# Latex Tutorial

My Name

**Abstract**—This paper is designed to introduce basic LATEX features to people. The paper goes over simple elements such as creating the .tex file, to learning how to use sections and subsections. The paper will also go over how to use multiple mathematical symbols and how to create tables or images.

**Index Terms**—CMPE185, LATEX Tutorial, IEEEtran, journal, LATEX, paper, template.

✦

## 1 Introduction

THIS tutorial will go over the basics of LATEX and should give beginners enough information about the system to create their own papers. It will go over the very basics of the system, such as creating a .tex file and how the LATEX syntax works. This tutorial will also cover several different features such as creating tables or using different packages to display mathematical equations or symbols.

LATEX gives users extreme control over the formatting of their documents. The number of features that LATEX provides makes writing academic papers a lot simpler than using a regular word processing program such as google docs or microsoft word. For instance, creating math papers will be much easier in LATEX as once you learn the syntax, typing them down will be much easier than searching for the symbols on google docs.

## 2 Basics: Creating the .tex File

### 2.1 Environments

The first thing in LATEX is understanding the purpose of environments. In order to create an environment you start by typing

```
\begin{}
```

You then end and environment by typing.

```
\end{}
```

What you put inside the curly brackets is what type of environment you will be using. You can create custom environments or use ones from packages depending on your needs for the paper. However, in order to start anything in LATEX you need to use the ”document” tag. So you would then start your .tex file by typing

```
\begin{document}
\end{document}
```

Once that is done, everything that you want to appear in the final PDF of your paper will be typed in between these two tags. Environments will be used a lot throughout this tutorial, so there will be plenty of examples that will show its usage, other than simply starting the document.

## 2.2 Reserved Characters

There are special characters within LaTeX that are reserved for using the system's special features. For instance the \ character is used to start commands or type out special characters within LaTeX . The character $\sim$ is used for creating a space between characters or words. For example if we were to write down the symbols x and y we would get.

$$xy$$

In order to leave a space between them, you can not simply use a whitespace character to separate the two characters. This is where the $\sim$ comes in.

$$x~y$$

To see what this looks like, here is a simple comparison. This is the code snippet that created the two lines of math above.

```
\[xy\]
\[x~y\]
```

Another two crucial special characters are \\ and %. \\ is used to create a line break. It is technically a macro for the command

```
\newline
```

Where as the % character is used to create comments in the LaTeX code. This is especially helpful if you are working on a paper with several collaborators, this lets you leave notes for your partners on any section of the paper without flooding the output with random text. This also lets you comment out sections of the paper that you might want to remove without deleting all the code.

These are not the only special characters in LaTeX but more will be covered throughout the tutorial. If you wanted to use these characters in text, it is rather simple. For each special character, the only thing needed to be added is a backslash behind the reserved character. For instance,

```
 \%
```

displays the % character as such, within text. The only exception to this rule is the backslash. In order to display a backslash, one would need to go into math mode (Math mode will be explained later on) as such

```
 $\backslash$
```

## 2.3 Preamble

The preamble in a .tex file is everything that happens before `\begin{document}`.

### 2.3.1 Document Class

At the very top of the document, before you type anything else you have to specify what document class you want your paper to be in. This can be done by typing the following line at the top.

```
\documentclass[options]{class}
```

Within the `class` tag, this is where you specify the layout your paper will come out with. Some of these tags include `article`, `report`, or in the case of this tutorial `IEEEtran`. The options section, this is where you specify things such as font size, size of the paper, or the layout of your text. For example, a document with 12pt font size and the IEEE format will look like this

```
 \documentclass[12pt]{IEEEtran}
```

### 2.3.2 Packages

Packages allow for users to add more functionality or features to the system. For those familiar with programming, it is essentially the same thing as importing packages or libraries to your code. This is done by typing

```
\usepackage{package_name}.
```

Some examples of packages are `amsmath` which helps users create matrices within the paper or the `graphicx` package which gives users more options when adding images. The usage of packages differs depending on the purpose of the paper, but there are plenty of resources online to help look for the right packages for you.

#### 2.3.3 Title and Heading

When creating the title for you paper, LATEX has special commands that assist in the process. The format of your title would depend on your specified document class. In order to set the title, author and date you simply have to use the commands `title`, `author` and `date` respectively. As an example, when creating this tutorial I used these three lines.

```
\title{Latex Tutorial}
\author{My Name}
\date{10/15/2021}
```

There are other options when creating a title, but these three are the core commands that you need. Once these are completed the only thing need to be done is to write the commands

```
\maketitle
```

and you title will be rendered at the beginning of the paper.

## 3 Sections

When creating a paper, you are most likely going to need to have sections to help divide information. In order to create a section you use the command.

```
\section{section_name}
```

You can also create subsections and additional subsections for said subsections. as such

### 3.1 This is a subsection

Code for this looks like this.

```
\subsection{This is a subsection}
```

#### 3.1.1 Subsubsection

The code for this looks like.

```
\subsubsection{Subsubsection}
```

These sections and subsections are all automatically numbered by LATEXḞor instance if I were to create another subsection, its number would be incremented automatically.

### 3.2 Another one

If you want to remove the automatic numbers for any reason you simply have to add an asterisk before the curly brackets as such

```
\section*{name}
\subsection*{name}
\subsubsection*{name}
```

## 4 Body Text

When creating regular text in LATEX it is as simply as typing in a word document. However, there are several commands to keep in mind to edit your text. For instance if you wanted to bold or italicize your text, you can use the commands `textbf` or `textit` respectively. If you want to have your text in courier font to show a code snippet you can use the command `texttt` . You can also align text by creating an environment and using tags such as `center` or `flushleft`.

When creating a new paragraph, you can use the double backslash \\ or use the command `par`. You can also edit the length of your indentations for a paragraph by typing

```
\setlength{\parindent}{10ex}
```

The 10ex tag in the command refers to the number of ”x” characters your paragraph will be indented by.

## 5 Tables

When creating a table, you need two different environments, the `table` and `tabular`. In order to create a table you need to enclose the contents of the table with these two tags. An example of a small table in LATEX might look like this

```
\begin{table}[]
\begin{tabular}{ll}
A & B \\
C & D
\end{tabular}
\end{table}
```

This code generates a table that looks like this

| | |
|---|---|
| A | B |
| C | D |

The two `l`'s next to the `tabular` tag indicate how many columns there are and how they are aligned. You can also use the characters `c` and `r` to align the columns either center or right. The number of characters will signify the number of columns there are.

You can also see the \\ at the end of the first row of the table. This signifies that you are the end of the row and want to go down to the next one. In order to create borders around your table, the command `hline` can be used to create a horizontal line. For vertical lines, the `|` character can be added between the `tabular` tags to signify which columns gets border. For example, if we wanted to add borders to the entire table we would change up our original code to look like this instead.

```
\begin{table}[]
\begin{tabular}{|l|l|}
\hline
A & B \\ \hline
C & D \\ \hline
\end{tabular}
\end{table}
```

This will then generate a table that looks like

| | |
|---|---|
| A | B |
| C | D |

## 6 Figures

In order to add images, you first need to include the package `graphicx` by using the `usepackage` command. Once it has been added, you must create a `figure` environment and use the command `includegraphics` as such

```
\begin{figure}[tags]
  \includegraphics[tags]{image_filename}
  \caption{caption}
  \label{label}
\end{figure}
```

Make sure that the image file is in the same directory as the .tex file. As an example, let us take this code snippet. Within these environments, you can also use the commands `caption`, to add a caption to the image and `label` in order to reference the image in text later on using the `ref` command.

```
\begin{figure}[!h]
  \includegraphics[scale=0.2]{slug.pdf}
  \caption{Our pride and joy.}
  \label{fig:slugman}
\end{figure}
```

This generates the image.

Fig. 1. Our pride and joy.

`image.png`

You can see that in the code snippet that both the `figure` and `includegraphics` environments have tags that can be added.

### 6.1 Figure Environment

These are the tags available

`h`: place the image here, wherever the code you wrote would be rendered in text
`t`: place the image at the top
`b`: place the image at the bottom
`p`: place the image on the page. There are more tags but these are the main ones that need to be known

### 6.2 Includegraphics environment

You can edit the image by using several tags, here are a few essential ones
`width=x`: in order to change the width of the image, `x` can be replaced with a number or you may also use the `linewidth` command to automatically fit the image within your margins.

`scale=x`: This changes the overall scale of the image on the final PDF.

### 6.3 Titration Plot

There are packages available for people in order to help create graphs such as `pgfplots`. In this case I simply input the numbers into Google Sheets and took a screenshot, I then used the same code from above to put it in as an image.

## 7 Mathematical Formulas

When writing math in LaTeX there are two ways to get into ”math mode”. You can use two $ to indicate and inline math equation. or you can use \[\] to indicate a display on a new line. For example the code snippets at the bottom will show the differences between the two.

```
The equation is $y = mx + c$
```

The equation is $y = mx + c$

This is an example of the inline math mode, if we use the display mode as such

```
The equation is
\[ y = mx + c \]
```

The equation is

$$y = mx + c$$

There are many different symbols available for use when in math mode. For instance, the entire greek alphabet is available for use. Other common symbols such as $\pi$ or $\infty$ can also be created by using the commands `pi` or `infty` in math mode respectively. There are plenty of resources online that list out all the difference symbols that can be created in LaTeX .

In order to do fractions you can use the following example

```
\frac{a}{b}
```

Which then displays $\frac{a}{b}$.
Superscripting and Subscripting is also very simple in LaTeX is also very simple. However, instead of using a command you use some of the reserved symbols. In order to superscript something you use the ^ character. For instance if we typed

```
$x^n$
```

we would get $x^n$. If you with to have multiple characters in the superscript you need to add curly brackets as such

```
 $x^{abc}$
```

This would then be rendered as $x^{abc}$. The same logic applied when subscripting anything. Instead you use the _ character to signify and subscript. Here is a short example

```
 $x_{abc}$
```

which would then display $x_{abc}$.

## 8 How To: Acknowledgements

I have not done this yet.

## 9 How To: References

In order to add a reference, you need to use the `thebilbliography` environment. Once created, you can create the citation by using the `bibitem` command. An example of this can be seen here.

```
\begin{thebibliography}{1}
\bibitem{ref1}
Citation goes here
\end{thebibliography}
```

From here you can use the label you gave in the `bibitem` commands to create in text citations by using the `ref` command.

## 10 Conclusion

In conclusion, I think that LaTeX is a good way to create academic papers as it gives users a lot of power with how they want their paper formatted. The number of features that the system provides makes it quite convenient, whether its creating math equations or creating references and in-text citations, LaTeX makes it easy to do all of this.

## Acknowledgements

have not done this section just yet

## References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.