A Quick Introduction to LaTeX

Ben Pfaff <pfaffben@msu.edu>

21 June 2000

Contents

1 Introduction 1
- 1.1 Suggested further reading . . . 1

2 Getting started 2
- 2.1 Installing LaTeX . . . 2
- 2.2 Running LaTeX . . . 2
- 2.3 Previewing output on-screen . . . 2

3 PostScript and printing 2

4 High level 3

5 Low level 4
- 5.1 Dashes . . . 4
- 5.2 Special characters . . . 4
- 5.3 Quotation marks . . . 4
- 5.4 Special symbols . . . 4
- 5.5 Line breaks . . . 4
- 5.6 Emphasis . . . 4
- 5.7 Fonts . . . 5
- 5.8 Comments . . . 5

6 Middle level 5
- 6.1 Title page . . . 5
- 6.2 Tables of contents . . . 5
- 6.3 Quotations . . . 5
- 6.4 Lists . . . 5
- 6.5 Verbatim text . . . 6

7 Advanced features 6
- 7.1 Macros . . . 6
- 7.2 Graphics . . . 6
- 7.3 Figures . . . 7
- 7.4 Cross-references . . . 7
- 7.5 Mathematics . . . 7
- 7.6 Tables . . . 7
- 7.7 Indexes and bibliographies . . . 8

8 Odds and ends 8
- 8.1 Emacs and LaTeX . . . 8
- 8.2 Margins . . . 8
- 8.3 Re-running LaTeX . . . 8

1 Introduction

Most of the time when I'm using a computer, I have little need for pretty documents. But from time to time, I want to produce hardcopy that isn't just nice-looking, it's actually beautiful. When there's no room for a compromise, I turn to LaTeX, which lets me produce beautiful output with just a little more effort than it takes to type my document.

LaTeX is a system designed for writing technical documents, including articles, reports, and books. It is also useful for writing less formal documents such as letters and even the occasional slide presentation. I've used it for all three purposes; this article was written using LaTeX.

Some of the features of LaTeX include strong support for mathematical formulas, figures and tables, tables of contents, indexes, graphics, and bibliographies. Due to its popularity, there are numerous extension packages available for use with LaTeX: if LaTeX doesn't support what you're trying to do directly, it is probably possible to find a package to help you do it.

1.1 Suggested further reading

This introduction barely scratches the surface of what can be done with TeX and LaTeX. If you use LaTeX much at all, then you should consult other sources of information as well. My favorites are listed in the References. I especially recommend [1], written by LaTeX's author, which serves as both a tutorial and reference.

2 Getting started

Let's step our way through installing and running LaTeX, then previewing and printing a document, before we go on to learn how to write LaTeX files.

2.1 Installing LaTeX

First, make sure that you have LaTeX installed on your computer. If you're using Debian GNU/Linux, then you can install the latest version of TeX and LaTeX with one simple command:

```
apt-get install tetex-bin tetex-doc
```

You can omit `tetex-doc` if you're low on disk space; it uses about 16 MB of disk space on top of the 34 MB or so eaten up by the basic installation.[1] If you're not using Debian GNU/Linux or a derivative, then you'll have to figure out this step on your own.

2.2 Running LaTeX

Now that we've got LaTeX installed, let's run it on the sample LaTeX file shown in Figure 1. Start by typing in the text as shown. Be careful not to substitute forward slashes (/) for backslashes (\). Save the file as `sample1.tex`. Then, at a shell prompt, type

```
latex sample1.tex
```

and hit Enter. TeX will run and, uncharacteristically for programs in a UNIX environment[2], spew lots of messages to the console. Feel free to ignore most of its output. In particular, the message `No file sample1.aux.`, if you see it, is not an error.

(On the other hand, if TeX stops in the middle of processing and prompts you with `?`, that is indeed an error. It's an indication that you mistyped something in your input file. Type `x` at the question-mark prompt, hit Enter, and compare the file you typed to what's in the figure. The error message from TeX may or may not be meaningful.)

The command that you just executed produced an output file called `sample1.dvi`, where `dvi` stands for "device-independent." You can use this file to preview your results on-screen. You can also convert it to PostScript, or print it.

[1] Remember when you could fit a full GNU/Linux install in 100 MB?

[2] But very characteristically for software written by Knuth.

```
\documentclass{article}
\begin{document}

\section{Air-Layering}

Air-layering is a variation on layering.
To reproduce an herb by air-layering,
score one branch of the plant's stem
with a knife.
Tie a plastic bag with
soil and water tightly above the score
line.

Now let's take a look at some individual
herbs in more detail.  Let's start with
sweet basil. Sweet basil is a 2-foot to
3-foot-tall annual with oval-shaped
leaves 2--3 inches long. When
its leaves are brushed, it gives off a
sweet odor.

\end{document}
```

Figure 1: Sample input file for LaTeX.

2.3 Previewing output on-screen

For an on-screen preview of the results, bring up a terminal window within X11, and type

```
xdvi sample1.dvi
```

then hit Enter. In a few moments a window should pop up showing a page that looks like Figure 2. When you run `xdvi` for the first time, or if your document uses font styles that you haven't used before, it may need to compile fonts; just be patient.

`xdvi` has plenty of features, but being a GUI they're all pretty self-evident, so I won't go into them here. Feel free to explore them on your own, of course. `xdvi` also has an informative manpage.

3 PostScript and printing

To convert `sample1.dvi` to PostScript format, use this command:

```
dvips -o sample1.ps sample1.dvi|
```

If `-o sample1.ps` is omitted from this command, then the PostScript output is, in the default `dvips`

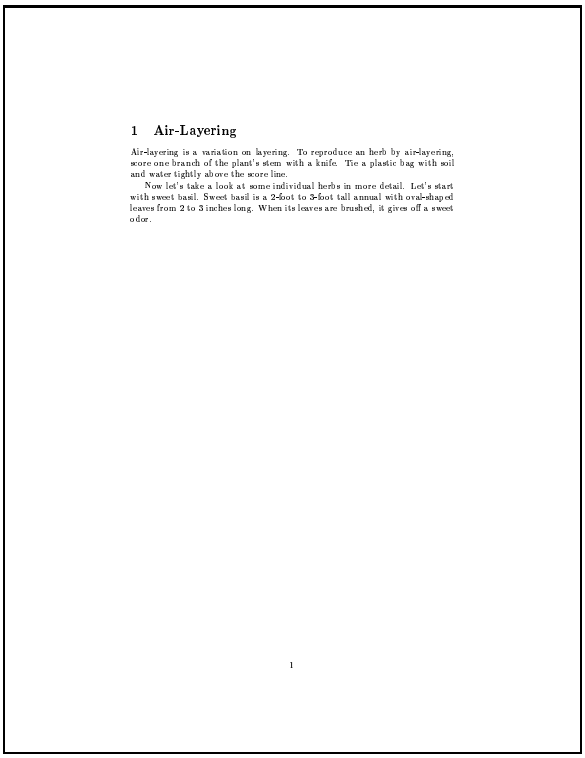

Figure 2: Formatted version of Figure 1.

setup, automatically directed to the default local printer. In conjunction with a PostScript printer or Ghostscript to translate to a given printer's language, this is the most common way that LaTeX is printed.

`dvi` filters specific to other printer languages also exist. Examples include `dvihp` for Hewlett-Packard LaserJet printers and `dvi2fax` for Group 3 fax format.

That's all you need to know in order to run TeX. Proceed to the section below to find out what all the gibberish in Figure 1 means.

4 High level

At this point you may be wondering just what the hell are all those backslashes and curly braces. The answer is that they form LaTeX commands. LaTeX commands take the following general form:

name{*argument*}

Above, *name* represents the command name, consisting either of a sequence of letters and digits or a single non-alphanumeric character. *argument* is an argument to the command.

Let's see what we can learn from the commands used in `sample1.tex`. The first command in that document is

```
\documentclass{article}
```

Every LaTeX document begins with such a command, which specifies what type of document you're writing. The possible document classes include the following, among others:

article for short articles like the one you're reading

report for technical reports and similar

book for books (duh!)

slides for presentations and slide shows

letter for letters

The differences between these document types mainly lie in fine points of presentation and sectioning. For instance, articles have titles at the top of the first page, whereas reports have separate title pages. Articles and reports are divided into sections at their top level, but books have chapters at their top level. Here, we'll discuss only articles and reports, which are the most commonly used document classes.

The `\documentclass` command can also take optional arguments, as can several other LaTeX commands. Optional arguments are enclosed in square brackets (`[]`) and precede required arguments. The following options, among others, are supported by `\documentclass`:

11pt to set a base 11-point font (the LaTeX default is 10 point)

12pt to set a base 12-point font

titlepage to request a separate title page

twocolumn to use a two-column layout, like this article

twoside to properly format the output for two-sided printing

This article is formatted in two columns for two-sided printing, so its document class is specified this way:

```
\documentclass[twocolumn,twoside]{article}
```

Notice how multiple option names are delimited by commas.

Following the `\documentclass` command, the next command in the example document is this:

```
\begin{document}
```

This command introduces what is called an **environment** in LaTeX parlance. Environments affect the text and commands that they contain. Environment commands are always paired, consisting of `\begin` and `\end` commands with the same argument. In particular, `\begin{document}` and `\end{document}` enclose a LaTeX document's entire contents. We'll see more types of environments later.

The third command in `sample1.tex` is this:

```
\section{Air-Layering}
```

This command introduces a new section in the printed document, in this case called "Air-Layering." It also enters the section name into the document's table of contents. You can also start subsections and subsubsections `\section` by using the `\subsection` and `\subsubsection` commands, respectively, in an analogous manner.

Between the `\section` command and the end of the document environment appear the body of the section. This consists of lines of text typed, for the most part, the same way you would type them in a plain-text document. Separate paragraphs by a blank line. You need not neatly format your lines, because TeX will reformat them itself.

5 Low level

We've already covered enough to let you type up basic documents with LaTeX, but there's a lot more that you can do. Let's quickly go over a few low-level details:

5.1 Dashes

Produce dashes within a word ("GNU-riffic") with a single dash character: `-`. Use two dashes for numeric ranges ("2–3 inches"): `--`. For long ("em") dashes—like this—use three dashes in a row: `---`.

5.2 Special characters

LaTeX has ten special characters that can't appear literally in a document, because LaTeX uses them for its own purposes. These characters are: `# $ % & ~ _ ^ \ { }`. You produce the first seven of these by typing a backslash in front, so that `\#` produces '#', `\$` produces '$', and so on. The last three aren't characters you want in normal text; if you think that you do, reassess your sense of taste.

To produce less than and greater than symbols, use `$<$` and `$>$`, respectively. (By themselves, `<` and `>` print as "¡" and "¿", respectively.)

5.3 Quotation marks

You can produce left quotation marks like ` with ` and right quotation marks like ' with '. If you use two of them, as `` and '', then you get double quotes, "like this."

You shouldn't use ASCII double quote marks " in your input file, because they always come out facing the same direction, "like this."

5.4 Special symbols

Use the `\ldots{}` command to produce an ellipsis, like this... To produce a © symbol, use the `\copyright{}` command. For the TeX logo, type `\TeX{}`; for the LaTeX logo, type `\LaTeX{}`.

5.5 Line breaks

Occasionally you may want to produce a line break without beginning a new paragraph; for instance, when specifying an address:

Ben Pfaff
12167 Airport Rd
DeWitt, MI 48820

For this effect, use `\\`, like so:

```
Ben Pfaff \\
12167 Airport Rd \\
DeWitt, MI 48820
```

5.6 Emphasis

Use `\emph{...}` to emphasize text: `\emph{foo}` produces *foo*.

5.7 Fonts

You can select an italic font by enclosing text within the `\textit{...}` command: `\textit{italic}` produces *italic*. Similarly, you can get **bold** output with `\textbf{...}`, *slanted* text with `\textsl{...}`, `typewriter` text using `\texttt{...}`, sans serif with `\textsf{...}`, and SMALL CAPS with `\textsc{...}`.

If you're going to use a particular font often, it might be better to define a macro for it that corresponds to the font's purpose, rather than to its name. For instance, in this article, when new terms are introduced, they are put in **bold** using a macro `\dfn{...}`. If it was ever decided that such terms should be italicized or slanted instead, only one change would need to be made. See section 7.1 on page 6 for more details.

5.8 Comments

You can include **comments**, text ignored by TEX, in your LATEX file, as notes to yourself or for any other reason. In LATEX, comments are introduced by `%` and continue through the end of a line.

6 Middle level

We've discussed the high level structuring of a LATEX document and the low-level typing of text. Now it's time to take a look at the units that fit somewhere in between.

6.1 Title page

You can have LATEX automatically generate a "title page" appropriate for the class of document you're writing by using the `\maketitle` command. This title page contains the document's title, author, and date. Before using `\maketitle`, you must declare at least the document's title and author, and optionally the date, with `\author`, `\title`, and `\date`, respectively. If `\date` is not specified, the current date is used. For example:

```
\title{A Quick Introduction to \LaTeX{}}
\author{Ben Pfaff $<$pfaffben@msu.edu$>$}
\date{21 June 2000}
\maketitle
```

`\maketitle` produces a title page appropriate to the type of document you're writing. For reports, this is a separate page; for articles, it is at the top of the article's first page.

6.2 Tables of contents

You can include a table of contents in your document by specifying `\tableofcontents` at the point you want the table of contents to appear. LATEX will automatically collect the entries for the table of contents from the sectioning commands scattered throughout your document.

Figures and tables don't appear in the table of contents. If you have lots of them, you can include separate tables using the `\listoffigures` and `\listoftables` commands, respectively.

6.3 Quotations

Sometimes you want to embed a quotation inside your text. You can use the `quote` or `quotation` environment to do this. For instance, the following quotation:

> "I'm always embarrassed when I see an index an author has made of his own work. It's a shameless exhibition—to the *trained* eye. Never index your own book."
> Kurt Vonnegut, Jr., *Cat's Cradle*

could be produced with the following LATEX:

```
\begin{quote}
``I'm always embarrassed...book.'' \\
Kurt Vonnegut, Jr., \textit{Cat's Cradle}
\end{quote}
```

Use the `quote` environment for single-paragraph quotes; use `quotation` for multi-paragraph quotes. The difference in style is that whereas `quotation` indents the first paragraph of the quoted text, `quote` does not.

6.4 Lists

You can easily produce three types of lists with LATEX: bulleted lists, numbered lists, and descriptive lists. All of these work the same way: an environment surrounds the list items, and each item begins with an `\item` command. For example:

```
\begin{enumerate}
\item A Linux-capable server.
```

```
\item One or more clients.
\item A hub.
\item A range of IP addresses.
\item A geek, for debugging purposes.
\end{enumerate}
```

produces the following list:

1. A Linux-capable server.
2. One or more clients.
3. A hub.
4. A range of IP addresses.
5. A geek, for debugging purposes.

As shown above, use the `enumerate` environment for numbered lists; `itemize` produces a bulleted list and `description` is for descriptive or definition lists. In the latter case, you may supply an argument to `\item` within square brackets to use as the item "title":

```
\item[ln] Creates links to a file.
```

for an item that appears as follows:

ln Creates links to a file.

6.5 Verbatim text

Sometimes, you'll want to include verbatim text in your documents; for example, you might want to put LaTeX source as part of your document's ouput, in the same way that this tutorial does. This is where the `verbatim` environment comes in handy. Within this environment, anything goes: you can include any character literally, including characters that normally need to be preceded by \ or that can't be included at all. The only text you can't include is the exact line `\end{verbatim}`, because that ends the environment. For example, this input:

```
\begin{verbatim}
\item[ln] Creates links to a file.
\end{verbatim}
```

produces this output:

```
\item[ln] Creates links to a file.
```

Text in a `verbatim` environment is set off in a separate paragraph. For verbatim text within a paragraph, use the `\verb/.../` command. Just put the verbatim text inside the / characters. Instead of /, you can substitute any single character of your choice that doesn't appear within the text.

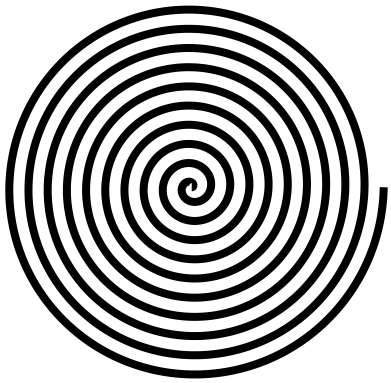

Figure 3: The spiral whose PostScript source is shown in Figure 4.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -72 -72 72 72
150 setlinewidth .02 .02 scale newpath 0 0
moveto 0 1 3600 { /t exch def t cos t mul t
sin t mul lineto } for stroke showpage
```

Figure 4: PostScript source for spiral shown in Figure 3.

7 Advanced features

Now let's take a look at some of the more advanced features of LaTeX. Of course, we're only skimming the surface, but perhaps this will serve to give you an idea of what LaTeX can do.

7.1 Macros

Use `\newcommand{\`*name*`}[`*n*`]{`*body*`}` to define a new command with *n* arguments named `\`*name* that expands to *body*. In specifying *body*, refer to the argument numbered *n* with the syntax `#`*n*. For example:

```
\newcommand{\var}[1]{\textit{#1}}
```

defines a macro, `\var`, that italicizes its argument, so that after this definition, `\var{text}` produces *text*.

7.2 Graphics

Suppose that you just designed a spiral in PostScript, like that shown in Figure 3, whose corresponding PostScript source is shown in Figure 4. In order to hypnotize your readers, you want to include this spiral in your document. How can you do it?

The answer is the `\includegraphics` command, which takes the name of a graphics file as its argument. Typically, this is an Encapsulated PostScript (EPS) file. You can use tools such as GIMP to save graphics in EPS format.

If you use `\includegraphics` then you must tell LaTeX to use the `graphics` package. Do this by inserting the command

```
\usepackage{graphics}
```

between `\documentclass` and `\begin{document}` at the top of your file.

You can resize a graphic arbitrarily by putting it inside a `\resizebox{`*width*`}{`*height*`}{...}` construction, where *width* and *height* are the desired size. You can specify *width* or *height* or both; if you want to specify only one, put `!` as the other, and the graphic will be scaled proportionally. Valid values for *width* include `3in` for a 3-inch width, `\textwidth` for the page width within the margins, and `\columnwidth` for the width of the current column.

For example, to include `x.eps`, 1 inch wide, with proportionally scaled height, write the following:

```
\resizebox{1in}{!}{\includegraphics{x.eps}}
```

7.3 Figures

More often than not, graphics go inside figures. Use LaTeX's `figure` environment to produce figures. Within `figure`, you can use `\caption{...}` to specify a caption and `\label{`*name*`}`, which must immediately follow `\caption`, to give the figure a symbolic name for later reference.

For instance, the figure containing the spiral above was produced with the following LaTeX code:

```
\begin{figure}
\centering \includegraphics{spiral.eps}
\caption{The spiral whose PostScript source
is shown in Figure \ref{spiral-source}.}
\label{spiral}
\end{figure}
```

`\centering`, which horizontally centers a figure's contents, is another new command introduced above.

7.4 Cross-references

The *name* given in a figure's `label` can be used to refer to the figure using the `\ref{`*name*`}` and `\pageref{`*name*`}` commands. The former expands to figure *name*'s number; the latter expands to its page number. For instance,

```
Figure \ref{spiral}, page \pageref{spiral}
```

might appear when printed as "Figure 3, page 6."

You can also use `\label` to label sections; put it after the sectioning command (`\chapter`, `\section`, etc.). References to sections produce the section number.

7.5 Mathematics

LaTeX has strong support for typesetting mathematics. Enclose mathematical expressions within `\(` and `\)` to embed them within a paragraph or put them between `\[` and `\]` to "display" them. For example, writing `\(t_p = C_t{(LL_c)}^{0.3}\)` produces the formula $t_p = C_t(LL_c)^{0.3}$, whereas writing `\[t_p = C_t{(LL_c)}^{0.3}\]` produces

$$t_p = C_t(LL_c)^{0.3},$$

set off from the rest of the paragraph.

It is beyond the scope of this article to detail LaTeX's support for mathematics, but it's worth noting how to do a few things. Within a formula, use `^` for superscripts and `_` for subscripts, as shown above. (If a subscript or superscript contains more than one character, surround it in braces `{}`.) Use `\frac` to produce a fraction: `\frac{4}{5}` prints as $\frac{4}{5}$. Greek letters are represented by commands corresponding to their names: `\omega` becomes ω, `\Omega` becomes Ω. Use `\sqrt` for square roots: `\sqrt{n^3}` prints as $\sqrt{n^3}$.

7.6 Tables

The `tabular` environment is used for tables. Begin the environment with a line having the form `\begin{tabular}{`*cols*`}`, where *cols* is a set of single-character column alignments, with `l` for left-justified columns, `c` for centered columns, `r` for right-justified columns, and `|` for a vertical rule. Within the table, separate columns by `&` and rows by `\\`; horizontal rules can be drawn with `\hline`.

For example, the following commands:

```
\begin{tabular}{r|c|c}
XOR & 0 & 1 \\ \hline
0 & 0 & 1 \\ \hline
1 & 1 & 0
\end{tabular}
```

result in this table:

XOR	0	1
0	0	1
1	1	0

You can put tables into running text as above, or you can put them into figures. Another option is to put them into a `table` environment, which works just like the `figure` environment except that it is intended to contain tables.

7.7 Indexes and bibliographies

Producing indexes and bibliographies with LaTeX isn't all that difficult, but there are lots of details. There's not enough room in this article to properly describe it. I recommend [1] as the definitive guide to these aspects of LaTeX.

8 Odds and ends

Here are some final points that didn't seem to fit anywhere else.

8.1 Emacs and LaTeX

GNU Emacs has a special LaTeX mode. In this mode, there are several convenient commands for editing LaTeX, including the following:

`C-c C-f` Saves the file and runs LaTeX on it in another window.

`C-c C-p` Prints the `.dvi` file generated by LaTeX.

`C-c C-v` Previews the `.dvi` file generated by LaTeX.

`C-c C-r` Runs LaTeX on only the currently selected region.

Also, typing a plain double quote " in LaTeX mode will automatically insert `` or '', as appropriate.

Normally, Emacs will automatically detect that a file is written in LaTeX and choose LaTeX mode. If it doesn't, use the command `M-x latex-mode`.

8.2 Margins

By default, TeX and LaTeX have very large margins: something like 2 inches on the left side and 1 inch on the right, with equally large top and bottom margins. You probably don't want them that big for final drafts. The following will set the margins to something more reasonable, at approximately 1 inch on each side:

```
\setlength{\textwidth}{6.5in}
\setlength{\oddsidemargin}{0pt}
\setlength{\evensidemargin}{0pt}
\setlength{\textheight}{8.5in}
\setlength{\topmargin}{0pt}
```

These are the margins used for this article.

8.3 Re-running LaTeX

Quite often it's necessary to run LaTeX more than once on a file to get the references correct. If this is true, then LaTeX will print this message:

```
LaTeX Warning: Label(s) may have changed.
Rerun to get cross-references right.
```

Just do like it says and run LaTeX again if you want to make sure that all the references refer to the right figures or pages, etc.

Alternatively, you can use the `texi2dvi` program that comes with Texinfo. This program will automatically re-run LaTeX until all references are properly resolved. For more information on `texi2dvi`, see the documentation for Texinfo.

References

[1] Leslie Lamport. *LaTeX, A Document Preparation System: User's Guide and Reference Manual.* 2nd ed. Addison-Wesley 1994. ISBN 0-201-52983-1.

[2] Torsten Martinsen. *LaTeX: The macro package for TeX.* 1996. In Debian GNU/Linux, found at `/usr/share/info/latex.info.gz`.

[3] Jon Warbrick. *Essential LaTeX ++.* 1994. In Debian GNU/Linux, found at `/usr/lib/texmf/doc/latex/general/essential.dvi.gz`.

[4] Donald E. Knuth. *The TeXbook.* Addison-Wesley 1988. ISBN 0-201-13448-9.