

# Combined Tutorial for all L<sup>A</sup>T<sub>E</sub>X Users

Shane Miller

July 31, 2019

# Contents

<b>1</b>	<b>Installing <math>\text{\LaTeX}</math></b>	<b>1</b>
1.1	Windows Instructions . . . . .	1
1.2	Mac Instructions . . . . .	2
1.3	Linux Instructions . . . . .	2
<b>2</b>	<b><math>\text{\LaTeX}</math> Important Notes</b>	<b>3</b>
<b>3</b>	<b>General Formatting of a <math>\text{\LaTeX}</math> Document</b>	<b>3</b>
<b>4</b>	<b>Title Page</b>	<b>5</b>
<b>5</b>	<b>Table of Contents and Hyperlinking</b>	<b>6</b>
<b>6</b>	<b>Section Types: Main Section</b>	<b>6</b>
6.1	Section Types: Subsection . . . . .	7
6.1.1	Section Types: Subsubsection . . . . .	7
<b>7</b>	<b>Paragraphs</b>	<b>7</b>
<b>8</b>	<b>Text Formatting</b>	<b>8</b>
8.1	Font Size . . . . .	8
8.2	Font Style . . . . .	8
8.3	Verbatim . . . . .	8
<b>9</b>	<b>Lists</b>	<b>9</b>
9.1	Bulleted List . . . . .	9
9.2	Ordered List . . . . .	9
9.3	Combined Lists . . . . .	10
<b>10</b>	<b>Math</b>	<b>10</b>
<b>11</b>	<b>Document Formatting</b>	<b>11</b>
<b>12</b>	<b>Tables</b>	<b>12</b>
12.1	The Table Environment . . . . .	13
12.1.1	Setting-up the Table Environment . . . . .	13
12.1.2	Formatting Your Table . . . . .	13
12.2	Creating Your Table . . . . .	14
12.2.1	tabular . . . . .	14
12.2.2	tabularx . . . . .	16
<b>13</b>	<b>Images</b>	<b>17</b>
13.1	The Figure Environment . . . . .	17
13.2	Adding the Image . . . . .	18
13.3	Subfigures . . . . .	18
13.4	Image Examples . . . . .	19

13.4.1 Single Image . . . . .	19
13.4.2 Two Images . . . . .	20
<b>14 KOMA-Script</b>	<b>21</b>
<b>15 Use of Foreign Languages</b>	<b>22</b>
<b>16 Circuit Diagrams</b>	<b>22</b>
<b>17 Musical Composition</b>	<b>22</b>
<b>18 Flow Charts</b>	<b>22</b>
<b>19 3-Dimensional Graphics</b>	<b>22</b>
<b>20 Managing Large L<sup>A</sup>T<sub>E</sub>X Documents</b>	<b>22</b>
<b>21 Other Resources</b>	<b>23</b>

**Important Note:** A few of the examples in this document are specific to the article documentclass. For example, in the article documentclass sectioning starts at 1, but in other things such as report it would start at 0.1. Due to this, I would suggest starting your first project using the article document and then experimenting with other classes once you are comfortable with L<sup>A</sup>T<sub>E</sub>X. When you eventually switch to trying out different documentclasses, don't take everything here as exact. More or less things may be available for that documentclass. For example, I give an example of three commands you can use, but in other documentclasses, more or less commands may be available to use. \subtitle is available in the book documentclass, but not the article documentclass. In the intermediate tutorial I will go over the use of a different package that is commonly used instead of the default documentclass options L<sup>A</sup>T<sub>E</sub>X provides. Please take these things into consideration when using this document for reference.

## 1 Installing L<sup>A</sup>T<sub>E</sub>X

### 1.1 Windows Instructions

1. Go to [miktex.org](http://miktex.org).
2. Navigate to [Downloads](#).
3. Download the MiKTeX installer for Windows.
4. Run the MiKTeX installer once it finishes downloading.
5. Agree and continue.
6. Set preferred paper size and set "Install Missing Packages" to automatic.
7. When this is finished you will have the MiKTeX Console and the TeXworks editor. Make sure you have both.
8. Run the MiKTeX Console.
9. Select "Check for Updates" and allow it to install any it finds.
10. Once updates are finished installing, close the MiKTeX Console and open TeXworks.
11. Begin writing your first L<sup>A</sup>T<sub>E</sub>X document.
12. When saving, I recommend creating a LaTeX folder in which to store all of your L<sup>A</sup>T<sub>E</sub>X documents and then creating a subfolder for that specific document (eg: LaTeXDocuments → MyLaTeXProject → project files). When you save your .tex file, it will save a couple others there as well as the PDF when compiled.

## 1.2 Mac Instructions

1. Go to the MacTeX website [here](#).
2. Click [MacTeX Download](#).
3. Click on [MacTeX.pkg](#) to download MacTeX.
4. MacTeX.pkg will appear in your downloads folder once it is finished downloading. Double-click it to run the installer.
5. Follow the instructions in the installer to install MacTeX.
6. Once complete. navigate in finder to Applications.
7. In Applications, open the folder named “TeX.”
8. Inside it you will see an application named “TeX Shop.” Open it.
9. Begin writing your first  $\text{\LaTeX}$  document.
10. When saving, I recommend creating a LaTeX folder in which to store all of your  $\text{\LaTeX}$  documents and then creating a subfolder for that specific document (eg: LaTeXDocuments  $\rightarrow$  MyLaTeXProject  $\rightarrow$  project files). When you save your .tex file, it will save a couple others there as well as the PDF when compiled.

## 1.3 Linux Instructions

1. In the command line type: `sudo apt-get install texlive-full`.
2. In the command line type: `sudo apt-get install texmaker`.
3. In the command line type: `texmaker`.
4. The UI for TeXmaker should have booted. From there select: File  $\rightarrow$  New.
5. Begin writing your first  $\text{\LaTeX}$  document.
6. When saving, I recommend creating a LaTeX folder in which to store all of your  $\text{\LaTeX}$  documents and then creating a subfolder for that specific document (eg: LaTeXDocuments  $\rightarrow$  MyLaTeXProject  $\rightarrow$  project files). When you save your .tex file, it will save a couple others there as well as the PDF when compiled.

## 2 L<sup>A</sup>T<sub>E</sub>X Important Notes

- A L<sup>A</sup>T<sub>E</sub>X file might need to be compiled more than once.
  - This can happen for a couple of reasons, the main one being if something you use relies on something further down in the code.
  - The most common example of this is the title page. You will make sections after you use `\maketitle` to have L<sup>A</sup>T<sub>E</sub>X automatically make your title page. Because of this, your L<sup>A</sup>T<sub>E</sub>X will need to be compiled once to compile the rest of the document, and then one more time for your title page to add the sections to itself.
  - On Mac, if you add `% !TEX TS-program = pdflatexmk` to the beginning of your L<sup>A</sup>T<sub>E</sub>X document, that may compile it the correct number of times automatically if you have latexmk installed (which is often automatically installed with the distribution of L<sup>A</sup>T<sub>E</sub>X I included above).
  - On Linux, in TeXmaker, navigate to Options → Configure TeXmaker → Quick Build and add `latexmk -pdf %.tex` as a quick build command. If you need to for something else in the future, you can edit this command or make a different one.
- L<sup>A</sup>T<sub>E</sub>X is very powerful. It can do much more than make a simple text-based document. You can add images, hyperlinks, flow charts, musical scores, circuit diagrams, and much more. I will **not** cover everything L<sup>A</sup>T<sub>E</sub>X can do in this three-part tutorial series. I am only covering some of the more important basics and a select few of the more advanced features. If there is something you would like to do and it is not included in this tutorial series, please look it up online; there is often a package that allows you to do whatever you need doing.
  - A good source to look for answers is on the T<sub>E</sub>X StackExchange [here](#).
- After compiling your .tex file, some auxillary files will be added into the location your .tex file is. These auxillary files are simply used by L<sup>A</sup>T<sub>E</sub>X to help compile. You can ignore these files and only need to include the .tex when sending it to someone. It will also create a PDF file in the same location. It is up to you whether or not you want to include this when sending someone the .tex file.

## 3 General Formatting of a L<sup>A</sup>T<sub>E</sub>X Document

- When writing in L<sup>A</sup>T<sub>E</sub>X, indentation does not matter, but if you open the .tex file associated with this PDF, you can see the indentation convention I follow.
  - I indent based on “level.” This means that each time I go a level deeper (eg: section to subsection), I add one more indent.

- Additionally, I add an empty line between each unrelated piece of the document.
- A  $\text{\LaTeX}$  document's first line should always be a declaration of the document class.
  - This is done by typing `\documentclass{x}` where x is the type of document you are making.
  - The different document classes and their uses are as follows:

article	For articles in scientific journals, presentations, short reports, program documentation, invitations, et cetera.
IEEEtran	For articles with the IEEE Transactions format.
proc	A class for proceedings based on the article class.
report	For longer reports containing several chapters, small books, thesis, et cetera.
book	For full books.
slides	For slides. The class uses big sans serif letters.
memoir	For changing sensibly the output of the document. It is based on the book class, but you can create any kind of document with it.
letter	For writing letters.
beamer	For writing presentations.

- For more information please look [here](#).
- When you want to begin actually writing your document you use `\begin{document}`.
  - This allows you to start writing out text and using other commands shown later in this tutorial (such as `\section`).
- The area between `\documentclass` and `\begin{document}` is called the preamble.
  - This is the area where you set up the various things you may need for your document. You can use things like `\usepackage`, `\author`, and other things. I will cover these in more detail later.
- Anything that uses `\begin{x}` needs a `\end{x}` after all of the content inside that block.
- When trying to use reserved characters (characters reserved for commands in  $\text{\LaTeX}$ ), you need to use `\` to escape them. (Ex: To type \$ you need to type`\$`)

- Reserved characters include \$, {, }, #, %, \, and many others.
- An easy way to tell if the character you are using is a reserved character is the fact that it will be a different color from the rest of your text.
- The one exception to the rule of using \ to escape reserved characters is \ itself. To type \ in text you would use \backslash.
- To comment out a line in L<sup>A</sup>T<sub>E</sub>X (so that line isn't read at compile-time), preface the line with %.
- To type a new-line character in L<sup>A</sup>T<sub>E</sub>X, you use a \\. This is the equivalent of hitting return in various other word processors.
- If you wish to add text directly after a command that does not use {}, you need to encapsulate the command in \$'s.
  - To type \example in L<sup>A</sup>T<sub>E</sub>X you need to encapsulate the \backslash command in \$'s like this: `$\backslash$example`

## 4 Title Page

- To add a title page, simply add \title{Title Name Here}, \author{Author Name Here}, and \date{Date Here} to the preamble.
  - Note: If you would prefer to have the date be updated at compile-time rather than you having to change the date each day you are working on a document, leave the \date{Date Here} out of the preamble. In this case, the date will still be created and added to the title page, but it will automatically update what date it shows each time you compile your PDF.
- After your \begin{document} declaration add the following just below it:
  - \maketitle
  - \newpage
- If you don't want your title page numbered, add this command just above the \maketitle declaration:
  - \pagenumbering{gobble}
- You can also use \pagenumbering to begin the numbering in arabic numbering or roman numeral numbering. This can be done with either of the following commands:
  - \pagenumbering{arabic}
  - \pagenumbering{roman}
- If you call \pagenumbering{gobble} on your title page, your document will not start numbering things again until you call \pagenumbering{arabic/roman}.



## 5 Table of Contents and Hyperlinking

- To hyperlink in L<sup>A</sup>T<sub>E</sub>X, you need the hyperref package.
  - Get this by typing `\usepackage{hyperref}` in your preamble.
- To set up your hyperlinks to look nicer, add the following to your preamble after your `\usepackage` declarations:

```
\hypersetup{
  colorlinks = true,
  urlcolor = blue
}
```

- `colorlinks` colors your links rather than adding a box around them.
  - `urlcolor` sets the color of URL links to whatever color you choose. Default is a reddish-pink and here I set it to be a more common blue.
- If you plan on having a Table of Contents as well, use the following instead:

```
\hypersetup{
  colorlinks = true,
  linktoc = all,
  linkcolor = blue,
  urlcolor = blue
}
```

- `linktoc` determines what is linked on your table of contents. Here I set it to all, and thus everything on the table of contents for this document is linked.
  - `linkcolor` sets the color your table of contents links are. This color selection is separate from the URL hyperlink color set with `urlcolor`.
- To add a Table of Contents to your document simply use these two commands after the `\newpage` of your title page declaration:
  - `\tableofcontents`
  - `\newpage`

## 6 Section Types: Main Section

- Made by using `\section`.
- Sections change the number that comes before the first decimal and resets any subsections that come after it.

- If you were on section 1.2.6 and you called `\section`, you would be on section 2.
- Any text you want written in the section can go directly below your `\section` declaration.

## 6.1 Section Types: Subsection

- Made by using `\subsection`.
- Subsections change the number that comes after the first decimal and resets any subsections that come after it.
  - If you were on section 1.2.6 and you called `\subsection`, you would be on section 1.3.
- Any text you want written in the subsection can go directly below your `\subsection` declaration.

### 6.1.1 Section Types: Subsubsection

- Made by using `\subsubsection`.
- Subsubsections change the number that comes after the second decimal.
  - If you were on section 1.2.6 and you called `\subsubsection`, you would be on section 1.2.7.
- Any text you want written in the subsubsection can go directly below your `\subsubsection` declaration.
- There is nothing lower than a subsubsection. A subsubsubsection does not exist.

## 7 Paragraphs

### Main Paragraph

- Made by using `\paragraph`.
- Any text you want written in the paragraph can go directly below your `\paragraph` declaration.

### Subparagraph

- Made by using `\subparagraph`.
- Any text you want written in the subparagraph can go directly below your `\subparagraph` declaration.

## 8 Text Formatting

### 8.1 Font Size

- The following commands allow you to change the font size in L<sup>A</sup>T<sub>E</sub>X relative to the default document font size:

Command	Example	Description
<code>\tiny</code> Enter Text Here	Example Text	Makes the font size two units smaller.
<code>\small</code> Enter Text Here	Example Text	Makes the font size one unit smaller.
<code>\large</code> Enter Text Here	Example Text	Makes the font size one unit larger.
<code>\huge</code> Enter Text Here	Example Text	Makes the font size two units larger.

### 8.2 Font Style

- The following commands allow you to use different font styles in L<sup>A</sup>T<sub>E</sub>X:

Command	Example	Description
<code>\textbf{}</code> Enter Text Here	<b>Example Text</b>	Bold
<code>\textit{}</code> Enter Text Here	<i>Example Text</i>	Italic
<code>\texttt{}</code> Enter Text Here	<b>Example Text</b>	Typewriter
<code>\textrm{}</code> Enter Text Here	Example Text	Serif (Roman)
<code>\underline{ }</code> Enter Text Here	<u>Example Text</u>	Underline

### 8.3 Verbatim

- If you want to type a block of code or anything that may contain L<sup>A</sup>T<sub>E</sub>X commands, you can use the verbatim environment using `\begin{verbatim}` and `\end{verbatim}`.
  - The verbatim environment will write whatever is in it as text without running any commands that may be in it.

- The `verbatim` environment can't recognize tabs, but can recognize spaces. If you need to use a tab, convert them into spaces. I've found that six spaces indents similarly to a tab in the  $\text{\LaTeX}$  editor.
- If you want to use the `verbatim` environment in-line, then you can use `\verb|text here|`.

## 9 Lists

### 9.1 Bulleted List

- To make a bulleted list in  $\text{\LaTeX}$ , you need to use the `itemize` declaration within `\begin{}` and `\end{}`.
  - This will look like this: `\begin{itemize}` and `\end{itemize}`.
- For each bullet point you need to use `\item` followed by a space and then whatever it is you want after the bullet. This will go between the `\begin` and `\end`.
- If you want to make a sub-bullet point (the dashes in this document), within the `\begin{itemize}` and `\end{itemize}`, you need to add another `\begin{itemize}` and `\end{itemize}`.
  - Between the second set of `itemize`, you use `\item` followed by a space and then whatever it is you want after the bullet.

### 9.2 Ordered List

- To make a numbered list in  $\text{\LaTeX}$ , you need to use the `enumerate` declaration within `\begin{}` and `\end{}`.
  - This will look like this: `\begin{enumerate}` and `\end{enumerate}`.
- For each bullet point you need to use `\item` followed by a space and then whatever it is you want after the bullet. This will go between the `\begin` and `\end`.
- If you want to make a sub-enumeration point (indenting and labeling with a, b, c, et cetera instead of 1, 2, 3, et cetera), within the `\begin{enumerate}` and `\end{enumerate}`, you need to add another `\begin{enumerate}` and `\end{enumerate}`.
  - Between the second set of `enumerate`, you use `\item` followed by a space and then whatever it is you want after the bullet.

### 9.3 Combined Lists

- You can use both itemize and enumerate together. This will give you a mix of bulleted and numbered lists.
  - For Example: If you have a numbered list, but want to include a few bullet points regarding one of the numbers, you can nest an itemized list in it if you don't want to number those points in an order.
- You can nest as many lists as you would like.

## 10 Math

- To begin you should use the package `amsmath` by calling `\usepackage{amsmath}` in the preamble.
  - While  $\text{\LaTeX}$  does have a way to type an equation without a package (using `$$equation$$`), it is highly discouraged as `amsmath` is better in every way. It uses the default  $\text{\LaTeX}$  math as a base, so there shouldn't be anything in default  $\text{\LaTeX}$  math that you can't do in `amsmath`.
- When writing math, you want to use `\begin{equation}`, your equation, and then `\end{equation}`.
- Using this without a package will automatically number each line.
- To avoid this, you can use `\begin{equation*}`, your equation, and then `\end{equation*}`.

Here is an example of  $f(x) = x^2/4^\sigma$  as a  $\text{\LaTeX}$  equation:

$$f(x) = x^2/4^\sigma \tag{1}$$

Here is an example of  $f(x) = x^2/4^\sigma$  as an `amsmath` equation:

$$f(x) = x^2/4^\sigma$$

- In  $\text{\LaTeX}$ , you can type a bunch of mathematical symbols using various commands. As seen above, `\sigma` will get you  $\sigma$ . Please see [here](#) or [here](#) for information relating to what commands can be used.
- To use an equation or any math command within a normal sentence (such as this one), use `\(equation here\)` wherever you want the equation in the sentence.

- The parenthesis escaped by the `\` will not show up in the equation.  
If you want any parenthesis in your equation, use non-escaped parenthesis inside the escaped parenthesis.
- Here are a few example commands:
  - You can use the command `\frac{x}{y}` to make a fraction look like  $\frac{x}{y}$  instead of  $x/y$ .
  - You can use the command `x \times y` or `x \cdot y` to make multiplication look like  $x \times y$  or  $x \cdot y$  instead of  $x * y$ .
  - `\sqrt{x}` will make  $\sqrt{x}$ .
  - `x_{y}` will make a subscript such as  $x_y$ .
  - `x^y` will make a superscript (or exponent) such as  $x^y$ .
  - `\text{text here}` will allow you to type regular text within an equation block.
  - Encapsulating your equation in `\boxed{equation here}` will allow you to put a box around your equation.

## 11 Document Formatting

- To format a document, you need to add options to your `\documentclass` declaration.
- If you don't set anything in the options section, they will default to whatever the environment default on your machine is.
  - The two most common defaults are 10pt font, and letter paper size.
- To add options for your documentclass, you need to add “[options here]” between your documentclass declaration and your class parameter.
  - Here's how that would look: `\documentclass[options here]class here`.
- The main options are as follows:

10pt, 11pt, et cetera	Sets the size of the main font in the document.
letterpaper, a4paper, legalpaper, et cetera	Defines the paper size for the document.
fleqn	Display formulas left-aligned rather than centered.
leqno	Places the numbering of formulas on the left-hand side rather than the right.
titlepage, notitlepage	Specifies whether a new page should be started after the document title or not. The article class does not start a new page by default, while report and book do.
twocolumn	Instructs LaTeX to typeset the document in two columns instead of one.
twoside, oneside	Specifies whether double or single sided output should be generated. The classes article and report are single sided and the book class is double sided by default. Note that this option concerns the style of the document only. The twoside option does not tell the printer you use that it should actually make a two-sided printout.
landscape	Changes the orientation of the document to landscape.
openright, openany	Makes chapters begin either only on right hand pages or on the next page available. This does not work with the article class, as it does not know about chapters. The report class by default starts chapters on the next page available and the book class starts them on right hand pages.
draft	Makes LaTeX indicate hyphenation and justification problems with a small square in the right-hand margin of the problem line so they can be located quickly by a human. It also suppresses the inclusion of images and shows only a frame where they would normally occur.

- Here is an example of what you would write if you wanted a two-columned, double-sided article with 12pt font on landscape size A4 paper:
  - `\documentclass[12pt, a4paper, landscape, twocolumn, twoside]{article}`
  - Note: the order of the options does not matter.

## 12 Tables

- In this tutorial, I am only going to go over the “basics” of how to make a table. If you need to do something not included in this tutorial, please see the [L<sup>A</sup>T<sub>E</sub>X WikiBooks table page](#).

## 12.1 The Table Environment

### 12.1.1 Setting-up the Table Environment

- To begin making a table you may want to start by using the table environment.
  - Although it isn't necessary, it can sometimes be useful.
  - Throughout this document, I never used the table environment because it was not necessary to format my tables.
- The table environment is used to set-up various table settings.
- To declare the table environment you would use `\begin{table}` and `\end{table}`.

### 12.1.2 Formatting Your Table

- After you declare your table environment, you can use options to set the table location.
  - This can be done by modifying your declaration to look like this:  
`* \begin{table}[location option]`
  - The list of possible positions are as follows:

h	Where the table is declared (here).
t	At the top of the page.
b	At the bottom of the page.
p	On a dedicated page of floats (objects such as tables and pictures).
!	Override the default float restrictions.

- The “!” option tries to force L<sup>A</sup>T<sub>E</sub>X to use one given position based on what option you chose, but will not force it if it is impossible.
  - You can use more than one position option. Doing this will treat it as a “wishlist” where L<sup>A</sup>T<sub>E</sub>X will go through each one in order trying to see if the table fits well in that position.
  - If you don't set any position option, the default is `/textttbtp`.
  - If you choose not to have a table environment, you can set the location option on the environment you actually use to create the table, but in most cases, you will not need to set the position if you aren't using the table environment.
- To center your table in the given space, you can use `\centering`.
  - `\centering` should be added inside your table environment, but just before you create your table.
- To add a title to your table, use `\caption`.



- If you choose to have a title for your table, this command typically is added directly after your table environment declaration.
- Another thing you can add in the table environment is a reference label.
  - A reference label allows you to reference the table just like you would any other reference.
  - The command for a reference label looks like this:
 

```
* \label{table:AddTableReferenceNameHere}
```

## 12.2 Creating Your Table

- When creating your table, there are five options:
  - tabular
  - tabularx
  - tabulary
  - tabular\*
  - tabu
- In this tutorial, I will only be going over tabular and tabularx.
- For information related to tabulary, please see [here](#).
- For information related to tabular\*, please see [here](#).
- For information related to tabu, please see [here](#) and [here](#).

### 12.2.1 tabular

- tabular is the original table maker. tabularx, tabulary, tabular\*, and tabu are all spinoffs of tabular.
- To use tabular, you would set up the tabular environment using `\begin{tabular}[pos]{table specifications}` and `\end{tabular}` where [pos] is optional.
- If you would like to define the height each row should be you can add the command `\def\arraystretch{insert number here}` just above where you set up the tabular environment. The default value is 1, so 1.5 would be 150% taller.
- Table Specifications:
  - The table specifications are where you define the layout of your table.

- The options are as follows:

<code>l</code>	Left-justified column.
<code>c</code>	Centered column.
<code>r</code>	Right-justified column.
<code>p{‘width’}</code>	Paragraph column with text vertically aligned at the top.
<code>m{‘width’}</code>	Paragraph column with text vertically aligned in the middle (requires the package <i>array</i> ).
<code>b{‘width’}</code>	Paragraph column with text vertically aligned at the bottom (requires the package <i>array</i> ).
<code> </code>	Vertical line.
<code>  </code>	Double vertical line.

- For example, if you wanted a table with three centered columns separated by vertical lines, you would use `\begin{tabular}{c | c | c}`.
- Tables in  $\text{\LaTeX}$  will not automatically wrap text in calls and will overrun the width of the page if long enough. For these columns that will contain long texts, it is recommended to use a paragraph column.
  - \* To specify length of a paragraph column, you would include it using `{enter length here}`.
  - \* Specifying a `p` column with a size of 5cm will look like this: `p{5cm}`

- Table Content:

- Your table content will go inside of your tabular environment.
- Each line can be thought of as a row (while this isn’t entirely correct, it is how it is most often used and is the easiest way to actually write them).
- In each row, you can use these commands to define where your content should go in your table:

<code>&amp;</code>	Column separator.
<code>\\</code>	Start new row.
<code>\hline</code>	Horizontal line.
<code>\newline</code>	Start a new line within a cell (in a paragraph column).
<code>\cline{i-j}</code>	Partial horizontal line beginning in column <i>i</i> and ending in column <i>j</i> .

- Using the three column example from earlier, if we wanted the numbers 1, 2, and 3 to be in each column respectively, you would use `1 & 2 & 3\\` for the row.

- Example Table:

1	2	3
4	5	6
7	8	9

- For the example table above, the code is as follows:

```
\begin{tabular}{| c | c | c |}
\hline
1 & 2 & 3\\
\hline
4 & 5 & 6\\
\hline
7 & 8 & 9\\
\hline
\end{tabular}\\
```

### 12.2.2 tabularx

- To use tabularx, you need to add `\usepackage{tabularx}` to your document's preamble.
- tabularx adds a new column type X.
  - The X specifier defines a column to stretch to make the table as wide as specified.
- When using tabularx, you need to define your table width when you create your tabularx environment.
  - Setting up a tabularx environment for a table half the width of what a line of text is with two equal-length columns with a vertical line between them would look like this: `\begin{tabularx}{0.5\textwidth}{X | X}`
- Other than a that and couple of things less-important things I won't be going over in this tutorial, tabularx functions the same as tabular. Everything in the tabular section applies to this section as well.
- Please see [here](#) for more tabularx information.

## 13 Images

### 13.1 The Figure Environment

- The figure environment is where you set up your image details.
- If you only have one image to include, then this is also where you include your image.
  - If you have more than one image to include in this area, then you actually will use subfigure to include the images. I will go over this in more detail soon.
- To begin, you *need* to use the package graphicx, otherwise nothing will work.
  - You can add the package by adding `\usepackage{graphicx}` to your preamble.
- To create the figure environment, you would use `\begin{figure}` and `\end{figure}`.
  - On the begin, you can include a position option in the same way as a table. Just in case you haven't read that section of this tutorial, I will include the details again below.
  - Setting the figure environment position can be done by modifying your declaration to look like this:  
`* \begin{figure}[location option]`
  - The list of possible positions are as follows:

h	Where the figure is declared (here).
t	At the top of the page.
b	At the bottom of the page.
p	On a dedicated page of floats (objects such as tables and pictures).
!	Override the default float restrictions.

- The “!” option tries to force L<sup>A</sup>T<sub>E</sub>X to use one given position based on what option you chose, but will not force it if it is impossible.
- You can use more than one position option. Doing this will treat it as a “wishlist” where L<sup>A</sup>T<sub>E</sub>X will go through each one in order trying to see if the image fits well in that position.
- If you don't set any position option, the default is *tbp*.

## 13.2 Adding the Image

- To include an image you need to use `\includegraphics` to define the image width and file location. This will look like this:
  - `\includegraphics[width = width option]{file location}`
- Typically, when defining the width it's easiest to use `\linewidth` as a basis.
  - `\linewidth` is the width of the left-most text can be to the right-most margin.
  - Adding a number before `\linewidth` acts as a multiplier. `0.4\linewidth` would be 40% of the width of the page.
  - Images will scale to be the defined width. If they are smaller, they will expand to the correct size. And if they are larger, they will shrink to the correct size.
- When defining your file location, it's usually easiest to include a folder named "images" where your document is stored to include all your images. This way, this file location will look like "images/image\_name".

## 13.3 Subfigures

- Using subfigures allows you to include more than one image in a defined area.
- When using subfigures, you should use the package *subcaption* in addition to *graphicx*.
- Subfigures work the same way as regular figures do. The only difference if you put the subfigure environment in the figure environment.
  - You can make the subfigure environment using `\begin{subfigure}` and `\end{subfigure}`.
- Please see the examples below to better understand how a subfigure fits into the figure environment.

## 13.4 Image Examples

### 13.4.1 Single Image

- Example Image:



Figure 1: The L<sup>A</sup>T<sub>E</sub>X Logo

- The code for the image above is as follows:

```
\begin{figure}[h!]  
  \centering  
  \includegraphics[width=0.5\linewidth]{images/LaTeXLogo.jpg}  
  \caption{The \LaTeX{} Logo}  
\end{figure}
```

### 13.4.2 Two Images

- Example Images:

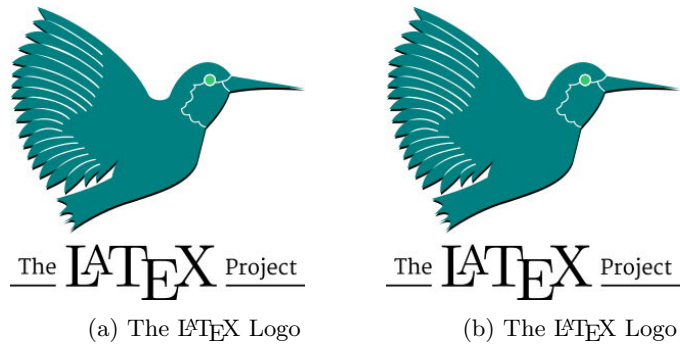


Figure 2: Two Images Side-by-Side

- The code for the images above are as follows:

```
\begin{figure}[h!]  
  \centering  
  \begin{subfigure}[b]{0.4\linewidth}  
    \includegraphics[width=0.8\linewidth]{images/LaTeXLogo.jpg}  
    \caption{The \LaTeX{} Logo}  
  \end{subfigure}  
  \begin{subfigure}[b]{0.4\linewidth}  
    \includegraphics[width=0.8\linewidth]{images/LaTeXLogo.jpg}  
    \caption{The \LaTeX{} Logo}  
  \end{subfigure}  
  \caption{Two Images Side-by-Side}  
\end{figure}
```

## 14 KOMA-Script

- In this tutorial I will not be going over exactly how to use KOMA-Script. Instead, I will explain what it is and why it is used. In addition to that, I will link the KOMA-Script manual.
- KOMA-Script is used as a replacement for the default document classes that L<sup>A</sup>T<sub>E</sub>X comes with.
- Most T<sub>E</sub>X editors will come with KOMA-Script pre-installed and ready to use, but that doesn't necessarily mean every editor will have it.
  - You can find out if it is or isn't installed by either checking your installed packages or simply trying to use it.
    - \* If you try the latter and it isn't installed, then it may either install it for you or throw an error depending on the T<sub>E</sub>X editor you are using.
    - \* If it throws an error, you will need to manually install it using your editor's console.
- KOMA-Script consolidates a lot of the class-specific features in the regular document classes.
  - For example, \subtitle can't be used in the regular article document class, but can be used in KOMA-Script's article document class.
- Additionally, KOMA-Script already includes some packages, so you wouldn't need to use \usepackage for those packages.
- It is often used in the L<sup>A</sup>T<sub>E</sub>X community as most people see KOMA-Script as a better alternative to the regular document classes with little to no downside.
- The KOMA-Script document classes are not necessary to use. None of these tutorials were written using the a KOMA-Script document class, but I have written other documents using them. You'll have to decide for yourself if you want to use them.
- For more information, please check out the translated KOMA-Script Manual [here](#).



- 15 Use of Foreign Languages
- 16 Circuit Diagrams
- 17 Musical Composition
- 18 Flow Charts
- 19 3-Dimensional Graphics
- 20 Managing Large L<sup>A</sup>T<sub>E</sub>X Documents

## 21 Other Resources

Here is a list of other great resources regarding learning LaTeX. Most of these were used throughout making this tutorial and contain some information not included in this tutorial.

- [RPI L<sup>A</sup>T<sub>E</sub>X Intro](#)
- [A Simple Guide to L<sup>A</sup>T<sub>E</sub>X](#)
- [L<sup>A</sup>T<sub>E</sub>X Tutorial Figures Page](#)
- [University of Edinburgh's L<sup>A</sup>T<sub>E</sub>X for Beginners](#)
- [T<sub>E</sub>X StackExchange](#)
- [Short Math Guide for L<sup>A</sup>T<sub>E</sub>X](#)
- [L<sup>A</sup>T<sub>E</sub>X Math Command Cheat Sheet](#)
- [University of Leeds Table Help Doc](#)
- [L<sup>A</sup>T<sub>E</sub>X WikiBooks Home Page](#)
- [L<sup>A</sup>T<sub>E</sub>X WikiBooks Table Page](#)
- [KOMA-Script Manual](#)
- The .tex file included in the zip for this document has everything that was used to write this tutorial. Please look through it to see how things in this document were used in practice.