LaTeX for Word Processor Users
version 1.2

Guido Gonzato, PhD
guido.gonzato at gmail.com

2023-08-28

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.

Contents

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 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 perceived to be easier than LaTeX since 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, and unusable. People who had to write long texts with lots of tables and figures know what I'm talking about.

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 word processor.

That's why I wrote this quick reference, which is meant to be a concept-by-concept guide. As I said, it assumes some basic LaTeX knowledge; if it's not the case, I suggest that you have a look at one of the following guides:

- The (Not So) Short Introduction to LaTeX2e
- Learn LaTeX in 30 minutes
- https://en.wikibooks.org/wiki/LaTeX/

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 1.1.3.

Packages and other TeX-related material are available at many sites that constitute the Comprehensive TeX Archive Network (CTAN), whose web site is https://www.ctan.org.

From now on, CTAN: means 'your favourite CTAN mirror here, starting from the TeX directory'. For instance, you can get this document from CTAN://latex4wp, which corresponds to https://www.ctan.org/pkg/latex4wp.

To write your documents, you will need a good UTF-8 enabled text editor. A better choice for beginners is a *LaTeX IDE*: an editor dedicated to writing LaTeX source, with output preview and many other facilities.

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

- TeXstudio (multiplatform; GNU/Linux AppImage and portable app for Windows available): https://www.texstudio.org
- TeXworks (multiplatform; GNU/Linux AppImage available): https://tug.org/texworks/ (included in TeX Live)
- Texmaker (multiplatform): https://www.xm1math.net/texmaker/index.html
- TeXShop (Mac OS X): https://www.uoregon.edu/koch/texshop/
- TeXnicCenter (Windows): https://www.texniccenter.org/

Finally, I assume modern LaTeX usage, i.e. PDF output using `pdflatex` or `xelatex` or `lualatex`. The same goes for graphics inclusion; no PostScript or EPS files will be considered.

1.1.1 Typographic Conventions

Throughout this document, I'll be using the facilities provided by the CTAN://latexdemo package to typeset LaTeX snippets along with their output. Occasionally, `latexdemo` does not work properly; in those cases, I'll be using a local modification of the old CTAN://example package.

1.1.2 Editor-Supported Features

LaTeX is just a typesetter: 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.

1.1.3 Adding Packages

The following information applies to TeX Live, a comprehensive TeX system that ships with most GNU/Linux distributions. It should also work on MacTeX, but I have no first-hand experience. Instructions for MiKTeX (probably the most popular implementation for Windows) will follow below.

A large collection of LaTeX packages is natively supported; for instance, Debian-based GNU/Linux variants provide lots of `texlive-*` packages. Please note the ambiguity: we're talking of `.deb` *packages* containing LaTeX *packages*.

Action	**Emacs**	**Vim**	**Jed**
command mode	`Alt-X`	`ESC`	`Alt-X`
insert mode	n/a	`i a o O`	n/a
line editor mode	n/a	`:`	n/a
file operations			
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`
moving around			
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`	`$`	`Ctrl-QD`
page up	`Alt-V`	`Ctrl-U`	`Ctrl-R`
page down	`Ctrl-V`	`Ctrl-D`	`Ctrl-C`
start of buffer	`Alt-<`	`1G`	`Ctrl-QR`
end of buffer	`Alt->`	`G`	`Ctrl-QC`
line n.	`Alt-G n.`	`n.G`	`Ctrl-QI`
deleting			
character left	`Ctrl-H`	`X`	`BS`
character right	`Ctrl-D`	`x`	`Alt-G`
word left	`Alt-DEL`	`db`	`Alt-BS`
word right	`Alt-D`	`dw`	`Ctrl-T`
end of line	`Ctrl-K`	`d$`	`Ctrl-QY`
line	`Ctrl-A Ctrl-K`	`dd`	`Ctrl-Y`
search & replace			
search	`Ctrl-S text`	`/text`	`Ctrl-QS`
replace	`Alt-%`	`:s/old/new/g`	`Ctrl-QA`
blocks			
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: Useful key bindings for Emacs, Vim, and Jed in IDE mode.

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, you may use the MiKTEX console or a command line procedure. 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:

```
\usepackage{foo}
```

1.1.4 Adding the Info Page

'Man' and 'Info' pages are command-line sources of documentation for software, widely employed in UNIX and GNU/Linux systems. If your LATEX distributions lacks the `latex2e.info` info page, do the following:

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

2. run these commands:

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

Now, `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 is designed to produce output of the highest possible quality. The less you mess with formatting parameters, the better;
3. that said, there are times when you do want to break the rules. If you know what you are doing, that's fine. This guide breaks the rules a lot.

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

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}
\thispagestyle{empty} % no page number
\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--2023.
\end{thebibliography}
\end{document}
```

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:

- `TeX4ht` is probably the best LaTeX to HTML/XML converter:
  https://tug.org/tex4ht
- `latex2html`, another converter to HTML:
  CTAN://latex2html
- `latex2rtf`, a converter to Rich Text Format:
  CTAN://latex2rtf
- `detex` (a command line converter) removes all LaTeX tags and outputs plain text:
  https://github.com/pkubowicz/opendetex,
  CTAN://detex/

But please read Section 2.4 to learn what `Pandoc` can do fo you. See also Section 2.6 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

This is taken from the home page of `Pandoc`, https://pandoc.org:

> If you need to convert files from one markup format into another, pandoc is your swiss-army knife.

`Pandoc` is an outstanding converter that produces excellent LaTeX output, and much more. It's an invaluable tool when you need to write (or convert) documents in multiple formats.

`Pandoc`'s native format is an extended version of Markdown, which translates smoothly into good LaTeX. For example, this is a document written in Markdown:

```
---
title: |
  This is the title: \
  now write the rest
author: Guido Gonzato
date: August 2023
abstract: |
  This is the abstract.
...

<!--- This is a comment. -->

[comment]: # (This too is a comment)

# Section

This is **bold text**, this is *emphasized text*, this is normal text.

If you're bored, go to [The End].
```

```
## Subsection

This is `verbatim text`.

# The End

My dad used to say:

> Damn, Pandoc has not been invented yet!

but not it has.

<!--- end of pandoc_template.md -->
```

To turn this document into a standalone LaTeX source, type:

```
pandoc -s pandoc_template.md -o pandoc_template.tex
```

To typeset this source to PDF using LaTeX, type:

```
$ pandoc --toc \
  -V urlcolor=blue -V toccolor=red \
  -V geometry:margin=2cm \
  pandoc_template.md -o pandoc_template.pdf
```

The *-V* lines are used to specify options; please consult `Pandoc`'s documentation.

You can use `Pandoc` to convert lots of different formats to LaTeX. For example:

```
$ pandoc -s index.html -o index.tex
$ pandoc -s text.docx -o text.tex
$ pandoc -s text2.rtf -o text2.tex
$ ...and so son.
```

The *-s* flag means "standalone", i.e. the output file is a complete LaTeX document.

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 the `\setlength` command, which is used to change the value of variables and counters. It's better to use the geometry package, which 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.

```
\usepackage{geometry}  % top of document
...
\geometry{paperwidth=25cm}
\geometry{paperheight=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{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 (or `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:

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

2.6 File/Print Preview

That's very simple: produce a PDF file directly with `pdflatex`, `xelatex`, or `lualatex`, then use your favourite PDF viewer.

Some packages like hyperref and url make the resulting PDF file browseable; see Section 4.18. However, using `pdflatex` you might experience problems with other packages. More details in Section 4.14.

2.7 File/Print

Simply use the File/Print menu entry of your PDF viewer..

2.8 File/Versions

Several programs for revision control and collaborative writing are available. While old-style geeks may want to use a single-user tool like RCS, more powerful and multi-user tools like Subversion, Git, Mercurial etc. are preferable. Git is probably the most commonly used, and some documentation about LaTeX integration is here:

- https://www.desy.de/~bargheer/gitintro/git.html

- https://www.math.cmu.edu/~gautam/sj/blog/20130929-git-quickstart.html

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:

```
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:

```
\newcommand{\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 ˜ (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
Another line.\\
Ok, now you get it.
```

I am stretched!
But I am not.
Another line.
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 4.14.

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}
```

* with an asterisk;

- with a dash;

. with a dot.

Another way is to redefine the style of the *counters*[1] that 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}
\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

Alternatively, use the enumerate package. It redefines the `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) `\Alph`, `\alph`, `\Roman`, `\roman`, or `\arabic`. To include

---

[1] each text element that is numbered by LaTeX (sections, lists, figures, etc.) has a counter associated with it.

some text, enclose it in braces:

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

Example I. First example.

Example II. Second example.

Example III. Last example. Go to Item I.

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}
```

3. Example 3.
4. Example 4.
6. Example 6.

Lists within paragraphs are implemented by the paralist package, which provides the `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}
  [\itshape a) \upshape]
  \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. paralist can do much more, and I suggest that you read its documentation.

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 `\char` command. For example, you could insert the characters $&^~ entering `\char36` `\char38` `\char94` `\char126`.

There are packages that provide thousands of unusual characters and symbols. For instance, pifont provides the commands `\ding`, `\dingfill`, `\dingline`, and `\dinglist`. The first command produces the Dingbat character of the specified code. The other commands are equivalent to the

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

Table 2: How to obtain some special characters.

`\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

Another nice variant is this:

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

① one

② two

③ three

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

4.3.1 The € Sign

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

```
\usepackage[gen]{eurosym}
\usepackage[official]{eurosym}
```

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

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

4.4 Insert/Formula

LaTeX is particularly strong at typesetting math. To insert math symbols in the main text (*inline mode*), you must enclose them between $:

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

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

The environments `displaymath` and `equation` typeset formulae aside from the text; this is called *display mode*. `equation` adds an equation number for later reference:

```
Fermat's Last Theorem is
defined as:
\begin{equation}
x^n + y^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:

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

Can you prove Eq. 1?

4.5 Using SymPy

Programming languages and spreadsheets use a standard syntax to enter math formulae. Using a great Python library called SymPy, you can get LaTeX formulae from math expressions. You don't have to be proficient in Python; just use the standard math syntax.

This is a sample SymPy session. We want to obtain the LaTeX code for the expression $\left(\frac{x}{2}-\frac{2y}{3}\right)^3$ and for its expansion $\frac{x^3}{8}-\frac{x^2y}{2}+\frac{2x}{3}y^2-\frac{8y^3}{27}$:

```
In [1]: from sympy import *

In [2]: init_session()
IPython console for SymPy 1.9 (Python 3.10.6-64-bit) (ground
types: python)

These commands were executed:
>>> from __future__ import division
>>> from sympy import *
>>> x, y, z, t = symbols('x y z t')
>>> k, m, n = symbols('k m n', integer=True)
>>> f, g, h = symbols('f g h', cls=Function)
>>> init_printing()

Documentation can be found at https://www.sympy.org

In [3]: expr = (x/2 - 2*y/3)**3

In [4]: latex(expr)
Out[4]: '\\left(\\frac{x}{2} - \\frac{2 y}{3}\\right)^{3}'

In [5]: latex(expr.expand())
Out[5]: '\\frac{x^{3}}{8} - \\frac{x^{2} y}{2} + \\frac{2 x}{3}
y^{2} - \\frac{8 y^{3}}{27}'
```

Now you can copy and paste the resulting formulae into your editor, only changing \\ to \.

If you are reasonably proficient in Python, you may be interested in the python LaTeX package and the LaTeX Expression project module.

4.6 Insert/Footnote

The command `\footnote[n]{footnote text}` is all you need; the optional parameter `[n]` modifies the footnote number. The `\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!}
This footnote\footnote
{I mean this one.}
says it all.
```

This footnote[read me!] says it all.

[read me!] I mean this one.

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!}
```

> 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: * † ‡ § ¶ ‖ ** †† ‡‡

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.

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

4.6.1 Footnotes and Endnotes

The endnotes package provides a new command, `\endnote`, which you use instead of `\footnote` to write footnotes that are printed after a `\theendnotes` command.[1] This is useful when you want to print footnotes at the end of a chapter or section.

You can also move all footnotes at the end of the document. You'll have to add this line to the preamble:

```
\let\footnote=\endnote
```

and add these lines as the last thing in your document:

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

Notes

[1] This is an endnote.

4.7 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
```

4.8 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 simple 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: 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 `\hspace` like in this example:

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

```
This is a              2-cm-wide hole.
```

4.9 Insert/Tabs

The `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:

Command	**Action**
`\=`	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

This example shows some of the available commands:

```
\begin{tabbing}
% let's set the tab positions
~ \hspace{1cm} \= ~ \hspace{1.7cm} \=
~ \hspace{2.2cm} \= \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{\dots} \= new tab 2 \\
new \> tab \\
\poptabs  % restore tab positions
Zero \> One \> Two \> Three
\end{tabbing}
```

```
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.

4.10 Insert/Cross Reference

The commands `\label`, `\ref`, and `\pageref` are all you need to insert labels in the text and do cross referencing. The standard format of labels is 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 can be obtained 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 4.10 on page 17.

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

4.11 Insert/Margin Notes

Very simple: use `\marginpar{text}`. This feature is not commonly used with word processors. *Note!*

4.12 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 textpos package, as shown in the example listed in fig. 10 (see Appendix A).

If you don't need to specify a fixed position, then use the `minipage` (miniature page) environment. This text is enclosed in a minipage…

…and this is some other text enclosed in a `boxedminipage` environment, provided by the package of same name.

I remind you what a `minipage` declaration looks like:

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

In a `boxedminipage`, the space between the frame and the text is set with this command:

```
\setlength{\fboxsep}{5mm}
```

4.13 Insert/Image

Please note: an *image* and a *figure* are not the same thing! Please see Section 4.14 for details.

Images can be inserted as PDF, JPG, PNG, and TIFF files. If you need to insert PostScript or EPS files, then you're a geek and you surely know how to convert these formats to PDF.

The almighty graphicx package provides the `\includegraphics` command:

```
This is a lovely picture

\includegraphics[width=3cm]{piper.jpg}

of a guy playing the pipes.
```

This is a lovely picture

of a guy playing the pipes.

Parameters `height`, `scale`, and `angle` can also be specified. `width` can be set to exact units as above, or related to text or line width:

```
width=\textwidth     % width of text on page
width=\linewidth     % width of text in current environment
width=0.5\linewidth  % half of line width
```

To centre or right-align an image, use the adjustbox package and add this line to your document:

```
\usepackage[export]{adjustbox}
```

The adjustbox package adds options `left`, `right`, `center`, `outer`, and `inner` to `\includegraphics`. The latter two options apply to multi-column text.

Needless to say, bitmap images don't have the same quality as PDF vector images; besides, they can make the resulting output file quite large.

4.14 Insert/Figure

Inserting a *figure* is not the same as inserting an *image*. In fact, a figure is not necessarily an image; it has no fixed position on the page; it usually has a caption; and it can be referenced to. All this is accomplished with the `figure` environment. Two examples of figure follow.

```
\begin{figure}[htbp]
% [htbp] specifies the preferred placement:
% here, top, bottom, or separate page.
  \centering
  \texttt{=8-)}
  \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'; use '`see fig.~\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 here package that provides an optional placement argument `H` (meaning, "right HERE!").

To include figures side by side, use the subcaption package that provides the `subfigure` environment (after Figure 2):

```
\begin{figure}[htbp]
\centering
\includegraphics%
[width=0.7\textwidth, angle=-90]%
{gnuplot.pdf}
\caption{A Gnuplot graph.}
\label{fig:gnuplot}
\end{figure}
```

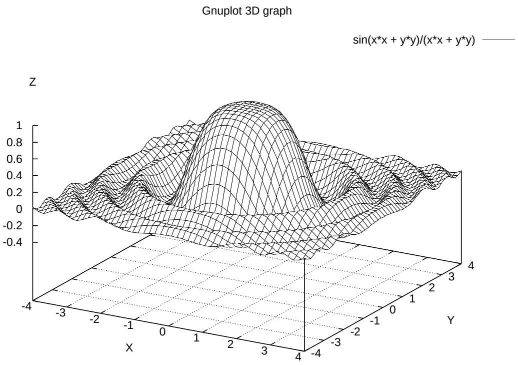
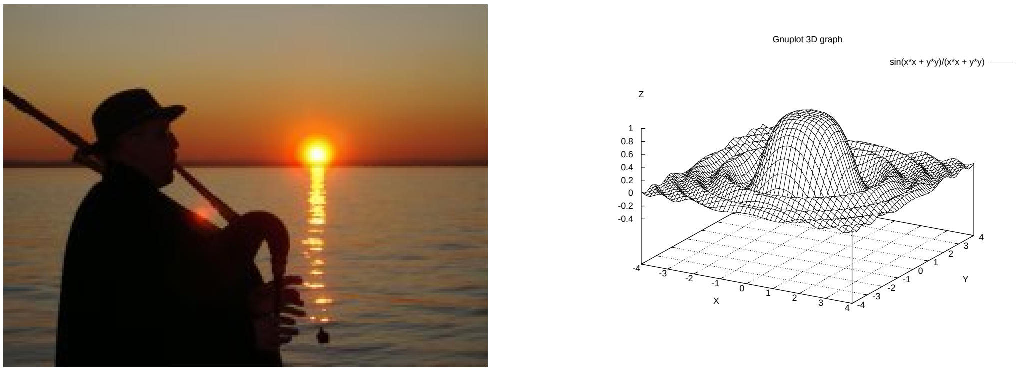


Figure 2: A Gnuplot graph.

```
\begin{figure}[h]
  % first subfigure
  \begin{subfigure}{0.5\textwidth} % half figure for 1st subfigure
    \includegraphics[width=0.9\linewidth]{piper.jpg}
    \caption{Caption of 1st subfigure.}
    \label{fig:subfig1}
  \end{subfigure}
  % second subfigure
  \begin{subfigure}{0.5\textwidth}
    \includegraphics[width=0.9\linewidth, angle=-90]%
    {gnuplot.pdf}
    \caption{Caption of 2nd subfigure.}
    \label{fig:subfig2}
  \end{subfigure}
  % whole figure
  \caption{Caption of the figure with two images.}
  \label{fig:image2}
\end{figure}
```



(a) Caption of 1st subfigure.

(b) Caption of 2nd subfigure.

Figure 3: A figure with two subfigures.

4.14.1 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 clear 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?
```

If you meet this guy, give him some money.

=8-)

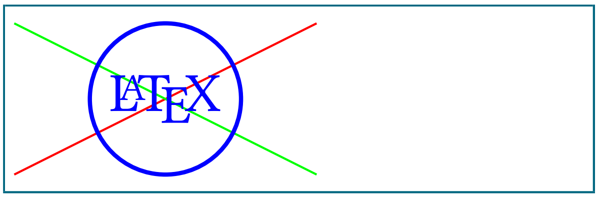
The reason may not be clear to you, but I can assure that your money will end up in good hands. I say again, if you meet this guy, give him some money: he knows how to use it properly. OK?

The parameters are the number of lines to be narrowed, the figure placement (like in `htbp`, the overhang, and the figure width.

4.15 Insert/Shapes

PGF/TikZ PGF is a powerful package for generating graphics programmatically. This is a minimal example:

```
\begin{tikzpicture}
\draw[red,thick] (-2,-1) -- (2,1);
\draw[green,thick] (-2, 1) -- (2,-1);
\draw[blue,ultra thick] (0, 0) circle
  (1cm) node{\huge \LaTeX};
\end{tikzpicture}
```



but you can do miracles, as shown at the TikZ.net page.

If you prefer to use a WYSIWYG graphics program, I recommend that you start with Inkscape, https://inkscape.org/. This program is LaTeX aware, i.e. it can interact with LaTeX to typeset formulae and text.

Start Inkscape and draw any shape you wish using its tools. To insert text rendered by LaTeX, select Extensions/Text/Formula (pdflatex)... or Extensions/Render/Mathematics/LaTeX (pdflatex)..., insert your text as in fig. 4, 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 (Figure 5). The resulting picture can be exported to several formats supported by LaTeX, such as PDF, PNG, and many others.

Many graphics programs provide LaTeX-friendly output, both programmatically and interactively. The first that spring to mind are:

- GLE (Graphics Layout Engine) is a graphics scripting language designed for creating publication quality figures. […] GLE relies on LaTeX for text output and supports mathematical formulae in graphs and figures:
  www.gle-graphics.org
- Asymptote is a powerful descriptive vector graphics language that provides a natural coordinate-based framework for technical drawing. Labels and equations are typeset with LaTeX:

https://asymptote.sourceforge.net/

- LaTeXDraw is a graphical drawing editor for LaTeX:
  https://latexdraw.sourceforge.net/

- Graphviz takes descriptions of graphs in a simple text language, and make diagrams for inclusion in LaTeX:
  https://graphviz.org.

These packages let you make publication-quality 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, https://www.tug.org/texshowcase/.

4.16 Insert/Page

Inserting external PDF pages without producing overfull errors is easily done with the pdfpages package. The following command:

```
\includepdf[fitpaper]{pandoc_template.pdf}
```

includes the PDF document that was made by `Pandoc` (Section 2.4).

This is the title:
now write the rest

Guido Gonzato

January 2018

Abstract

This is the abstract.

Contents

Section

This is **bold text**, this is *emphasized text*, this is normal text. If you're bored, go to The End.

Subsection

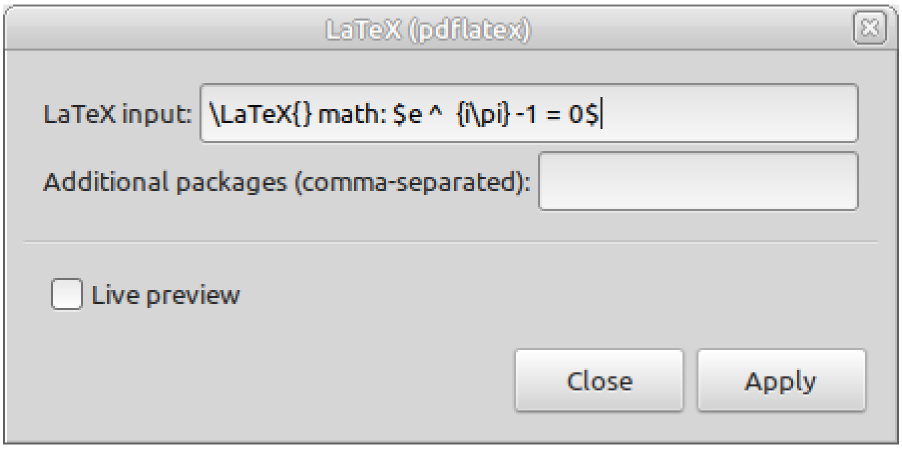
This is `verbatim text`.

The End

My dad used to say:

> Damn, Pandoc has not been invented yet!

but not it has.



LaTeX input: \LaTeX{} math: $e ^ {i\pi} -1 = 0$

Additional packages (comma-separated):

☐ Live preview

Figure 4: Inserting a LaTeX formula.

Obviously, pdfpages can do much more and its documentation is a good read.

4.17 Insert/Rule

Draw lines of any length 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 \euro~10
```

Total price . . . . € 10

4.18 Insert/Hyperlink

The hyperref package lets you write URLs and other external references, making the PDF output browseable. For instance, this document uses this declaration:

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

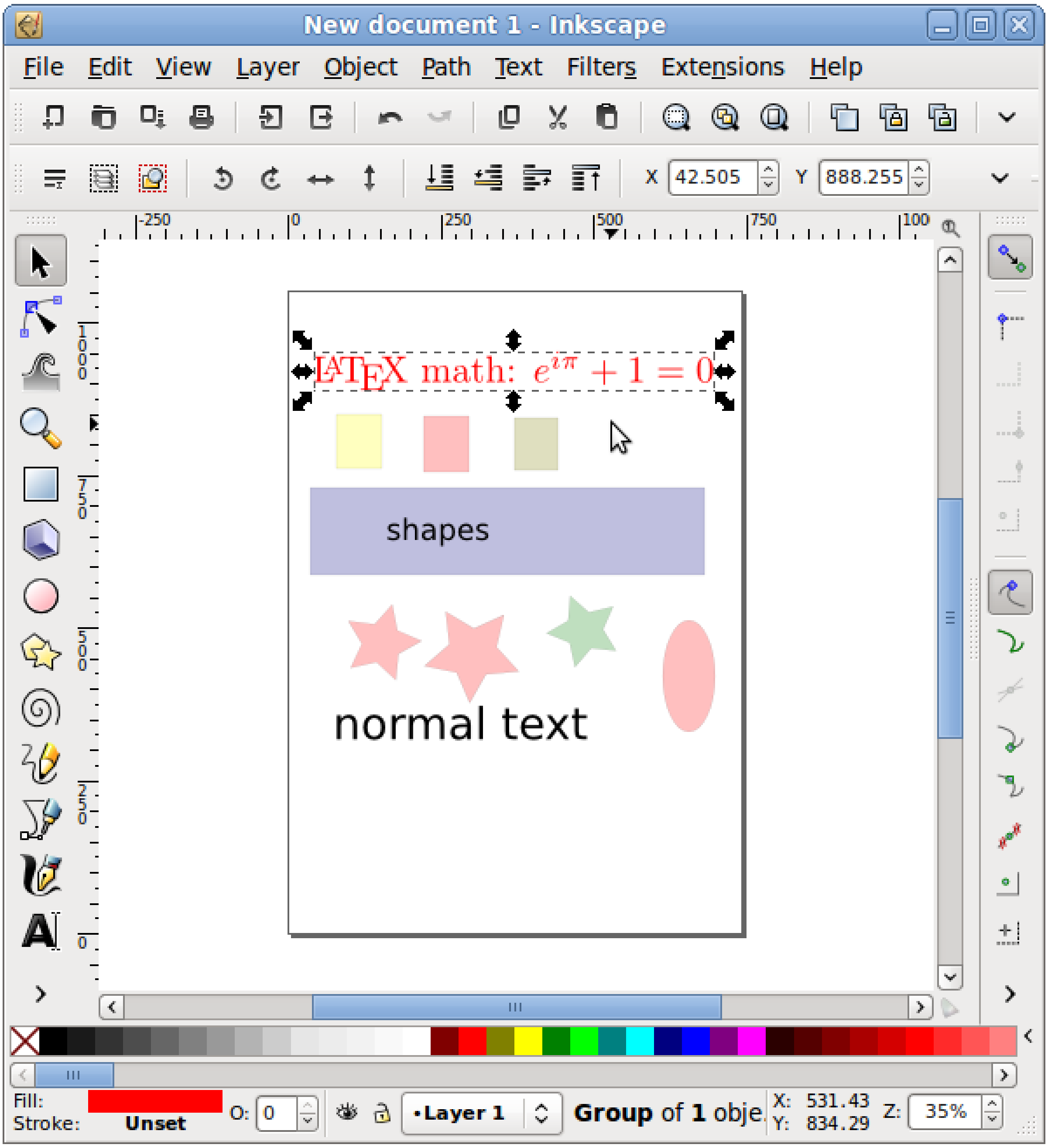

New document 1 - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X 42.505 Y 888.255
LaTeX math: $e^{i\pi}+1=0$
shapes
normal text
Fill:
Stroke: Unset
O: 0
•Layer 1
Group of 1 obje
X: 531.43
Y: 834.29
Z: 35%

Figure 5: A LaTeX object can be edited as desired.

Let's see an example:

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

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

Click \hyperlink{ctan}{here} to go
back to the top.
```

The CTAN main site is https://www.ctan.org, a.k.a CTAN://.
Listen to this MIDI file.
Click here to go back to the top.

By default, the `\url` command typesets its contents using a monospace font. To use the same font as the remaining text, use the command:

```
\urlstyle{same}
```

after the `\hyperref` declaration, as this document does.

The `\hypertarget` and `\hyperlink` commands provide internal links, just like HTML; `\href` creates links to URLs or external files. Note the `run:` parameter: you can run external programs like multimedia players, office applications, whatever. This feature only works with some PDF readers, though.

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

4.19 Insert/Comment

This is done inserting % before each line, or by using the comment package that provides `\comment`, `\endcomment`, and the `comment` environment:

```
This text % boring example
\begin{comment}
what a boring example
\end{comment}
is just an example.
```

This text is just an example.

Pop-ups comments in PDF output are provided by the pdfcomment package. This document uses this declaration:

```
\usepackage[icon=note,color={1 1 0}]{pdfcomment}
```

```
This text is nothing special
\pdfmargincomment{yup, definitely.},
but it contains a couple of \
pdfmarkupcomment{nice}%
{simple, more than nice} comments.
```

This text is nothing special, but it contains a couple of nice comments.

Please note that not all PDF viewers can display pop-up comments correctly.

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 5.2.3.

5.1 Format/Line Spacing

The setspace package provides 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\\
are spaced normally.
\end{spacing}
```

These two lines

are crazily spaced!

Much better, these lines
are spaced normally.

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.

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

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

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

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, but the mhchem package provides a simpler command. Digits are printed as subscripts by default, and are printed as superscript when preceded by ^. Formulae must be enclosed in the `\ce` command:

```
\ce{H2O + CO2 -> H2CO3}\\
\ce{CaCO3 -> Ca^2+ + CO3^2-}\\
\ce{CO3^2- + H2CO3 -> 2 HCO3^-}\\
\ce{CaCO3 + H2CO3 -> Ca^2+ + 2 HCO3^-}
```

$H_2O + CO_2 \longrightarrow H_2CO_3$
$CaCO_3 \longrightarrow Ca^{2+} + CO_3{}^{2-}$
$CO_3{}^{2-} + H_2CO_3 \longrightarrow 2\,HCO_3{}^{-}$
$CaCO_3 + H_2CO_3 \longrightarrow Ca^{2+} + 2\,HCO_3{}^{-}$

Alternatively, you could use the chemformula package that uses a similar syntax.

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
`\textsuperscript`	n/a	this is $^{\text{superscript}}$
`\mathrm`	n/a	$\mathrm{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathbf`	n/a	$\mathbf{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathsf`	n/a	$\mathsf{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathtt`	n/a	$\mathtt{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathit`	n/a	$\mathit{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathnormal`	n/a	$\mathnormal{x^n + y^n \neq z^n \forall n \neq 2}$
`\mathcal`	n/a	$\S^\backslash + \dagger^\backslash \neq \ddagger^\backslash\forall\backslash \neq \in$

Table 3: Font attributes.

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 ulem package provides some fancy styles:

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

important urgent boat ~~wrong~~ ~~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 extsizes package 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 type1cm package, which provides commands like the following:

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 4: Font sizes.

Default font size	10pt	11pt	12pt
`tiny`	5	6	6
`scriptsize`	7	8	8
`footnotesize`	8	9	10
`small`	9	10	10.95
`normalsize`	10	10.95	12
`large`	12	12	14.4
`Large`	14.4	14.4	17.28
`LARGE`	17.2	17.28	20.74
`huge`	20.7	20.74	24.88
`Huge`	24.8	24.88	24.88

Table 5: Actual font size in pt.

```
\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}
```

1-cm tall

DROPPED capitals at the start of a paragraph can be obtained using the lettrine package, which provides a fully customisable `\lettrine` command. This paragraph uses the default behaviour:

```
\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 METAFONT subsystem. This ensures portability and yields very good results. However, many of us are accustomed to other fonts: Times, Helvetica, Sans Serif…

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.

Fortunately, several LaTeX engines can use PostScript Type 1 and Open Type OTF 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 results. This document uses libertine.

The only caveat is that LaTeX handles maths at its best only with Computer Modern fonts: using other fonts might render your formulas slightly less appealing.

A world of possibilities is available when you use `xelatex`, which supports Unicode, OpenType and TrueType fonts. Please visit the LaTeX Font Catalogue and enjoy.

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

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

This is Computer Modern Roman, this is Helvetica!

5.2.5 Format/Character Colour

You can colour words using the color package 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?

5.2.6 Format/Character Outline

If coloured text is not fancy enough, you can also have outlined (contoured) text using the contour package. The `\contour` command creates 16 copies of the text (or a number you specify):

```
\contourlength{0.5pt}
\Large
\textcolor{blue}{\contour{red}
{Blue text, thin red outline}}
\contourlength{2pt}
\textcolor{white}{\contour[32]{blue}
{White text, thick blue outline}}
```

Blue text, thin red outline
White text, thick blue outline

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}
```

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 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.

Environment	**Purpose**
`abstract`	abstract
`array`	Math arrays
`center`	Centered lines
`description`	Labelled lists
`displaymath`	Formulas on their own line
`document`	Encloses the whole document
`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
`math`	In-line math
`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.

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.
```

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 use `\null`, which sets 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 Paragraph/Margins

Normally, 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 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:

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

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

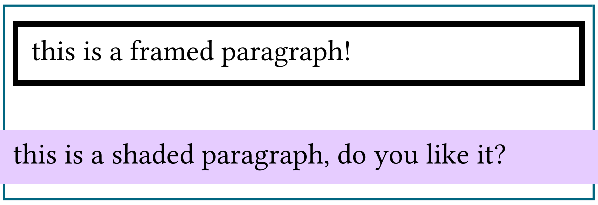
```
\setlength{\parskip}{3pt}
```

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 calc package 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

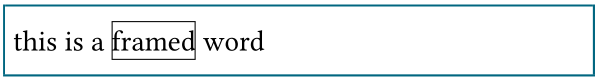
```
\framebox{
  \begin{minipage}[c]{\linewidth}
  text to be framed
  \end{minipage}
}
```

are functionally equivalent to the `boxedminipage` environment.

`\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:

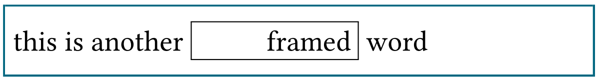
```
this is a
\framebox[\width]{framed}
word
```



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).

5.3.6 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}
}
```

I am a minipage, my colour is yellow!

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 `\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 multicols package 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

Welcome to hell—kind of.

Just like an *image* is not a *figure* (Section 4.13), a *`tabular` environment* is not a *table*. A LaTeX table is a float (as explained in Section 4.14) that must fit on one page; it has a caption, and it usually contains a `tabular` environment. The latter is typeset using the special syntax explained below.

Default LaTeX tabular material is defined as follows:

```
\begin{tabular}{|l|c r|} % left, centre,
right alignment
% three columns, items separated by (ugly
) vertical bars
\hline
1 & 2 & 3 \\ % end of line
\hline       % horizontal line
a &   & c \\ % second column is blank
one & two & three \\
\hline
\end{tabular}
```

1	2	3
a		c
one	two	three

where the definition specifies vertical bars, alignment, and number of columns. Each line in the tabular contains elements separated by &.

The booktabs package provides better-looking tabular material:

```
\begin{tabular}{lcr}
\toprule
1 & 2 & 3 \\
\midrule
a &   & c \\
one & two & three \\
\bottomrule
\end{tabular}
```

1	2	3
a		c
one	two	three

The following mess is the general format of tabular material:

```
% 4-column tabular; alignment is left, centered, right, fixed width
\begin{tabular}{lc|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 & ~ \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}
```

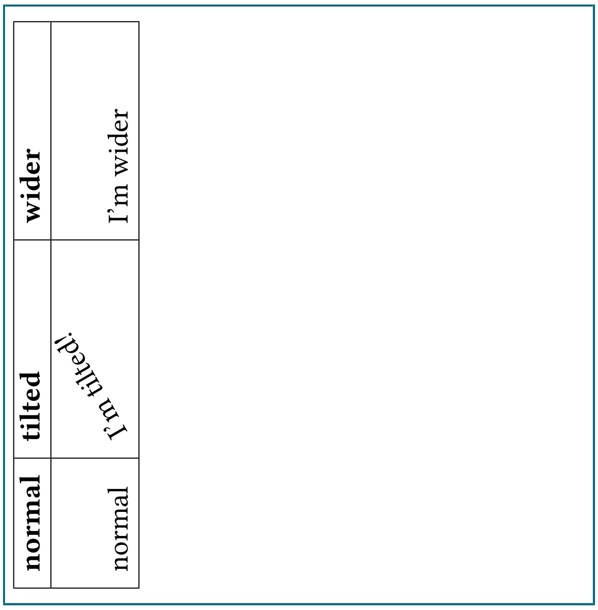
Left	**Centre**	**Right**	**4 cm**
row 1, col 1	row 1, col 2	row 1, col 3	row 1, col 4
row 2, col 1	row 2, col 2	row 2, col 3	row 2, col 4
spanning two columns		row 3, col 3	row 3, col 4
row 4, col 1	row 4, col 2	row 4, col 3	right
row 5, col 1	row 5, col 2	row 5, col 3	left
row 5, col 1	row 5, col 2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

Sometimes, a tabular is too wide and won't fit on the page. The first solution is using the longtable package, which provides a multi-page version of Pkgtabular. Besides, the rotating package provides the new environment `sideways` that can be used to rotate a tabular (or a single cell) sideways or by a specified angle.

The tabularx package lets you specify tabular columns of fixed width: the `X` column specifier indicates that a column can be spread as needed.

Here’s an example:

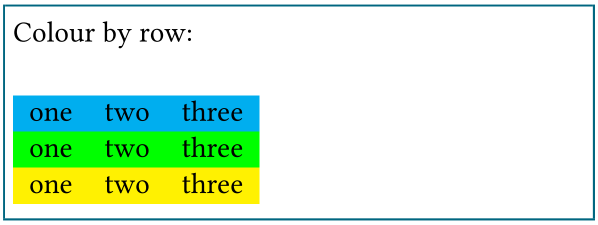
```
\begin{sideways}
  \begin{tabularx}{7.5cm}{|l|X|X|}
    \hline
    \textbf{normal} & \textbf{tilted} &
    \textbf{wider}\\
    \hline
    normal & \rotatebox{30}%
    {I'm tilted!} &
    I'm wider\\
    \hline
  \end{tabularx}
\end{sideways}
```



normal	**tilted**	**wider**
normal	I’m tilted!	I’m wider

Other useful packages are longtable, supertabular, and xtab.
To enable colours in tables, you use the colortbl package:

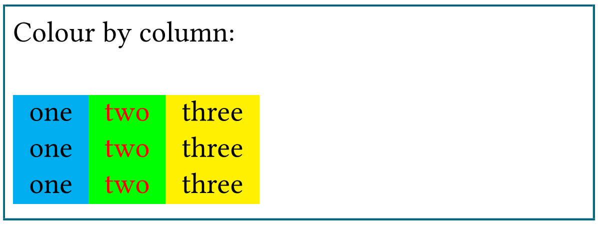
```
Colour by row:\\\vskip 2mm
\begin{tabular}{lcr}
  \rowcolor{cyan}
  one & two & three\\
  \rowcolor{green}
  one & two & three\\
  \rowcolor{yellow}
  one & two & three\\
\end{tabular}
```



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}
  one & two & three\\
  one & two & three\\
  one & two & three\\
\end{tabular}
```



Colour by column:

one	two	three
one	two	three
one	two	three

6.1 Table/Line Spacing

A table 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 `\rule` of 0 length and specified height. To add space *after* a line, use `\\` followed by optional space. Here is an example:

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

```
one              two      three
0.3 centimeters  after    this line

one              two      three
one              two      three


1.2 centimeters  before   this line
```

6.2 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 `@` column specifier, which in practice is useful in tables 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}
```

```
  3.14159
  1.61803
  1.41421
100.00000
```

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, '.'), 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}}
  \toprule
  \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\\
  \bottomrule
  \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
```

6.3 Using diagbox

This package add the `\diagbox` command:

```
\begin{tabular}{|l|l|l|}
  \toprule
  \diagbox{Lesson}{Date} &
  Monday & Tuesday\\
  \midrule
  Stratigraphy & room A & room A\\
  Chemistry & room B & Lab $\alpha$\\
  Physics & room C & Lab $\delta$\\
  \bottomrule
\end{tabular}
```

Date Lesson	Monday	Tuesday
Stratigraphy	room A	room A
Chemistry	room B	Lab $\alpha$
Physics	room C	Lab $\delta$

6.4 LaTeX Table Generators

Let's face it, inserting tabular material is quite a chore. Fortunately, there are web sites that allow you to create LaTeX code for tabular material in a visual way. Try these:

- https://www.tablesgenerator.com/
- https://www.latex-tables.com/
- https://tableconvert.com/latex-generator
- https://products.aspose.app/tex/latex-table-generator

6.5 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 ';' 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}{"; \
        for (i = 1; i <= NF; i++) {printf "l"}; \
        printf "}\n"
      }
      for (i = 1; i < NF; i++) \
        {printf $i" & "} printf $NF"\\\\ \n"} \
      END {printf "\\end{tabular}\n"}' $1

# end of dat2tex
```

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 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 3×8 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{GNU/Linux Sysadmin}}
  % now on to the boxed labels
  \boxedaddresslabel[\fboxsep=4mm\fboxrule=1mm]
  {\textbf{Guido Gonzato}, Ph.D.\\
  \textsl{GNU/Linux Sysadmin}}
\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`.)

Your LaTeX distribution contains a file called `language.dat` (usually `$TEXMF/tex/generic/con\-fig/language.dat` that contains a list of languages. Editing this file you choose the languages for which you want hyphenation patterns.

If you are not a native English speaker, you'll want to use the babel package as in the following example:

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

☞ babel alters the way some characters behave in a language-dependent way. If you get odd problems with some characters, try and insert them using the `\charXX` syntax.

In addition, to type accented letters and in general UTF-8 characters you may have to use enable UTF-8 in your editor. Besides, packages inputenc and fontenc were needed before 2018:

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

Since 2018, LaTeX defaults to UTF-8 input.

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 man.
```

7.5 Tools/Spell Check

LaTeX 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 LaTeX 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, https://www.ctan.org/tex-archive/info/.

- `info latex` (Unix systems) gives a concise but very complete on-line summary of commands and concepts;
- https://www.ctan.org/tex-archive/info/LatexHelpBook/ is a very nice help system for LaTeX, fully integrated with Windows.
- don't forget the https://groups.google.com/group/comp.text.tex/topics newsgroup: it's an invaluable source of help.

As of this writing, most GNU/Linux distributions ship with `TeXLive`, 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/`. Besides, the `\texdoc` command is very useful. For example:

```
shell> texdoc fancyvrb
```

will open the fancyvrb documentation.

9 The End

This document is copyleft 🄯 Guido Gonzato, 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 `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 6: Book template.

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

Figure 7: 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 8: 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 9: 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}
\thispagestyle{empty} % no page number

\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.pdf}
  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 10: How to write a poster.