

Intro to LaTeX

LaTeX is a professional document typesetting system that makes beautiful, professional documents. What makes it special is that you program the content and format of the document in code. This is nice because the output is easier to organize and fine details are more customizable in code than in word processors.

Setup

- Probably easiest way to make LaTeX documents is using [Overleaf](#), an online editor that compiles your document for you.
- If you wish to compile your documents locally, you can download a LaTeX distribution for your operating system. Options are listed on [LaTeX Project](#). Most distributions will come bundled with an editor that will compile your documents.

Architecture and syntax

To start, every LaTeX document has components that look like this:

```
1 \documentclass{article}
2
3 % this area is called the preamble
4
5 \begin{document}
6 Hello world. This is a beautiful document.
7 \end{document}
```

LaTeX takes commands (or macros) in the form of `\someCommand`, and you can input arguments to those commands with `{}` and sometimes `[]`. Shown above are the `\begin{}` and `\end{}` commands. These are used for a variety of things, but here, we use them to delimit the boundaries of your document.

The area before you begin your document is called the **preamble**. This is where you declare additional formatting behavior for your document. As shown above, in the preamble, every LaTeX document must declare what class of document it is to determine basic styling. We chose report, but some other common options include: `book` or `letter`.

If you wish to do something more sophisticated than what is in vanilla LaTeX, you can load additional **packages** to change formatting. You do this with the `\usepackage{}` command, as shown in the code block below. There are many different packages included with LaTeX distributions, and the documentation for their usage is most easily found on [CTAN](#).

Note that TeX uses `%` for comments, so anything coming after (and including) a `%` on a line will be ignored during compilation. This is useful for writing notes to yourself. If you wish your `%` symbol to be compiled, you can use the command `\%`.

Similarly, LaTeX uses pairs of `$` and `$$` to mark **math mode**. A `$` marks inline math, and `$$` marks a math block.

```
1 \documentclass{article}
2
3 % this area is called the preamble
4 \usepackage{newtxtext, newtxmath} % sets the text and math fonts to times new roman
5
6 \begin{document}
7 Look, I am using math inline:  $\beta = 5$ 
8
9 Now I will have a block of math:
10
11 $$
12 E = mc^2
13 $$
14 \end{document}
```

Omitting the formatting, the output will look like

Look, I am using math inline: $\beta = 5$

Now I will have a block of math:

$$E = mc^2$$

Common commands and macros

Vanilla LaTeX accommodates all the standard structures in typesetting normal documents. How to apply them is shown below.

Headings

```
1 \chapter{This is the title of your chapter}
2 Four score and seven years ago.
3 \section{This is your numbered section title}
4 What a fine section this is.
5 \subsection*{This is a subsection without numbering}
```

Depending upon your selected document class, you may have a slightly different hierarchal outputs for your heading commands. For example, a book will use chapters, while an article won't. Additionally, if you wish to omit numbering for a heading, you can append a `*` to the command.

Font styles

```
1 \textbf{bold text}
2 \emph{italicized text}
3 \underline{underlined text}
```

bold text
italicized text
underlined

Lists

```
1 \begin{itemize}
2   \item I make a good point
3   \item Here I am making another good point
4 \end{itemize}
```

- I make a good point
- Here I am making another good point

```
1 \begin{enumerate}
2   \item first level item
3   \begin{enumerate}
4     \item second level item
5   \end{enumerate}
6 \end{enumerate}
```

1. first level item
 1. second level item

Figures

To embed figures, you have to use the `graphicx` package. Your included images should be in the same folder as the file your compiling from, or, if you wish to put them in a different folder, you can select the path to the folder.

```

1 \documentclass{article}
2 \usepackage{graphicx} % important!
3 % \graphicspath{ {figures/} } % if you have all your pictures in a folder, you can
  select the folder with this
4
5 \begin{document}
6 Shown below is a picture of a cat.
7
8 \includegraphics{cat.png}
9 \end{document}

```

The above simply inserts an image into your document. Below shows a captioned figure that is expanded to be as large as the text block.

```

1 \documentclass{article}
2 \usepackage{graphicx} % important!
3 % \graphicspath{ {figures/} } % if you have all your pictures in a folder, you can
  select the folder with this
4
5 \begin{document}
6 Shown below is a picture of a cat.
7
8 \begin{figure}[h] % `h` declares that you want your picture approximately where you
  inserted it
9 \centering % centers the image within this figure
10 \includegraphics[width=\textwidth]{cat.png} % width= determines the size of your
  image. you could also put width=2in or other measurements
11 \caption{A tabby cat}
12 % \label{fig:cat} % this isn't necessary, but can be helpful if you wish to
  reference this figure later
13 \end{figure}
14
15 \end{document}

```

The `[h]` parameter tells the compiler you want your picture roughly where you inserted it. This is important because sometimes figures can be positioned awkwardly on a page, so the compiler will choose a good place for us. If you wish to put it exactly where you inserted it, you can use `[H]` instead. Alternatively, you can use `[t]` or `[b]` for top or bottom, respectively. This can be a little finicky if you have a lot of images you are trying to place on one page.

Tables

Tables can be a bit tricky. First, tables are created with `\begin{tabular}` `\end{tabular}` commands, as shown below

```

1 \begin{tabular}{l | c r }
2 \hline
3 \textbf{Name} & \textbf{ID number} & \textbf{Color} \\
4 \hline
5 Arya & 072 & Cyan \\
6 Jane & 958 & Orange \\
7 John & 123 & Blue \\
8 \hline
9 \end{tabular}

```

Name	ID number	Color
Arya	072	Cyan
Jane	958	Orange
John	123	Blue

The `{ l | c r }` shows that you have three columns, the left with `l` left aligned text, the center with `c` entered text, and the right with `r` ight aligned text. The `|` places a vertical line between the columns. The `&` delimits each cell in a row, the `\\` shows where the row ends, and the `\hline` adds a horizontal line between rows..

Bibliography and citations

Bibliographies are one of TeX's strongest points. LaTeX links to a file where you save the known information for your bibliography. You can adjust the formatting of the bibliography the fly using the `\bibliographystyle{}` command.

```

1 \bibliographystyle{IEEEtran}
2 \bibliography{references}

```

In the code block above, we set the bibliography style to IEEE, and link to a `references.bib` file that contains out bibliography information. The contents of a `.bib` file are shown below.

```

1 @article{customBibLabel,
2   title={The book title},
3   author={Wayne, R},
4   journal={The Journal},
5   volume={72},
6   pages={500--505},
7   year={2012}
8 }

```

To edit your `.bib` file, you can use a bibliography manager such as BibDesk (recommended), or you can edit it by hand.

In the body of your document, you can cite certain entries in your bibliography with the `\cite{label entry}` command. It will automatically make the citation in the selected style. An example is shown below.

```
1 | The cells underwent mitosis \cite{customBibLabel}.
```

```
| The cells underwent mitosis [1].
```

Notice that you can name the first field, the label, in the `.bib` file whatever you want, but this label is what you refer to in your `\cite{}` command.

Footnotes

Footnotes can be placed with the following command `\footnote{content of footnote}`. It will automatically be numbered and placed in the footer.

And more

A more comprehensive review is provided by [Overleaf](#).

File types

- `.tex` files contain your LaTeX code you can compile from.
- `.bib` files contain your bibliography data.
- `.cls` or *class* files contain LaTeX code for a document class (book, report, article, etc.) formatting that you can easily load in. Inside these files is mostly code that you would find in the preamble, but we place it in their own separate files so that they are portable and can be used as templates.
- `.bst` files contain formatting code for bibliography notation styles. E.g. IEEE vs APA vs MLA.