LaTeX Workshop

Don Brower[*]

2016-5-11

§1. TeX and LaTeX. LaTeX is a typesetting system which makes it easy to create typographically sophisticated documents. It also has a challenging, unintuitive interface. A LaTeX document starts as a file which has special markup characters (similar to HTML or Wiki Formatting), which is then processed by the LaTeX program to produce a PDF document.

Part of LaTeX is the program TeX, which came first in 1978. TeX (which sounds like *tech*, since the final *X* is the greek letter chi) provides the foundational typesetting commands. LaTeX is an add-on developed in 1986 to make using TeX easier—so much so that LaTeX is now the standard everyone uses, although one may find references to old-style TeX commands occasionally.

This course tries to provide a hands-on guide to using LaTeX. It assumes you have no experience with LaTeX, but that you do have some computer experience. More comprehensive guides and manuals are presented at the end, in §17.

To show off the abilities of this new typesetting software, Donald Knuth, the creator of TeX, made a fancy arrangement of the letters: TeX. I find these annoying, so I will use TeX and LaTeX to refer to the two programs.

§2. Where is TeX?. TeX is not a single program. It is a group of programs, and one usually installs it by installing a *TeX distribution*. The distribution to use depends on the operating system on your computer. Some common distributions are in the following table.

Operating System	Distribution
Windows	MiKTeX
Mac	MacTeX
Linux	TeX Live

In addition to a distribution, you will also need a *text editor*. A text editor is a program which lets you type and edit text documents. The most important criteria is that the program can save plain text documents. Microsoft Word could work, but you need to specifically save yours files as “Plain Text (.txt)”. Notepad is another text editor. TextEdit on Macs is almost a text editor, but one first needs to change the “Format” option in the preferences to “Plain Text”.

[*] email: `dbrower@nd.edu`

Most distributions include speciality editors which are designed to work with LaTeX. Two such are TeXshop and TeXstudio. This guide will assume you are using TeXWorks.

§3. Hello World. Type the following into your editor, save it as the file *hello.tex* and then run LaTeX on the file.

```
\documentclass{article}

\begin{document}
Hello World!
\end{document}
```

You should see a document with the phrase "Hello World!" on it. The backslash, '\', tells LaTeX that the following word is a formatting command. The first one, `\documentclass`, specifies which layout we want to use. Then the pair of `\begin{document}` and `\end{document}` wrap the words we want to be typeset. The text before the `\begin{document}` is called the *preamble.*

There are other special characters besides the backslash. They are

% $ { } _ # & ˜ > <

These symbols all have special meanings in a LaTeX file. If you should need one of these symbols in your document, each character has a corresponding command to represent it:

Special Character	Command
\	`\textbackslash`
%	`\%`
$	`\$`
{	`\{`
}	`\}`
_	`\_`
#	`\#`
&	`\&`
˜	`\textasciitilde`
>	`\textgreater`
<	`\textless`

§4. Text Formatting. We can make text bold and italic. Enter this between the begin and end pair from before.

```
This is \textbf{bold text}, and this is \textit{italic}.
They \textit{can be \textbf{nested}}.
```

It should produce something which looks like the following.

This is **bold text**, and this is *italic*. They *can be **nested***.

These commands start with a backslash, followed by an opening brace, ’{’, the text to format, and then a closing brace, ’}’. The opening and closing braces need to be matched. If there is more of one than the other, LaTeX will complain.

Try using the following character styles:

Style	Command
Normal Text	`\textnormal{...}`
Emphasis	`\emph{...}`
Roman Font	`\textrm{...}`
Sans Serif Font	`\textsf{...}`
Monospaced Font	`\texttt{...}`
Italic Shape	`\textit{...}`
Slanted Shape	`\textsl{...}`
Small Capitals	`\textsc{...}`
Uppercase	`\uppercase{...}`
Lowercase	`\lowercase{...}`
Bold Shape	`\textbf{...}`
Medium Weight	`\textmd{...}`

§5. Spaces, Justification, and Hyphenations. Most of the time you will use normal spaces in your document. LaTeX will automatically format your lines, and group them into paragraphs. It will even hyphenate words to improve the line breaks!

LaTeX will put lines which are next to each other into the same paragraph. To start a new paragraph, enter a blank line.

```
    This line will be in a paragraph.
    And this line will
    be in the same paragraph.

    This line is in a new paragraph.
    And this line is put
    in a third paragraph. Along with
    this line.
```

Produces output like the following:

> This line will be in a paragraph. And this line will be in the same paragraph.
>
> This line is in a new paragraph. And this line is put in the second paragraph. Along with this line.

The paragraphs are adjusted to fully justify on the page. LaTeX will break lines in places that gives the best justification, and will even hyphenate words. Multiple spaces will be collapsed to a single space. Comments:

- Keep LaTeX from spliting a line at a given space by replacing the space with a tilde, ’˜’. For example, Mrs.~White will not insert a line break between Mrs. and White.[1]

- You can force a line break by ending an input line with two backslashes, ‘\\’.

- If LaTeX gets the hyphenation of a word wrong, tell LaTeX by using the \hyphenation{...} command. E.g. \hyphenation{wysiwyg Zoo-lander} tells LaTeX to not hyphenate *wysiwyg*, and that *Zoolander* can be split after the *Zoo*. This command should only be needed, if at all, for technical vocabulary.

- You can force a page break command using \pagebreak

- You can put in bigger spaces using the \hspace{...} command. This line has   \hspace{1in} 1 inch of space.

§6. Quotes and Comments. Don’t use the usual quote character " in your files. Instead use two back-ticks `` and two apostrophes like so: ``Quote'' to get the output “Quote”.

There are three kinds of dashes: the hyphen, - ; the en-dash, – ; and the em-dash, —. The hyphen is used when breaking words between lines and when combining watcha-may-call-its. The en dash, made using --, is used to indicate number ranges, e.g. 5–10 business days. The em dash, made with ---, is used to indicate changes in thought—or, whatever.

[1] See http://tex.stackexchange.com/questions/15547/when-should-i-use-non-breaking-space for more discussion.

The percent sign, %, starts a comment inside your source file. LaTeX will discard the percent sign as well as any remaining text on the line. This can be useful to either leave notes to yourself in the source file, or to temporary remove text without deleting it.

```
% discuss comments
% TODO: insert cat picture here?
The percent sign, \%, starts a comment inside your source file.
LaTeX will discard the percent sign as well as any remaining
text on the line.   % comments can also end a line
This can be useful to either leave notes to yourself in the
source file, or to temporary remove text without deleting it.
```

§7. Environments. We have already seen the environment `document`. Environments enclose sections of text, and they all begin with the command `\begin{...}` and conclude with the command `\end{...}`. Some other environments are `center`, `quote`, `flushleft`, and `flushright`.

```
\begin{center}
    This line will be in its own paragraph, centered.
\end{center}
\begin{flushright}
    This paragraph will be aligned
    against the right margin.
    All the lines will be pushed as far to the right
    as possible.
    Lines will wrap themselves, or we can force \\
    a new line.
\end{flushright}
```

Produces

This line will be in its own paragraph, centered.

This paragraph will be aligned against the right margin. All the lines will
be pushed as far to the right as they can. Lines will wrap themselves, or
we can force
a new line.

§8. Lists. There are three kinds of lists in Latex: Enumerated Lists, Itemized Lists, and Description lists.

- An itemized list provides a sequence of bullet points.
- Each point may be its own paragraph
- the overall list is offset in indented

```
\begin{itemize}
    \item An itemized list provides a sequence of bullet points.
    \item Each point may be its own paragraph
    \item the overall list is offset in indented
\end{itemize}
```

Each bullet point is started with an `\item` command. The other kinds of lists are similar.

1. Enumerated lists are numbered, or lettered.
2. Lists can even be nested.
   (a) This list appears inside an item for the outer list.
   (b) Another item, just to create a bullet point.
3. The last enumerated sentence.

```
\begin{enumerate}
    \item Enumerated lists are numbered, or lettered.
    \item Lists can even be nested.
        \begin{enumerate}
            \item This list appears inside an item for the outer list.
            \item Another item, just to create a bullet point.
        \end{enumerate}
    \item The last enumerated sentence.
\end{enumerate}
```

Description Lists A description list consists of pairs of terms and descriptions.

Intention While this description list is using the item headings as on organizational tool, it is intended to use them as a way to define a bunch of terms.

Occasionally

You may want the terms to be on seperate lines than the terms being defined. Then the “\hfill newline” trick is helpful.

```
\begin{description}
    \item[Description Lists] A description list consists of pairs
        of terms and descriptions.
    \item[Intention] While this description list is using the
        item headings as on organizational tool, it is intended
        to use them as a way to define a bunch of terms.
    \item[Occasionally] \hfill \\
        You may want the terms to be on seperate lines than the
        terms being defined.
        Then the ``\textbackslash hfill newline'' trick is helpful.
\end{description}
```

§9. Math. The ability to typeset math formualae is the main reason many people choose to use LaTeX at first. A formula may be typeset *inline*, such as x is a real variable, and $f(z) = e^z$ is an analytic function. More complex formula can be typeset in *display style*:

$$H(a, r) = \sum_{n=0}^{\infty} ar^n = \frac{a}{1-r}.$$

Inline formulae are surrounded by dollar signs, $, and display formulae use `\[` and `\]`.

```
A formula may be typeset \emph{inline}, such
as $x$ is a real variable, and $f(z) = e^z$ is an analytic function.
More complex formula can be typeset in \emph{display style}:
\[
    H(a, r) = \sum_{n = 0}^\infty ar^n = \frac{a}{1-r}.
\]
```

Math formulae use carets, ˆ , for superscripts and underscores, _ , for subscripts. Other math notation use commands, such as `\frac{..}{..}` for fractions, and `\sum` for the summation symbol. You can get Greek letters using `$\alpha, \beta, \gamma, \ldots$`, to give $\alpha, \beta, \gamma, \ldots$

There are, unfortunately, too many math commands to discuss each here. If you have especially demanding math needs, the `amsmath` package has many specialized environments for mathematics.

LaTeX makes a distinction between commands entered in *math mode*, as opposed to *text mode*. Some commands only work in one or the other. For example, the `\frac` command only works in math mode.

§10. Pictures and Packages. To include a picture, we need to use the `graphicx` package. In the preamble—right after the `\documentclass{..}`, put `\usepackage{graphicx}`.

Then to include a picture—in EPS, PDF, PNG, or JPG formats—use the command `\includegraphics{...}`. For example,

```
\begin{center}
    \includegraphics[scale=.5]{hello-yes-this-is-dog}
\end{center}
```

will center the picture *hello-yes-this-is-dog* in its own text block. The `scale` option can be used to fine-tune the size of the image.

§11. Tables. There are two main table commands:

- There is the `tabular` command to arrange figures into rows and columns.
- There is the `table` command to produce a floating figure at a convenient piece of the page, say the bottom or the top.

Usually the two are used together, with a `\table` command and a `\tabular` inside of it.

The `tabular` environment takes a list of characters giving the justification for each column in the table. After that, each row of the table is listed. Each row ends with a double backslash (`\\`). The columns in a row are separated with ampersands (`&`). For example, the table at the beginning of this document looks like this:

```
\begin{center}
\begin{tabular}{ll}
    \toprule
    Operating System & Distribution \\
    \midrule
    Windows & MiKTeX \\
    Mac & MacTeX \\
    Linux & TeX Live \\
    \bottomrule
\end{tabular}
\end{center}
```

The `{ll}` says this table has two columns, and both are left-justified. The `\toprule`, `\midrule`, `\bottomrule` commands are from the booktabs package. They draw the lines in the table. You can leave them out, or if you want to use them add the line `\usepackage{booktabs}` after the `\documentclass` at the beginning of the file.

In the above example, the `tabular` environment has an enclosing `center` environment to center it on the page. Otherwise, the table will appear in the output file between the same text it appears between in the source file. The `table` environment lets the table float out of line to the top or bottom of a page. It provides an area to give a `tabular` environments as well as a caption and a internal reference number (see §13).

It is important to say that it is not necessary to use the `table` environment. The actual table formatting is done with `tabular`, and the `table` only serves to produce a floating figure.

Example of `table` usage:

```
\begin{table}[t]
  \centering
  \begin{tabular}
    ... tabular from before ...
  \end{tabular}
  \caption{This table's caption is very modest}
  \label{tbl:modest}
\end{table}
```

The `[t]` after the environment says we prefer this table to float to the top of a page. The `tabular` environment is as before. A caption is added using `\caption` and the `\label` gives an internal name to this table for cross-reference purposes (see §13).

§12. Sections and Table of Contents. Sections are started by using the `\section{}` command. The text in the braces is the section title. Everything following a section command (until the next section command) is considered part of that section. There are also commands for `\subsection` and `\subsubsection`.

The exact sectioning commands available depends on the `\documentclass` your document is using. For example, the book type includes `\chapter` commands.

A table of contents is inserted using the command `\tableofcontents`.

```
\tableofcontents
```

```
\section{Introduction}
In this report we revisit the early research done by...

\subsection{Layout of the report}
More text here...

\subsubsection{Points and Counterpoints}
And more...

\subsection{Goals}
And more...
```

You will need to run latex *twice* for this to turn out. The first time latex records the titles of the sections and the second time they will be inserted into the table of contents.

§13. References. LaTeX makes it easy to generate cross-references between sections. Most sectioning commands and figure commands can be labeled using a `\label` directive, which assigns the command an internal name. To refer to the section/figure/table one then uses a `\ref` command with the same internal name. For example, the following code will tag the section "Further Reading" with the label `sec:further-reading`.

```
\section{Further Reading}
\label{sec:further-reading}
```

To refrer to the section we then use the `\ref` command as follows:

```
See section~\ref{sec:further-reading}.
```

The tilde, `~`, inserts a non-breaking space to keep the section number from being separated from the word "section".

I usually use prefixes, like `sec:`, `eq:`, and `tbl:`, to remind me whether the label is referring to a section, subsection, table, figure, or equation. But they are not required by latex.

The `showlabels` package will display all your labels in the margin next to the item being tagged. It is very handy when writing initial drafts.

§14. Bibliography. The BibTeX program will generate bibliographies for LaTeX documents. BibTeX is a separate program which will use an auxiliary file that LaTeX generates to figure out which entries in your master "database" to include in each document's bibliography. You can manage the BibTeX database either with a program or by hand. Each citation has an identifying key, which you may assign or change yourself. To include a citation into a paper use the command `\cite{}`.

For example, if the BibTeX database contains the following entry,

```
@article {black73,
    title = {The pricing of options and corporate liabilities},
    author = {Black, Fischer and Scholes, Myron},
    journal = {The Journal of Political Economy},
    volume = {81},
    issue = {3},
    year = {1973},
    pages = {637--654},
}
```

which has the key `black73`. To cite this source in a document write

```
That option was not worth it \cite{black73}.
```

which will give

> That option was not worth it [1].

To insert the bibliography into your document, put the following code where you would like it to appear.

```
\bibliographystyle{plain}
\bibliography{research}
```

When using citations, up to four passes may be needed to convert your file to a PDF.

1. A first pass with LaTeX to generate an *aux* file listing the citations used.

2. A second pass with BibTeX to generate the bibliography.
3. A third pass with LaTeX to re-adjust the locations of the sections, tables, etc.
4. A fourth pass with LaTeX to insert the correct cross-references and table of contents.

§15. File Organization. LaTeX generates many auxiliary files when processing a document. For this reason, I usually put each document into its own folder. Then all the additional files are kept in the same place as the source files, and the extra files for different documents don't get mixed together.

For documents larger than an article, such as a book or a dissertation, I have found it helpful to make a folder for the overall project, and then put each chapter into its own sub-folder.

```
book-project/
    references.bib
    background
        only.tex
        background.tex
    why-spheres
        only.tex
        spheres.tex
    introduction.tex
    preamble.tex
    history-of-idea
        only.tex
        history.tex
    book.tex
```

In the above example directory layout, the file *preamble.tex* contains any shared definitions which I intend to use across the chapters. (e.g. I defined the symbol
`sss` in there which inserts three large stars so I can see places that I left incomplete. `\newcommand{\sss}{\ensuremath{\star\star\star}}`)

The main book is defined in the *book.tex* file. It sets up the preamble, title, abstract, table of contents, and then includes each chapter with code similar to the following

```
\chapter{Background and Prior Work}
\label{ch:background}
\input{background/background}
```

Inside each chapter folder there are two tex files. One holds the main content of the chapter, e.g. *background.tex*. This file does not include any preamble stuff nor the `\documentclass` or `\begin{document}` code. The other, *only.tex*, wraps the content file so that the chapter can be processed by itself. The *only.tex* file will look similar to the following.

```
\documentclass[12pt]{amsart}
\input{../preamble}
\begin{document}
\title{Background and prior work}
\maketitle
\input{background}
\end{document}
```

To only process a specific chapter, I run LaTeX on the file *background/only.tex*. To create the entire book I run LaTeX on the file *book.tex*.

§16. Online Collaboration. There are websites that allow one to work on LaTeX documents collaboratively, much like Google Docs. They also keep your files on their servers, so you can access them from any computer. I think both offer free and paid plans. I haven't used either, and would welcome any feedback on which is better or on other services I haven't listed.

- ShareLaTeX, `https://www.sharelatex.com/`
- Overleaf, `https://www.overleaf.com/`

§17. Further Reading.

- Andrew Roberts, "Getting to Grips with LaTeX", `http://www.andy-roberts.net/writing/latex`.

  This is a nice introduction to latex tutorial, similar to this document, but I feel it does a better job of surveying all the basic formatting commands.

- Wikibooks LaTeX reference, `http://en.wikibooks.org/wiki/LaTeX/`

  This is a **great reference** to keep handy when working on a latex document. It covers most common areas of LaTeX in great detail, and offers good extension packages to use for special formatting cases.

- Jürgen Fenn, "Managing Citations and Your Bibliography with BibTeX", `http://www.tug.org/pracjourn/2006-4/fenn/fenn.pdf`

  Jürgen Fenn article answers many basic questions about using BibTeX and pointers to handling special situations, such as proper names in a title. It only shows its age when it discusses the future of BibTeX (section 5.6), where it turns out that BibTeX standard has been accepted by libraries and other archives as an export format for citations (but not the ND Library). In part the ubiquity of BibTeX is due to the stability of its citation data format.

- Michael Downes, "Short Math Guide for LaTeX", `ftp://ftp.ams.org/ams/doc/amsmath/short-math-guide.pdf`

  This short (17 page) paper lists the extra math commands, symbols, and fonts provided by the American Mathematical Society LaTeX packages (*amssymb* and *amsmath*).

- Leslie Lamport, "LATEX: a document preparation system"

  The book by the originator of LaTeX on how to use the system. Notre Dame has copies in the library, `http://onesearch.library.nd.edu/NDU:malc_blended:ndu_aleph001563762`

§18. LaTeX at Notre Dame. In addition to the resources on the internet, Notre Dame has a mailing list for tex help, `nd-latex-users@listserv.nd.edu`. I am on this list, as are a few people who are extremely knowledgeable on LaTeX. Most questions on the list are about the Notre Dame thesis style file, but any question is welcome. Please use the list as a resource if you run into problems.

§19. Other document classes. Other layouts in addition to *article* are *report*, *letter*, *book*, and *slides*. Fancier types are the AMS article classes *amsart*, and the slide class *beamer*.

References.

[1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81:637–654, 1973.