

A Brief Introduction to \LaTeX

Allin Cottrell

revamped by Iñaki Amatria Barral

Universidade da Coruña – Autumn 2019

1 Introduction

\TeX , written by Donald Knuth, is a very powerful computer typesetting program. \LaTeX , originally written by Leslie Lamport, is a large set of macros (essentially, shortcuts) for \TeX . Most people use \LaTeX rather than “plain” \TeX . If you want to know, \TeX is pronounced “tech” rather than “tex”.

Typesetting with \LaTeX can get as sophisticated as you like—it can be used to produce printed copy that is better than most books being published these days—but we will stick to the basics here.

The basic idea with \TeX and \LaTeX is that you start with a plain text file containing the text you want to typeset. You indicate the way you want the typesetting done by means of certain special codes, most of which begin with a backlash (\backslash). When the text file is ready, you compile it using the \TeX processor, then preview the product on screen. If it is OK you can go ahead and print it, otherwise go back to the editing stage, re-compile the file, preview it again, and so on.

Superficially, \LaTeX looks more difficult to use than a standard word processor, because it is not WYSIWYG (“What you see is what you get”). As you are typing your basic text file with the typesetting commands included, you do not see what it will look like after compilation. But after a little practice you will find this is not really a problem. Also, when you get to the preview stage, “what you see” is a considerably more accurate representation of “what you’ll get” at the printer than with ordinary word processors. The great advantages of \LaTeX over word processors are:

- You don’t have to fuss with the detailed appearance of your document. You concentrate on (a) getting the text right and (b) the overall logical structure of the piece. \TeX takes care of the detailed formatting.
- The support for the typesetting of mathematics is unrivaled.
- Input files for \TeX are entirely portable. If I have a document in the form of a \LaTeX file I can email it to anyone who has access to the \TeX processor—whether on the PC, the Mac, or a Unix system—and they will be able to reproduce the same printed document.



2 Document skeleton

The skeleton of the \LaTeX document looks like this

```
\documentclass{article}

\begin{document}
Hello World!
This is my very first \LaTeX{} document.
\end{document}
```

Figure 1: A minimal \LaTeX file.

The first line specifies the type or class of the document, in this case an article. This is the class you are mostly likely to use (other classes include report, book, letter and beamer). The default print size for an article is 10-point type. If you find it a bit too small you can use the 11-point (or even the 12-point) option:

```
\documentclass[11pt]{article}
```

Beyond that, the `\begin{document}`, `\end{document}` pair constitutes the minimal set of special commands for a “valid” \LaTeX document. In between these commands goes your text.

Moving beyond the minimal involves learning a little about these topics:

- Document-structuring commands
- Typeface-changing commands
- Special characters
- \LaTeX math mode

We will examine these in turn.

3 Document-structuring commands

3.1 Lines and paragraphs

You needn’t pay much attention to the line breaks in your source file; \LaTeX will make its own decisions on where to break lines in order to produce properly justified text. The only important thing to remember is that if you want a paragraph break, insert a blank line, or in other words hit the Enter key twice in your source file.

3.2 Author, title, date

The title and author's name can be inserted at the beginning of the document thus

```
\title{Pride and Prejudice}
\author{Jane Austen}
\maketitle
```

Figure 2: Author, title declaration.

Coding the elements in this way will automatically put them into a centered format, using a larger font than the basic text. The `\maketitle` command also inserts the date on which the document is compiled. Since this is inappropriate for *Pride and Prejudice* it can be overridden by specifying a date before `\maketitle`.

```
\date{1813}
```

3.3 Sectioning

Couldn't be easier: Just type `\section{text}`, replacing *text* with the specified section-heading you want to use. The section-heading text will be automatically formatted, put in the boldface type, and given a number. The numbers will be recalculated if you move text around. For instance, the section-heading above was produced using

```
\section{Document-structuring commands}
```

Variations: If you want an unnumbered section heading, do

```
\section*{Example of unnumbered heading}
```

That is, adding a `*` turns off the numbering. If you want a subsection heading, use `\subsection{}`. You turn off the numbering for subsections in the same way.

3.4 Footnotes

To insert a footnote, just type (in the text itself, where you want the footnote-maker to appear) `\footnote{text}`, where *text* is replaced by the text of the footnote. The notes will be typeset at the foot of the page in a smaller type-size, and will be automatically numbered.

3.5 Itemized lists

To get a bulleted list, just do

To have the list enumerated rather than bulleted, use `\begin{enumerate}` and `\end{enumerate}` instead. The numbering will be taken care of automatically.



```
\begin{itemize}
\item This is a first item.
\item And this is a second one.
\item And so on...
\end{itemize}
```

Figure 3: A simple bulleted list.

4 Typeface-changing commands

You will need less of these than with a standard word processor, because \TeX is smart enough to change the typeface itself, in a consistent and pleasing manner, for section-heading, footnotes and other special features of the text. But sometimes you want italics for a book-title or for emphasis, boldface for emphasis or definitions, or whatever. Here is how to do it

This <i>word</i> is in italics.	This <code>{\it word}</code> is in italics.
These words in boldface.	<code>{\bf These words}</code> in boldface.
Here's one in typewriter.	Here's one in <code>{\tt typewriter}</code> .
And some SMALL CAPS.	And some <code>{\sc Small Caps}</code> .

Figure 4: Simple typeface-changing commands.

That is, put braces ‘{’ and ‘}’ around the text the typeface of which you wish to change, and insert the readily memorable code for the type-style you want.

5 Special characters

The issue of special characters (outside of the math mode, which is discussed below) arises two instances:

- When you want a character that is not in the ASCII character set (and not on the computer keyboard).
- When you want to use in your text a character that has a special meaning to \TeX .

Take the non-ASCII characters first. You can consult one of the standard \TeX references (at the end of this document) for a full listing, but here are a few of the most common ones:

- Left-hand curly double-quote (“): type two accent graves or whatever you want to call them. Type just one of these for a single left-hand quote (‘).
- Right-hand curly double-quote (”): type two single straight quotes. Type just one for a single right-hand quote (’).
- Endash: This is properly used in place of a hyphen in giving number ranges (pp. 121–34; the war of 1914–18). Type two hyphens in succession (1914–18).

- Emdash: The proper “dash” to use in punctuating text—like this. Type three hyphens in succession.

Getting the above-mentioned symbols right is a matter of good style. But dealing correctly with the symbols that have a special meaning to \TeX is essential if your life is to compile correctly. Watch out for the following in particular:

- The dollar sign (\$): This is used as the delimiter for \TeX math mode (see below). If you want a regular dollar sign in your text (like this: \$9.99) you must prefix it with a backslash (like this: `\$9.99`).
- The percent sign (%): This is used as the comment character in \TeX : Anything following this sign on a given line of a \TeX input file is ignored by the processor. If you want one in your text, use `\%`.
- The ampersand (&): This is as the tab character in \TeX : Type `\&` if you want an ampersand to appear in the text.

6 \LaTeX math mode

As mentioned above, the ability to typeset math accurately is one of the principal attractions of \TeX and \LaTeX . To get started at this, you need to know the way into the two math modes. To get into inline math mode, type a dollar sign, `$`. Type another one to exit. By the “inline math” I mean mathematical symbols that appear as part of the regular text. To get into display math mode (for equations that are a set off on lines by themselves), type `$$` (or `\[`). To terminate this mode, close with `$$` again, or `\]` if you started with `\[`. One other variant should be mentioned: if you want an automatically numbered equation, enter math mode with `\begin{equation}` and exit with `\end{equation}`.

Once in math mode, you don’t have to worry about the spacing of what you type; \TeX will take care of that. You do have to remember some fairly simple codes for symbols that do not appear on the keyboard, and for special mathematical ways of arranging symbols. The standard references will give you all the details. For now, you can learn some of basics by example

$y = f(x)$	<code>\$y = f(x)\$</code>
$y_t = \alpha + \beta x_t + \epsilon_t$	<code>\$y_t = \alpha + \beta x_t + \epsilon_t\$</code>
$\frac{1}{3} = 0.333\dots$	<code>\$_{\frac{1}{3}} = 0.333\ldots\$</code>
$8 \times 9 = 72$	<code>\$8 \times 9 = 72\$</code>
$y = x^2$	<code>\$y = x^2\$</code>
$p \vee F \equiv p$	<code>\$p \vee F \equiv p\$</code>

Figure 5: Some basic math formulas.

As you can see, in math mode the Greek letters are obtained simply by spelling them out in English, preceded by a backslash. To get the uppercase versions, just capitalize the first letter (for instance `\Omega` gives Ω).

Subscripts and superscripts are obtained in math mode by means of `_` and `^` respectively. For complex subscripts and superscripts, enclose the argument in braces (for instance `$K^{\alpha + \beta}$` gives $K^{\alpha+\beta}$).

7 Files and filenames

At this point it may also be useful to learn about the file-naming conventions in the \TeX world. Your input file (text plus formatting commands) is typically given a `.tex` extension (for instance `my_file.tex`). The \TeX processor takes this as input and outputs a `.pdf` file (e.g. `my_file.pdf`). In the process of creating a pdf file, the processor also writes a log file and an aux file. Generally you won't be much concerned with these, but if things go wrong the log file may contain some useful information.

8 About the software

Although these pieces of software may be bundled together in a package, you need three distinct programs to constitute a fully functional \TeX system.

1. A suitable editor for preparing and modifying your input files.
2. The \TeX processor itself.
3. A program to display the compiled version of your file on screen.

The wonderful thing about all this software is that it is in the public domain. You can download it for free from any of many archive sites maintained at both academic institutions and commercial Internet servers around the world.

Further reading

The basic source on \TeX is Leslie Lamport's *\TeX , A Document Preparation System*, Second Edition, Addison-Wesley, 1994. Another useful reference is Gooseens *et al.*, *The \TeX Companion*, Addison-Wesley, 1994. For anyone who gets an urge to do some serious \TeX hacking (you can customize *any* aspect of \TeX 's operation if you're willing to learn a bit about the language), the essential reference is Donald Knuth, *The \TeX Book*, also Addison-Wesley 1994.

