

# An Introduction to L<sup>A</sup>T<sub>E</sub>X

Mark Schultz

May 31, 2017

L<sup>A</sup>T<sub>E</sub>X is a typesetting system that gives people immense control over what they do (especially compared to things like Word). That being said, it's much closer to coding than normal Word documents are, so it can have a rather steep learning curve. Things like “starting a new paragraph”, “right aligning text”, or “making quotations work correctly<sup>1</sup>” can be unfortunately difficult to do. This is a (non-exhaustive) introduction to doing some really basic stuff.

## 1 Making a L<sup>A</sup>T<sub>E</sub>X Document

LaTeX documents are separated into two distinct portions - the “Preamble”, and the “Body”. Here's a simple document (it could be saved as “exampledocument.tex”, the important part is that it's saved as a .tex file.)

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{fullpage}

\newcommand{\limn}{\lim_{n\to\infty}}
\newcommand{\paren}[1]{\left( #1 \right)}

\title{An Example Document}
\author{Mark Schultz}
\date{4/20/69}

\begin{document}
  \maketitle
  This is the Body of the document
\end{document}
```

### 1.1 Preamble

The preamble is where all of the “behind the scenes stuff” happens. Plenty of stuff in L<sup>A</sup>T<sub>E</sub>X will be written in the document, but won't be directly typeset. The preamble is “everything above the `\begin{document}`”, so in the above it includes four distinct sections:

#### 1.1.1 Setting the Document Class

The first section is the single line:

---

<sup>1</sup>See the left quotation on this.

```
\documentclass{article}
```

This tells L<sup>A</sup>T<sub>E</sub>X what kind of document you want to make. There are various options (ones for posters, for slideshows, for books, etc.), and you can even define your own (although I'm particularly bad with that). The important ones for a beginner are:

- `\documentclass{article}` - for generic problem set type things (the title only takes up a small part of the first page is a notable thing).
- `\documentclass{report}` - allows for various chapters as well. This might be good for a thesis.

You can pass various options to this as well, such as:

```
\documentclass[12pt,a4paper]{article}
```

will both modify the default fontsize of the document, as well as the paper it's formatted for (a4 is the European standard. The default is letterpaper, the American standard). A full list of options you can pass to this (and different kinds of document classes, like article and report) can be found at:

[https://en.wikibooks.org/wiki/LaTeX/Document\\_Structure#Document\\_classes](https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Document_classes)

It's important to note that LaTeX documents **MUST** start with a document class. The rest of the stuff in the preamble can be reordered however you like (although I think the way I present it is a good way to learn). This must come first though.

### 1.1.2 Importing Packages

The next section involves various package declarations. In the example document I have, it looks like:

```
\usepackage{amsmath}
\usepackage{fullpage}
```

This imports two packages, “amsmath” and “fullpage”, to be used in the document. amsmath is a set of enhancements to LaTeX that are useful to mathematicians (AMS is the American Mathematical Society). The “fullpage” package changes the margins to be smaller (wide margins are very common in journal articles, and are the default for LaTeX). The Reed Thesis template has information about packages that are more useful for Chem majors. That entire page is found here: <http://www.reed.edu/cis/help/LaTeX/thesis>

### 1.1.3 Defining Macros

The section:

```
\newcommand{\limn}{\lim_{n\rightarrow\infty}}
\newcommand{\paren}[1]{\left( #1\right)}
```

defines two new commands. One, “limn”, is shorthand for:

$$\lim_{n \rightarrow \infty} \tag{1}$$

We can normally do this by writing:

```
\begin{equation}
\lim_{n\rightarrow\infty}
\end{equation}
```

Instead, we can define a macro to do this for us. That’s what the “newcommand” does. Then, we can just write:

```
\begin{equation}
\limn
\end{equation}
```

and it’s equivalent to the first thing. Note here that we wrapped our command in an **equation** environment, which led to it being numbered, on it’s own line, and centered.

The “paren” command I defined is a little different. In LaTeX, we can write things like this:

```
\begin{equation}
f(x) = (\frac{x}{6})^2
\end{equation}
```

which displays like:

$$f(x) = \left(\frac{x}{6}\right)^2 \quad (2)$$

The parenthesis for this are clearly missized. We can fix this by writing:

```
f(x) = \left( \frac{x}{6} \right)^2
```

which renders as:

$$f(x) = \left(\frac{x}{6}\right)^2 \quad (3)$$

You do this very frequently (using what’s called “paired delimiters”) in Math. Enough that defining it’s own macro is probably worthwhile:

```
f(x) = \paren{\frac{x}{6}}^2
```

Macros also have the benefit of making it easy to modify things later. If you decide that:

$$f(x) = \left[\frac{x}{6}\right]^2 \quad (4)$$

looks better later, all you have to do is change how the macro is defined, instead of changing each time you use that notation.

You can either define macros to take no inputs (like “limn”), or any number of arguments. As an example:

```
\newcommand{\quadeq}[3]{#1 x^2 + #2 x + #3}
\newcommand{\quadform}[3]{\frac{- #2 \pm \sqrt{#2^2-4(#1)(#3)}}{2(#1)}}
```

defines a pair of macros that take information about a quadratic equation, and formats it nicely. As an example:

```
\begin{equation}
\quadeq{1}{2}{3}
\end{equation}
\begin{equation}
\quadform{1}{2}{3}
\end{equation}
```

will display as:

$$1x^2 + 2x + 3 \quad (5)$$

$$\frac{-2 \pm \sqrt{2^2 - 4(1)(3)}}{2(1)} \quad (6)$$

This obviously isn’t perfect, but shows how we can have macros with multiple arguments.

### 1.1.4 Setting Author Info

LaTeX has a default way to display the title, author, date, etc for a paper. This is done with:

```
\author{<name>}
\title{<title>}
\date{<date>}
```

Then, when you want to put the title in the actual document, you type `\maketitle`.

## 1.2 Body

The “Body” is contained entirely between the “`\begin{document}`” and “`\end{document}`” statements. This is where you write your document. The format:

```
\begin{<word>}
... stuff goes here ...
\end{<word>}
```

shows up a lot in LaTeX. It’s called an **environment**. The equation environment used before is another example of this. A few tips for basic typesetting with LaTeX:

- Paragraph breaks can be hard to understand for beginners. One thing to keep in mind is that linebreaks don’t lead to paragraph breaks. Typing:

```
This is line 1.
Is this line 2?
```

Will result in:

This is line 1. Is this line 2?

To do a paragraph break, the easiest thing to do is to leave an empty line between two sentences. As an example, using a Lorem Ipsum package, we can write;

```
\lipsum[1]

\lipsum[1]
```

displays as:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla

et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

The extra empty line between these leads to a new paragraph.

- Changing text (bolding, italicizing, etc.) can be hard to understand. The easiest way to do this (in my opinion) is as follows:

```
This text is \textbf{bold}.
```

Which displays as:

This text is **bold**.

Italicizing works similarly, but with `\textit{<word>}`. A reference on this can be found here:


[https://www.sharelatex.com/learn/Bold,\\_italics\\_and\\_underlining](https://www.sharelatex.com/learn/Bold,_italics_and_underlining)

- Including pictures can be done with the `\includegraphics{file_name.png}` command. Doing only this can result in stuff like this:



This picture is outrageously too large. We can fix this by writing



`\includegraphics[scale = .05]{orange-square.png}` (instead of `scale = 1`), to get: . If we want this as it's own thing, there's two real options - `figures`, and a `center` environment.

The center environment is:

```
\begin{center}
  \includegraphics[scale = .05]{orange-square.png}
\end{center}
```



Figures are a more “official” way of doing things, but have notorious issues (you can’t specify *exactly*) where a figure will show up, only within a page or two. Latex will try to find a natural place to put it approximately where you ask it to. This is a good resource on both figures and graphics:

[https://www.sharelatex.com/learn/Inserting\\_Images](https://www.sharelatex.com/learn/Inserting_Images)

- Numbering things and bullet points are done in two different environments. Numbering is done in `enumerate`:

```
\begin{enumerate}
  \item This is the first item.
  \item This is the second.
  \item[4] This one is numbered 4.
  \item[cow] Now this is just getting silly.
\end{enumerate}
```

1. This is the first item.

2. This is the second.

4 This one is numbered 4.

cow Now this is just getting silly.

In general, you can have LaTeX skip numbers without have to write all the subsequent number manually. You can also easily change from using arabic numerals, to roman numarals, to plenty of other systems. See here for more info:

<https://www.sharelatex.com/learn/Lists>

- Bullets are similar to numbering:

```
\begin{itemize}
  \item First bullet
  \item Second bullet
\end{itemize}
```

- First bullet
- Second bullet

The above url is also a good resource for this.

- The last thing to talk about is probably math mode. Latex has 3 different kinds of “modes” you can write text in:

1. Text mode - This is the default, and what I’ve typed all this text in so far. This is the basic wordprocessing thing for LaTeX.
2. Inline math mode. Have you ever just wanted to say “And she was such a  $f'''(x)$ ?” Probably not. But, intermingling math with text happens fairly frequently. To do this, we write:

Let  $f(x)$  be a continuous function from  $\mathbb{R}$  to  $\mathbb{R}$

to get:

Let  $f(x)$  be a continuous function from  $\mathbb{R} \rightarrow \mathbb{R}^2$ .

3. (other) math mode. This is how you write math as a block all by itself, like:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & \text{else} \end{cases} \quad (7)$$

The above example is in an equation, environment, and while the LaTeX that created it seems a little complicated:

```
\begin{equation}
  f(x) = \begin{cases} x & x > 0 \\ 0 & \text{else} \end{cases}
\end{equation}
```

It’s very similar to how you make matrices, so if you get comfortable with that this becomes easier “for free”. You may also sometimes want a few lines of math in a row. We can do this with:

```
\begin{align}
  f(x) &= (x+1)^2 \\
  &= x^2 + 2x + 1
\end{align}
```

Here,  $\backslash$  denotes a linebreak.

$$f(x) = (x + 1)^2 \quad (8)$$

$$= x^2 + 2x + 1 \quad (9)$$

This is very common throughout LaTeX. We may not want each line to be numbered for this (or for single line equations), in which case we use `\begin{align*} ... \end{align*}` or `\begin{equation*} ... \end{equation*}`. Additionally, it’s often considered aesthetically pleasing to align all of your equals signs. We can do this with an align environment:

```
\begin{align*}
  f(x) &= x + 1 \\
  f(3) &= 3 + 1 \\
  &= 4
\end{align*}
```

---

<sup>2</sup>This requires the package `amssymb`.



$$\begin{aligned}
 f(x) &= x + 1 \\
 f(3) &= 3 + 1 \\
 &= 4
 \end{aligned}$$

Here, we use an `&` on each line to tell LaTeX where to line things up at.  
 That's really all the basics of making LaTeX documents.