

L^AT_EX Workshop

Oct. 2013

§1. T_EX and L^AT_EX. L^AT_EX is a typesetting system which makes it easy to create typographically sophisticated documents. It also has a challenging, un-intuitive 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 all at once to produce a PDF document.

Part of L^AT_EX is the program T_EX, which came first in 1978. T_EX provides the foundational typesetting commands, and is pronounced as “tech”, since the uppercase X is really representing the greek letter chi, χ , and has the sound of a hard “ch”. L^AT_EX is an add-on developed in 1986, and it makes using T_EX easier—so much so that it is now the standard everyone uses, although one may find references to old-style T_EX commands occasionally.

This course tries to provide a hands-on guide to using L^AT_EX. It assumes you have no experience with L^AT_EX, but that you do have some computer experience. More comprehensive guides and manuals are presented at the end, in §14.

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

§2. Where is T_EX?. TeX is not a single program. It is a group of programs, and one usually installs it by installing a *T_EX 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 specify to save it as “Plain Text (.txt)”. Notepad is another text editor. TextEdit on Macs is almost a text editor, but one needs to change the “Format” option in the preferences to “Plain Text”.

Fortunately, most distributions include speciality editors which are designed to work with LaTeX. Two such are T_EXshop and T_EXstudio. This guide will assume you are using T_EXstudio.

§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
\end{document}
```

You should see a document with the word “Hello” on it. The words which begin with a backslash, ‘\’, tell LaTeX that the word is a formatting command. The first one, `\documentclass`, specifies which layout we want to use.¹ 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 command which represents it in the document:

Special Character	Command
\	<code>\textbackslash</code>
%	<code>\%</code>
\$	<code>\\$</code>
{	<code>\{</code>
}	<code>\}</code>
-	<code>_</code>
#	<code>\#</code>
&	<code>\&</code>
~	<code>\textasciitilde</code>
>	<code>\textgreater</code>
<	<code>\textless</code>

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

¹Other layouts are XXX

```
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 playing around with the following character styles:

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

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

LaTeX will group 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.
```

- Tell LaTeX to not split 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`.²
- 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. If you use Don't use the usual quote character " in your files. Prefer to 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 the grammar books tell us.

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.

²See <http://tex.stackexchange.com/questions/15547/when-should-i-use-non-breaking-space> for more discussion.

```
% 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 an 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 separate 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 separate lines than the
    terms being defined.
    Then the ‘‘\textbackslash hfill newline’’ trick is helpful.
\end{description}

```

§9. Math. The ability to typeset math formulae is the main reason many people choose to use LaTeX at first. A formula may be either typeset *inline*, such as x is a real variable, and $f(z) = e^z$ is an analytic function. More complex formula can use *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 either typeset \emph{inline}, such
as $x$ is a real variable, and $f(z) = e^z$ is an analytic function.
More complex formula can use \emph{display style}:
\[
  H(a, r) = \sum_{n = 0}^{\infty} ar^n = \frac{a}{1-r}.
\]

```

Math formulae use carets, ^, 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\$, \$\dots\$, to give $\alpha, \beta, \gamma, \dots$.

There are too many math commands here. If you have especially demanding math needs, the `amsmath` 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{...}`.



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

§11. Tables. There are two aspects to tables: first, one needs to layout the rows and columns of data. Second, one often needs to have the table to float to a convenient portion of the page, say the bottom or the top. The `\tabular` environment handles the first concern, and the `\table` environment handles the second.

§12. Sections, Table of Contents.

§13. Bibliography.

§14. Further Reading. Andrew Roberts, “Getting to Grips with LaTeX”, <http://www.andy-roberts.net/writing/latex>

Wikibooks LaTeX reference, <http://en.wikibooks.org/wiki/LaTeX/>