Basics of LaTeX

Patricia Ellis do Amaral

Abstract—This tutorial will explain how to create a simple LaTeX document and implement assorted useful features. The expected audience is students in STEM fields, particularly those comfortable using programming languages, who have never used LaTeX before.

Index Terms—CMPE185, LaTeX Tutorial, IEEEtran, journal, LaTeX, paper, template.

HELLO and welcome to this, an introductory LaTeX tutorial that will walk you through creating a LaTeX file and implementing some of its particularly useful features.

LaTeX is a markup language with which you can create beautifully formatted documents. The learning curve is a bit steep, but once you've gotten a feel for the syntax, you can breeze right past all the fiddly details of positioning and indentation that can be so tricky in word processors.

The most important thing to keep in mind is that LaTeX is built to let you write normally, and then smoothly integrate formatting commands into that writing. Nearly every command in LaTeX is therefore preceded by a \ to let LaTeX know that the next word should be treated as a special command. So as you read this tutorial, remember that the \ is the cornerstone upon which LaTeX builds its kingdom.

As you read through this tutorial, you'll find little boxed numbers like this: [0]. These link to the bibliography, where you'll find sources with further information on all of the commands described here. (Whatever it takes you to will be RIGHT at the top of the page, unfortunately.) Now go forth, and make beautiful documents!

Yes, you must pronounce it LaTeX every time. Good luck doing that with your mouth.[1]

1 Creating a .tex file

STEP one is going to be getting an Overleaf account. You could also go to the official LaTeX website and download a bunch of software, but don't do that. Overleaf is the best LaTeX IDE[2] out there. You can go to https://www.overleaf.com, get a free account with nothing but an email address, and have access to everything you need. Love yourself and use Overleaf.

Once you've gotten an account, hit the green "New Project" button at the top of the menu on the left of the screen, and choose "Blank Project". Overleaf will automatically create a preamble and a header for your new project.

2 Preamble

THE preamble is where you set up your document's structure. None of this will be seen by the reader. There are two important commands here:

1) The `\documentclass` command tells LaTeX what kind of document this is going to be. Overleaf defaults to `\documentclass[article]`, which is the most commonly used template. Other document classes include `report`, `book`, and `letter`,

1. There are actually several ways to pronounce it. Officially it's "lah-tek", though lots of people say "lay-tek" and some just say "lay-teks". The reason for the hard "k" sound at the end, rather than what you'd expect from an x, is that the X in LaTeX is in fact meant to be a Greek chi (χ), which has that hard sound.

2. An Integrated Development Environment, or IDE, is a program that handles all the background work of coding, such as compiling your code into a program file. LaTeX is a markup language, not program code, but it still needs to be compiled into a formatted PDF file. Overleaf will handle that for you.

which come with their own formatting options.[5]

2) The `\usepackage` command (or rather commands, since you can have lots of them) works similarly to #include or #import commands in C++ or Python. Features like the ability to embed images or format math equations have been bundled into packages. A package must be included in your document before you use any of the commands bundled into it. Convention dictates that all `\usepackage` commands happen at the top of the document, for neatness and so you don't lose track of what you have and haven't included.
   Overleaf will default to `\usepackage[utf-8]{inputenc}`. This simply means that you'll be able to use standard UTF-8 characters in your LaTeX document.
   Wikibooks has a convenient list of commonly used packages [11].

The preamble also commonly contains the title section.

3 Title Section

THE title section contains not just the title of your paper, but also the author and the date. Other templates can include more options, such as an abstract for a research paper, but the default options are `\title{}`, `\author{}`, `\date{}`, and `\thanks{}`, which is for adding footnotes to the title section. Put the title inside the curly braces of the `\title` command, the author inside the `\author` curly braces, and so on. Any one of these can be left blank. Any of these except the `\title{}` can be left out entirely. [14]

Writing the date as `\date{\today}` will tell LaTeX to show today's date when it compiles the document.

At the end of the title section, you must include the command `\maketitle`. This will essentially compile the title section; without it, the title section will not print. However, this command should go after the `\begin{document}` tag (described in the next section), not in the preamble.

4 Environments

ENVIRONMENT is just a fancy word for an organized chunk of text. Sometimes the text is organized into a list, sometimes it's organized into a table, sometimes it's organized around an image; for each of these, LaTeX has a specific formatting style, and the environment tags will tell LaTeX which one to use. These tags are simply `\begin` and `\end`.

Make sure that every time you open an environment, you also close it, and vice versa. If you have one without the other, LaTeX won't know *what* chunk of text to organize this way, and it'll freak out and give you really weird errors, often in a completely different part of your document. If you've ever written in C++ or Java, you probably remember how awful it is to have a loose {curly brace} floating around in your code – this is the same idea.[3] Keep track of your environment tags.

The most important environment of all is the document environment, aka everything you're writing. Immediately after your preamble, add the line "`\begin{document}`". Then at the bottom of your LaTeX file, add as your very last line "`\end{document}`". **You must include these two lines in any LaTeX paper.** If you don't, LaTeX won't know where your document begins and ends, and it won't compile right. (Overleaf will include these lines by default when creating a new project.)

4.1 Lists

Lists are a super useful environment. To make a numbered list, write:

```
\begin{enumerate}
    \item This is the first item.
    \item And look! The second one!
\end{enumerate}
```

3. If you haven't, just think of parentheses. (Wouldn't it be frustrating, and maybe even a bit confusing, if I never closed off this parenthetical statement? You'd have to guess where it ends from context clues. And let me tell you, humans are WAY better at guessing from context clues than even the best IDEs.

This will look like this:

1) This is the first item.
2) And look! The second one!

For a bulleted list, simply use {itemize} instead of {enumerate}. Lists can be nested, simply by opening a new environment within the list:

```
\begin{itemize}
    \item First layer
    \begin{itemize}
        \item Nested layer!
    \end{itemize}
    \item Back to the first layer
\end{itemize}
```

That markup code will become this list:

- First layer
  - Nested layer!
- Back to the first layer

Every nested list needs its own \begin and \end tags.

5 Sections

Organizing your document into sections tends to make it more readable and is often required by STEM-oriented style guidelines. LaTeX makes this organization super easy. To start a new section, all you have to do is write \section{} on a new line. Inside the brackets, write the title of your section. For instance, this section's title was written as \section{Sections}.

LaTeX will automatically number sections. To convince it not to, include an asterisk just before the opening curly brace, like so: \section*{Sections}.

5.1 Subsections

Subsections are just written as \{subsection {title of your subsection}. This one was written as \subsection{Subsections}.

5.1.1 Subsubsections

You can even have a subsubsection, though that's as deep as you can go. (If you do need to go deeper, you can always just DIY it with some text formatting.)

6 Body Text

This is the easiest part. Once you have the framing of \begin{document} and \end{document} at the head and foot of your LaTeX file, anything that you write without tagging it specially is going to be body text.

6.1 Text Formatting

For **bold text,** use \textbf{bold text}, with whatever you want bolded written normally inside the brackets. For *italics,* use \textit{italics}.

6.2 Alignment

By default, the body of the document will be fully justified, but you can center your text on the page, or align it to the right or left, with tags such as \center, \flushleft, and \flushright. [13]

7 Reserved Characters

Reserved characters in LaTeX are like reserved words in a programming language: they serve a specific function or encode a particular command, so you can't use them normally in your writing. That doesn't mean you can't use them at all – you just need to "escape" them, typically by adding \ in front of them as an "escape character". This will tell LaTeX to read them simply as characters, not markup commands.

A few useful reserved characters are:

\ The backslash is the most basic and most important LaTeX symbol. Every single command must start with a backslash, from the formatting tags like \textit{italics} to the environment definers like \begin{enumerate}. If you want to actually include a \ in your writing, you have to encode it as \textbackslash.

% The percent symbol denotes a comment, so LaTeX will ignore anything you write after a % symbol.[4]

4. Comments are good for writing notes to yourself that the reader isn't meant to see.

~ The tilde keeps words together on a single line. For instance, if you want to keep your name together on one line, but you're running up against the edge of the page, you could write it as `your~name` and LaTeX will push the whole name onto the next line rather than split it up. This is rarely important, but it sometimes makes things look much nicer, and the whole purpose of LaTeX is pretty formatting.
The escape for ~ is `\~{}`. Any character you put inside the curly braces will be accented with a tilde, like so: ñ. If you don't include braces at all, then whatever character comes directly after the tilde will be accented.

\\ A double backslash will force a newline – but it's not quite starting a new paragraph. A new body paragraph will be indented, but a double backslash won't indent. It's like the difference between hitting "Enter" in a word processor and hitting "Shift + Enter": one creates a new paragraph, another continues the current paragraph on the next line.
The escape for \\ is just writing `\textbackslash` twice.

Note: Some reserve characters, when you escape them as described above, will appear without a space after them, even if you put one there. For instance, if you write `\textbackslash`, the \will bump up against the next word, like it did just there. To avoid this, add another backslash after it, like so: `\textbackslash\` and then continue writing normally. What you're essentially doing is "escaping" the space character, forcing LaTeX to render a space there whether it wants to or not.

8 Mathematical Formulas

BEAUTIFULLY formatted math is arguably the biggest draw of using LaTeX. To format something as a mathematical equation, just put it within two dollar signs: `$ x^2 + y $` becomes $x^2 + y$.

If you want the equation to go on a separate line, you use two dollar signs:

```
$$ x^2 + y $$
```

Which will produce this:

$$x^2 + y$$

8.1 Superscripts and subscripts

A lot of the more basic math formatting commands are, if not intuitive, at least pretty straightforward. For instance, x^2 is just written as `$x^2$`. Less obviously but no less simply, y_3 is written as `$y_3$`.

If you need to put more than a single character in the superscript, you'll need curly braces. So x^{9+z} is written as `$x^{9+z}$`. Similarly, y_{z-4} is written as `$y_{z-4}$`.

This convention of a command followed by a pair of curly braces, inside of which you put everything that command is meant to affect, is a pattern that you'll find all over the place in LaTeX.

8.2 Fractions

You can make really lovely and cool fractions with LaTeX, but to do this you need to add this line to your preamble:
`\usepackage{amsmath}`

This lets you use the \frac command, among so many others.[6] All you write is `$\frac{a+b}{x}$`, and LaTeX will turn it into $\frac{a+b}{x}$ or, if you put it on a separate line as described at the start of this section,

$$\frac{a+b}{x}$$

8.3 Fun Math Symbols

LaTeX has encoding for just about any mathematical symbol you could want to use. Set theory? `$A \cup B$` and `$A\cap B$` will get

you $A \cup B$ and $A \cap B$. Calculus? `\int` will get you $\int$ and adding superscripts and subscripts can get you the whole kit and caboodle:

```
$$ \int^{x=3}_{x=7} x^4 \ dx $$
```

$$\Downarrow$$

$$\int_{x=7}^{x=3} x^4 \ dx$$

Overleaf has lots of convenient lists of mathematical symbols and their markup encodings, which you can peruse at leisure or google at will. [9] [10]

9 Tables

To make a table, invoke the tabular environment with `\begin {tabular}{c c c}` and `\end{tabular}`, then write the code for your table between those two commands.

The `{c c c}` is a parameter you pass to the tabular environment indicating that your table will have three columns, and their content will be centered. Using an `r` or an `l` will align the text to the right or left side of the column. You can also add column separators with the | symbol: `{c|c|c}`.

To add content, write out the contents of each row, separated by & symbols as column delimiters. A \\ will mark the end of each row. This is where you can add row separators: `\hline` between rows will draw a horizontal line.

So to encode this table:

one	thirty-three	five
twenty	four	sixty-seven

You would write:

```
\begin{center}
\begin{tabular}{r|l|c}
    \hline
    one & thirty-three & five \\
    twenty & four & sixty-seven \\
    \hline
\end{tabular}
\end{center}
```

(The `{center}` environment just centers the table on the page, as you would expect.)

If you keep poking around, you'll find a lot of built-in features beyond these basics with which you can create highly customizable tables.[12]

10 Figures

Including figures in a LaTeX document requires a graphics package, so you'll have to add this line to your preamble:

`\usepackage{graphicx}`

(The x is not a typo: the *graphicx* package is an extension of the older *graphics* package.)

To include an image, you'll have to open a figure environment with `\begin{figure}`. LaTeX will place the figure wherever it fits best on the page, so if you want to specify that it belongs at the particular point at which you're writing the figure environment, you can add the parameter [h] to the environment tag: `\begin{figure}[h]`. Be sure to close the environment with `\end{figure}` to avoid weird errors.

Actually inserting the image is done with the command `\includegraphics{}`, which you'll put inside the figure environment. Inside the curly braces, put the full file path to the image. If you're using Overleaf, you'll need to upload the image to the project folder,[5] and then all you have to put in the curly braces is the image file name. (For simplicity's sake, make the file name just one word, with no spaces.) For instance, I'm going to insert a picture of a cute fish I found on the internet (and I'm using Overleaf, because I love myself), so I'll write `\includegraphics{fishie.png}`.

Figures can be captioned and labeled. The caption is what your readers will see at the base of the figure, and should be used to explain what the image means. The label is for your own use: when referring to this image in your text, you can use the label to link to the image.

5. To upload an image, open the project and look at the top of the blue sidebar on the left-hand side of the page. Just above the list of files, there are five little white icons. The middle one, with an arrow sticking up out of it, is the upload button.

You can use parameters added to the `\includegraphics` command to control the way the image appears on the page. For instance, you can specify width, height, or both to control the image's size. You can be specific, e.g. `\includegraphics[height=5cm]{image.jpg}`, or you can set the image to span the width of the page with `[width=\textwidth]`.

So here's the code for my fish pic:

```
\begin{figure}[h]
\includegraphics[width=\columnwidth]
    {fishie.png}
\caption{A cute little Regal
    Angelfish!}
\label{fig:cute-fish}
\end{figure}
```

We start by opening the figure environment, specifying [h] to tie the figure to this particular point in the text. The `\includegraphics` command gets two parameters: the figure size, which here is the column width (on a paper with no columns, I might use `\textwidth`), and the name of the image. This image is stored in my Overleaf folder as fishie.png, but if I were writing this LaTeX document on my computer rather than in Overleaf, I'd include the full path to the image file.

The caption just goes directly in the curly braces. The label is arbitrary, which is to say you can label it anything you like – the reader won't see it anyway. That said, it's usually a good idea to use something descriptive, so you won't lose track of which figure is which. Don't use numbers – using numbers just means that if, later on, you add another figure above this one in the document, you'll have to go through and re-number all your figures. Not worth the hassle.

I'm putting the actual image code right here:

You'll notice it's not here. That's because even though I started the figure environment with a `[h]` to *insist* on putting the picture *right here*, there's simply no way for LaTeX to make it fit. LaTeX is above all about aesthetic, and when push comes to shove, it will pick making your document pretty over obeying your orders every time.

Fig. 1. A cute little Regal Angelfish!

Alright, here's the picture now. Note the caption and the "Fig. 1." numbering, which LaTeX adds automatically.

To reference this figure in the body of your document, write `\ref{fig:cute-fish}`. This will create a numbered link, 1, which you can click on to go to the figure. Note that if you don't include a `\caption{}` command in your figure environment, even if you leave the space between the curly braces empty, then `\ref` won't know what to reference to, and might default to the section or page containing the figure.

There are a lot of options for customizing figures in LaTeX beyond these basics that you should explore! [7] [8]

11 How To: References

Conveniently, LaTeX has built in functionality for a bibliography. Adding the environment tag `\begin{thebibliography}{9}` will create a new unnumbered section titled "References" where you can list your sources.

The 9 is a parameter which indicates how many characters you need for your references, but the number itself doesn't matter, only how many digits are in it. Basically, if you're going to have fewer than ten references, you can write `{9}` or some other single digit; any higher, and you should use a two-digit number for your parameter (or three-digit if you just really love having sources), but that number could be 28 or 70 just as easily as 99.

Each source in your bibliography should start with `\bibitem{item-name}`, where `item-name` is the variable name you'll use to cite this source in your text. After that, write your bibliography normally, adding formatting wherever relevant, such as italicizing book titles.

An example:

```
\begin{thebibliography}{11}

\bibitem{exampleA} Author. (Year).
\textit{Book Title.} Location:
    Publisher.

\bibitem{exampleB} Web page title.
    (Year). Retrieved from url.com.

\end{thebibliography}
```

⇓

[1] Author. (Year). *Book Title.* Location: Publisher.

[2] Web page title. (Year). Retrieved from url.com

When citing a source in the body of your paper, write `\cite{exampleA}` to get a [1] in your text that links to the bibliography, such as the number at the end of this sentence which you can follow to a source on LaTeX bibliographies. [4]

Conclusion

Now that you're equipped with the basics of LaTeX, go forth and make beautiful documents! For more in-depth information on any of what's described above, check out Overleaf's LaTeX documentation [2] or the LaTeX wikibook [3].

Overleaf will direct you to a great many packages that you can add to your document for all sorts of cool features, but there are a great many more that you can find online, made unofficially, since LaTeX is open-source. There are also plenty of custom document classes, which encode various interesting layouts. The one I've been using for this tutorial, for instance, is the journal subclass of the IEEEtran class, designed by the IEEE. [1]

Acknowledgements

I would like to thank the developers of Overleaf for making such a lovely and usable IDE for LaTeX, and for their numerous high-quality tutorials, which are much better than this one and which I used extensively.

A warm thanks also goes to the generous editors of wikibooks who created an entire online "book" detailing all you can do with LaTeX. It's genuinely very cool what people will do, for free, for absolute strangers that they'll never meet.

I also couldn't have made this tutorial without the erratic but thorough answers given on dozens of StackExchange questions. I'm not sure what motivates people to hang out on StackExchange and answer programming questions, but I'm very grateful that they do it.

I am grateful as well to the Purdue University website on bibliographies, for their thorough documentation of every kind of citation under the sun.

Deeply grudging thanks are extended to the creators of LaTeX. I'll grant that their markup language is pretty rad, but the name is deeply terrible and the logo is worse. I don't expect I shall ever forgive them for it.

References

[1] Shell, M. (2002). How to Use the IEEEtran LaTeX Class. *Journal of LaTeX Class Files, vol. 1.* Retrieved from https://ras.papercept.net/conferences/support/files/IEEEtran_HOWTO.pdf.

[2] Documentation. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Main_Page.

[3] LaTeX. (last updated 1/20). Wikibooks/LaTeX. Retrieved from https://en.wikibooks.org/wiki/LaTeX.

[4] Bibliography Management. (last updated 10/19). Retrieved from https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management on 1/30/2020.

[5] Document Structure. (last updated 10/19). Wikibooks/LaTeX. Retrieved from https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Document_classes on 1/30/2020.

[6] Fractions and Binomials. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Fractions_and_Binomials on 1/30/2020.

[7] Floats, Figures and Captions. (last updated 9/19). Wikibooks/LaTeX. Retrieved from https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions 1/30/2020.

[8] Inserting Images. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Inserting_Images on 1/30/2020.

[9] Integrals, sums and limits. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Integrals,_sums_and_limits on 1/30/2020.
[10] List of Greek letters and math symbols. (2020. Overleaf. Retrieved from https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols on 1/30/2020.
[11] Package Reference. (last updated 8/19). Wikibooks/LaTeX. Retrieved from https://en.wikibooks.org/wiki/LaTeX/Package_Reference on 1/30/2020.
[12] Tables. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Tables 1/30/20.
[13] Text alignment. (2020). Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Text_alignment on 1/30/2020.
[14] Title Creation. (last updated 11/18). Wikibooks/LaTeX. Retrieved from https://en.wikibooks.org/wiki/LaTeX/Title_Creation on 1/30/2020.