: you will want to try out `longtable`, `supertabular`, and `xtab`.

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

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

Colour by row:

```
\begin{tabular}{lll}
one & two & three \\
\hline
one & two & three \\
\hline
one & two & three \\
\hline
\end{tabular}
```
To conclude the subject, a neat little trick. If you think that writing \LaTeX\ tables is too complicated, you could be relieved by OpenOffice Calc and Calc2LaTeX. The former is the well-known free spreadsheet, while the latter is a plugin that lets you turn a cell range into a \LaTeX\ table. Links: http://www.openoffice.org/, http://calc2latex.sourceforge.net/.

### 6.1 Table/Line Spacing

A line adjusts itself to the height of the text it contains. To add some space before a line, the trick is to start it with a \textbf{\textbackslash rule} of 0 length and specified height. To add space after a line, use \textbf{\textbackslash} followed by optional space. Here is an example:

\begin{tabular}{lll}
one & two & three
0.3 centimeters & \textbf{after} & \textbackslash this line\\
one & two & three
one & two & three
\textbackslash rule\{0pt\}\textbackslash\{1.2cm\}1.2 centimeters & \textbf{before} & \textbackslash this line\\
\end{tabular}

### 6.2 Table/Rule Width

\begin{tabular}{|lll|}
one & two & three
\hline
one & two & three
\%setlength\{\arrayrulewidth\}\{5pt\}
one & two & three
\hline
four & five & six
\%setlength\{\arrayrulewidth\}\{1pt\}
\hline
\end{tabular}

### 6.3 Table/Aligning Numbers

A special case of a tabular environment is when we want to align numbers with respect to the decimal positions.

The simplest method is using the \textbf{\$} column specifier, which in practice is useful in tables.

\begin{tabular}{|lll|}
one & two & three
\hline
one & two & three
four & five & six
\hline
\end{tabular}
containing only numbers. The column separator \& is replaced by the decimal dot:

\begin{tabular}{r@{.}l}
  3&14159\ 
  1&61803\ 
  1&41421\ 
  100&00000
\end{tabular}

Alternatively, 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, \textquote{.}), 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 \verb|mbox|.

\begin{tabular}{|D{.}{,}{4.2}|D{.}{.}{5}|D{.}{.}{-1}|}
\hline
\verb|\mbox{One}| & \verb|\mbox{Two}| & \verb|\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}

\subsection{Using slashbox}

This package add the \verb|\backslashbox| command:

\begin{tabular}{|l|l|l|}
\hline
\verb|\backslashbox[2cm]{Lesson}{Date}| & Monday & Tuesday\ 
Stratigraphy & room A & room A\ 
Chemistry & room B & Lab $\alpha$\ 
Physics & room C & Lab $\beta$\ 
\hline
\end{tabular}

\subsection{Importing Data in \LaTeX Tables}

For many people, data files are the bread and butter of everyday's work. Most data files are simply ASCII text with columns of numbers, but some people use spreadsheets. Nearly all spreadsheet applications can export sheets in the ASCII-based .csv file format; values are usually separated by the \textquote{;} character.

Converting a data file into a \LaTeX table is quite a tedious process. The following script for UNIX will convert a datafile with an arbitrary number of columns to a table. It will also work on .csv files.
#!/bin/sh

# dat2tex.sh: converts tabular data to a tabular environment

if [ $# != 1 ]; then
  echo "Usage: $0 <datafile>"
  exit 1
fi

# is this a csv file?
grep ";" $1 > /dev/null
if [ $? = 0 ]; then
  AWK="awk -F;"
else
  AWK=awk
fi

# ok awk, make my day
$AWK 'if (1 == FNR) {
   printf "\begin{tabular}{l}\nfor (i = 1; i <= NF; i++) {printf "1"; 
   printf "}\n" }
   for (i = 1; i < NF; i++) \n   {printf $i " & "} printf $NF"\\ \n"}
END {printf "\\end{tabular}\n"'} $1

# end of dat2tex

7 The Tools

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: \texttt{Name}, \texttt{Surname}, and \texttt{Title}. Their values will be gathered from an external file, \texttt{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,\nmay I call you \Name{}?\nYours,\n}\hspace{3cm}Guido\clearpage
\end{document}

The fourth field, \texttt{\Void}, isn’t really necessary and it’s there for illustration. 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.

Marie
Curie
Mme

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\times8 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
7.3 Tools/Default Language

\documentclass{article}
\usepackage{labels}
\begin{document}
\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 \texttt{textmerge} and labels!

\begin{itemize}
\item \texttt{\usepackage[italian,english]{babel}}
\end{itemize}

\texttt{babel} alters the way some characters behave in a language-dependent way. If you experience odd problems, insert the offending characters using the \texttt{\charXX} syntax.

In addition, to type accented letters and in general non-standard ASCII characters\footnote{In computer jargon, ‘standard ASCII characters’ are the characters whose code is included between 32 (space) and 126 (tilde).}, you may want to use the packages \texttt{inputenc} and \texttt{fontenc}. In most cases, UTF-8 is the right choice: remember to enable it in your editor too!

\end{document}
A different way of inserting accented letters is configuring your editor to type those for you. For example, I set up my editor of choice (jed) to have it insert \'e whenever I type ‘é’. I included this in my .jedrc:

```c
#define latex_mode_hook ()
{
    set_abbrev_mode (1);
    if ( () = abbrev_table_p ("LaTeX") )
        use_abbrev_table ("LaTeX");
#endif
    local_setkey (" \‘a", "à");
    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{} usually does a good job at hyphenating words, sometimes manual intervention may yield better results. Manual hyphens 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 put it in \mbox:

Do not hyphen \mbox{internationalisation}, please. I’m a masochistic.

### 7.5 Tools/Spell Check

\TeX{} is not aware of spell spelling. This task is done using external tools like ispell, aspell or others. Under UNIX, you can use ispell this way:

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

The `-t` switch instructs ispell to ignore \TeX{} and \TeX{} commands. If your language is not 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.ctan.org/tex-archive/info/LatexHelpBook/ is a very nice help system for \LaTeX{}, fully integrated with Windows.
- don’t forget the http://groups.google.com/group/comp.text.tex/topics newsgroup: it’s an invaluable source of help.

As of 2015, most GNU/Linux distributions ship with TeXLive, probably the most complete \TeX{}/\LaTeX{} systems. A lot of documentation is provided; on my Ubuntu machine, it’s found in /usr/share/doc/texlive-doc/.

9 The End

This document is copyleft © Guido Gonzato, 2001–2015, and released under the GNU Free Documentation Licence. 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 \texttt{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 5: Book template.
This is the final report.

The report starts here.
Figure 8: 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; height 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 the 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 (6,4) 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 9: How to write a poster.