# Introduction to \LaTeX

NMT Office for Student Learning

May 18, 2016

Written by Ian Jones
Revised by Caleb Hightower

## Contents

1. What is \LaTeX? ...................................................... 2  
2. Your First \LaTeX Document ..................................... 2  
   2.1 Getting Started .............................................. 3  
3. Document Structure .............................................. 4  
   3.1 Titles, Dates, and Authors .................................. 4  
   3.2 Sections and Subsections .................................... 5  
   3.3 Reserved Characters ......................................... 5  
4. Images ....................................................................... 6  
5. Equations .................................................................... 7  
   5.1 Math Environment .............................................. 8  
   5.2 Other Symbols .................................................. 10  
6. Tables of Contents .................................................. 10  
7. Other Common Commands ......................................... 10  
   7.1 Text Appearance ............................................... 10  
   7.2 Lists ..................................................................... 11  
   7.3 Quotations ........................................................ 12  
   7.4 Tables ............................................................... 12  
8. Bibliographies ....................................................... 13  
9. \LaTeX Code for This Document ................................ 14  
10 Appendices ............................................................. 15
1 What is \LaTeX ?

\LaTeX (pronounced LAH-tek) is a typesetting language. This means that it takes text and commands in a scripted format, and then decides how best to present them on the page. Fortunately, it is very good at doing this, so even a rough \LaTeX document ends up looking very professional, and a well-written \LaTeX document can look better than almost anything Microsoft Word or similar programs can produce. This document was prepared in \LaTeX, and shows what a typical document will look like.

The best part about \LaTeX is that every single thing on the page can be modified to your preference. Some things are more difficult to modify than others, but it can still be done if you are willing to work a little extra. While these things can all be changed, \LaTeX is smart; if you don’t specify how you want something to look, it will put it on the page the best way it sees fit. It also handles the page spacing automatically. If you have one extra line at the bottom of the page, rather than making a whole separate page for that one line, \LaTeX will adjust spacing to fit it all onto a single page.

Things like equations, bibliographies, and images can be incredibly difficult to manage in other document creators (looking at you, MS Word). But in \LaTeX, they can take only a few seconds of typing commands to present them exactly how you want.

\LaTeX is the standard for typesetting scientific papers in many fields, and should be used for any kind of important presentation. Additionally, it looks nice and shows that thought and effort was put into the information being communicated. \LaTeX is a freely available program, and is already installed on all TCC machines.

Proficiency in \LaTeX is an excellent skill to have. It looks good on a resumé, and instantly makes you more hireable. Using a program like \LaTeX will make a very favorable impression on professors especially for large assignments, such as take-home tests and research papers.

2 Your First \LaTeX Document

This guide is written to be used on a New Mexico Tech TCC machine booted into Windows. If you prefer to use Linux, see Appendix 2 at the end of this document before moving on to learn how to write \LaTeX documents that way. If you do not have access to a TCC computer and would like to install \LaTeX on your own computer, see Appendix 1.

To open \LaTeX on a TCC machine in Windows, go to Start → All Programs → MiKTeX 2.9 → TeXworks
2.1 Getting Started

Once you’ve done that, we can begin writing our document! The first thing that needs to be done is to tell \LaTeX what kind of document we will be writing. The most common type of document that you will be using is article, however there are many other options. Since we just want a simple example, we can just use
an article format with \LaTeX{} defaults. This is done using the following code:
\documentclass{article}

Common alternatives to article include minimal (primarily for debugging), report (for a very long academic paper), book, beamer (for a slide presentation), and letter. We could also specify things like what font to use in this line. For example if we wanted to write a book in 12 point font, we would write:
\documentclass[12pt]{book}

The next thing that needs to be done is telling \LaTeX{} where your document starts and stops. This is done using the commands:
\begin{document}
\end{document}

These should both go underneath your document class line. Now whatever you type in between the begin document and end document lines will be automatically typeset by \LaTeX{} for you! A simple document that just prints out “Hello World!” is shown below.
\documentclass{article}
\begin{document}
Hello World!
\end{document}

Typing this into \LaTeX{}, and hitting compile (the green play button in the top left corner) will now create your first \LaTeX{} document.

3 Document Structure

Writing a bare text file is almost never enough. You will need to include titles, authors, and dates, as well as be able to structure the document in a logical manner.

3.1 Titles, Dates, and Authors

Information on what a paper is about, who wrote it, and when should be included in every paper. \LaTeX{} expects to receive this information immediately after the begin document line. This information is given by the commands \texttt{\author, \date, and \title} in any order, followed by a \texttt{\maketitle} command. For example, the code that was used to make the heading at the top of this document was:
\begin{document}
\title{Introduction to \LaTeX{}}
\author{NMT Office for Student Learning}
\date{\today}
\maketitle

The \today that you see in the date simply tells \LaTeX{} to use the current date when it is typesetting.

### 3.2 Sections and Subsections

Sections and subsections serve to organize your work in a logical manner, and allow a reader interested only in one part of what you’ve done to quickly find it. These are created by using the commands \texttt{\section{}} and \texttt{\subsection{}}. If you wanted to go into still finer divisions, \texttt{\subsubsection{}}, \texttt{\paragraph{}}, and \texttt{\subparagraph{}} can be used. If you are using anything below subsection you probably need to think about the structure of your document. For example some section and subsection in this document were created using the commands:

\begin{verbatim}
\section{Document Structure}
\subsection{Titles, Dates, and Authors}
\end{verbatim}

Notice that there are no numbers in the above commands. This is because \LaTeX{} automatically numbers your sections and subsections in order that they appear in your code. Need to add in a section right in the middle of your document? No problem; all of the subsequent sections will automatically be re-numbered accordingly.

### 3.3 Reserved Characters

Some characters are reserved by \LaTeX{} for doing special commands. These characters cannot be typed into normal text. These characters are: #, $, %, ^, &, _, {, }, ~, and \. In each case, these characters can be produced in plain text by using the commands:

\begin{verbatim}
\#, \$, \%, \^{}, \&, \_, \{, \}, ~{}, and \textbackslash
\end{verbatim}

Accidentally including a reserved character improperly is one of the reasons that a \LaTeX{} code will fail to compile, or will compile in an unexpected way. Often, error messages in such a failure will let you know this is the case so you can find the mistake in your code and fix it.

---

New Mexico Tech — Spring 2016 5
4 Images

Images can be added to \LaTeX documents. However, unlike other document programs the image insertion is not a drag-and-drop process; we need to put a bit more work in. \LaTeX alone cannot process these images, but packages have been developed as an extension to \LaTeX. There are many packages that have been developed by the makers of \LaTeX, all of which include different specialized sets of commands. To include a particular package in your document, use the following command.

\usepackage{packagename}

The packages that we will be using in this workshop for images are called graphicx and float. To use these packages, place their respective import commands after the \documentclass{article} command but before the \begin{document} as follows:

\documentclass{article}
\usepackage{graphicx}
\usepackage{float}
\begin{document}

Once this is done, an image can be included by adding the following set of commands:

\begin{figure}
\centering
\includegraphics[width=3in]{galaxy}
\caption{A sample image.}
\end{figure}

This code produces the image seen on the next page.

Let us dissect exactly what this code does. The commands \begin{figure} and \end{figure} tell \LaTeX that you want to go into the figure environment. In \LaTeX, an “environment” is a section of code where particular kinds of commands are valid. Therefore, some commands only work when you are in the associated environment. The figure environment is generally the best one to use for images, as it gives you access to many cool features, like figure numbering and captions. You can insert images outside of this environment, but this is not recommended.

The \centering command tells \LaTeX that the image is to be centered in the page. The line \includegraphics[width=3in]{galaxy} is what actually generates the image. It tells \LaTeX to look in the same folder where the .tex file is saved, find the file “galaxy”, and print it on the page with a width of 3 inches. It gives \LaTeX some flexibility to pick an appropriate place in the
document. Most common file extensions are recognized by \LaTeX, but usually .png or .jpg work best. The important part is the image must be saved in the same place as the .tex file.

Here, \LaTeX has placed the image on the page in the best way it can. However, if you’d like to override this and place the image at a precise spot in your document (in the same position as where the code is), replace the command

\begin{figure}

with the command

\begin{figure} \[H\]

This will tell \LaTeX, “No, I want this right here.”

Anytime something appears in square brackets in a command, that indicates it is an optional parameter; \LaTeX doesn’t need them in order to compile. This is what the additional [H] is in the above command. The letter H could be replaced by a t or b for placement at the top or bottom of the page respectively.

5 Equations

The primary area where \LaTeX excels over other programs is typesetting equations. \LaTeX has some basic equation functionality built-in, but using another package will be very helpful. In this tutorial, we will use the amsmath package, so you will need to add the corresponding command $\texttt{\usepackage{amsmath}}$ at
5.1 Math Environment

The math environment is what allows you to generate equations in a document. There are two types of math environments: inline math, and centered math.

Inline math, as the name implies, creates math expressions within a line of text by surrounding the relevant equations by $ signs. For example, the command

The equation of the line is $y = 3x - 4$.

produces the output:

The equation of the line is $y = 3x - 4$.

Centered math takes equations and presents them in the middle of the page, with a bit of extra space above and below for emphasis. If you want your equations to be numbered, then the commands \begin{equation} and \end{equation} should be used. As with sections and figures, \LaTeX\ handles the numbering of equations for you automatically. The commands

\begin{equation}
ax^2 + bx + c = 0
\end{equation}

produce the output

$ax^2 + bx + c = 0$ \hfill (1)

However, if you want your equation to be centered but not have a number, the commands [ and ] should be used to open and close the environment.

[ ax^2 + bx + c = 0 ]

produces:

$ax^2 + bx + c = 0$

It should be noted that many symbols, such as the summation symbol or fractions, look slightly different in centered math compared to inline math. Test out some of the commands below in both centered and inline to see the difference.
Exponents and Subscripts

Exponents and subscripts can be created as follows:

\[ e^x \quad x^2 \quad F_{total} \]

Which produces: \( e^x \), \( x^2 \), and \( F_{total} \).

Fractions

Fractions are produced using the \texttt{\textbackslash frac} command. This command works as follows:

\[ \frac{numerator}{denominator} \]

\[ \frac{A}{B} \]

This code produces: \( \frac{numerator}{denominator} \)

Greek Letters

Greek letters are produced by simply putting a \texttt{\textbackslash} before the name of the Greek letter you want. If you capitalize the first letter in the name of the Greek letter, you will get the capitalized version. For example:

\[ \alpha \quad \pi \quad \Sigma \quad \sigma \quad \tau \quad \theta \]

produces: \( \alpha \pi \Sigma \sigma \tau \theta \)

Special Functions

Several common functions, such as trigonometric functions or logarithms, have special commands in \LaTeX. Usually, it is just a \texttt{\textbackslash} before the usual name of the function. For example,

\[ \sin(\theta) \quad \log_2(y) \quad \arctan(x) \]

produces: \( \sin(\theta) \), \( \log_2(y) \), and \( \arctan(x) \). The command

\[ \sin^2(x) + \cos^2(x) = 1 \]

produces:

\[ \sin^2(x) + \cos^2(x) = 1 \]
Large Symbols

Large math symbols involving bounds or limits usually involve the symbol name and upper/lower limits typed as super- or sub-scripts.

\[ \sum_{i=1}^{\infty} n_i \]
\[ \int_{a}^{b} f(x) \, dx \]
\[ \lim_{x \to 0} h(x) \]

This produces:

\[
\sum_{i=1}^{\infty} n_i \quad \int_{a}^{b} f(x) \, dx \quad \lim_{x \to 0} h(x)
\]

5.2 Other Symbols

\LaTeX{} is capable of typesetting almost every math symbol that you will ever need if you give it the correct command. Visit http://web.ift.uib.no/Teori/KURS/WRK/TeX/symALL.html for an extensive list of symbols.

6 Tables of Contents

Many longer documents are required to have a table of contents. \LaTeX{} will automatically generate this for you by placing the

\texttt{\tableofcontents}

command immediately after your \texttt{\maketitle} command. \LaTeX{} will automatically generate your table of contents based on your section and subsection titles. NOTE: you have to compile the document \emph{twice} to make the table of contents appear.

7 Other Common Commands

7.1 Text Appearance

Italicized text can be created using the command

\texttt{\textit{This is the text to be italicized}}

This command produces the following:

\textit{This is the text to be italicized}

Bold text can be created using the command

\texttt{\textbf{This is the text to be bolded}}
This command produces the following:

This is the text to be bolded

Underlined text can be created using the command

\underline{This is underlined text}

This command produces the following:

This is underlined text

If you want a bit of extra space in your document, use the \hspace{value} or \vspace{value} commands for horizontal and vertical space respectively, and replace value with whatever amount you want. \LaTeX{} recognizes most standard units, so you could use a command like \vspace{2in} to create a vertical empty space 2 inches tall.

### 7.2 Lists

There are a few types of list environments in \LaTeX{}. The two most popular are itemize, which creates a bulleted list; and enumerate, which creates a numbered list.

To create lists, use the \begin{enumerate} and \end{enumerate} commands, or the \begin{itemize} and \end{itemize} commands. To place items in the list, use the command \item followed by that item’s text between these commands. The code

\begin{itemize}
  \item One thing
  \item Another thing
  \item Yet Another thing
\end{itemize}

\begin{enumerate}
  \item The first thing
  \item The second thing
  \item The third thing
\end{enumerate}

produces the following output:

- One thing
- Another thing
- Yet Another thing

1. The first thing
2. The second thing
3. The third thing
7.3 Quotations

If you want to put something in quote marks, the usual double quote key on your keyboard doesn’t quite work. Instead, you want to start a quote by putting two tick marks. (This is the symbol on the same key that has the ~ mark. It’s the key to the left of the 1 key on standard US keyboards.) You’ll want to end a quote using two single quote marks (apostrophes).

‘‘This is a quotation.’’

produces: “This is a quotation.”

7.4 Tables

Producing tables in \LaTeX\ are a bit more tricky than some other things. This is because there are a couple of more details to manage to make sure it gets put on the page properly. The best way to insert a table is to use the figure environment just as with images. The table itself is created in the \texttt{tabular} environment within the \texttt{figure} environment. Here’s an example of a table.

\begin{figure}[H]
  \centering
  \begin{tabular}{ccc}
    Name & Age & Height \\
    \hline
    Jeff & 21 & 6'1'' \\
    Amy & 19 & 5'8'' \\
    Sam & 24 & 5'10'' \\
    \hline
  \end{tabular}
  \caption{Random humans’ ages and heights.}
\end{figure}

This produces the output:

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jeff</td>
<td>21</td>
<td>6'1''</td>
</tr>
<tr>
<td>Amy</td>
<td>19</td>
<td>5'8''</td>
</tr>
<tr>
<td>Sam</td>
<td>24</td>
<td>5'10''</td>
</tr>
</tbody>
</table>

Figure 3: Random humans’ ages and heights.

On the line opening \texttt{tabular}, you must place in curly braces a set of letters indicating how many columns you will have, and how the text in each of those columns should be centered. The letter options are \texttt{l}, \texttt{c}, and \texttt{r}, for left, right, and centered, respectively. If you want to have vertical lines between the columns or on the edges, the | symbol should be placed in these curly braces in the places
you want them. For horizontal lines, you’ll place the command \hline between
the rows where you want it to go.

Rows should have a \ \ placed at the end of the line where you want that
row to end. Individual cells in a row are separated by the & symbol.

Here is an example of another slightly different table.

\begin{figure}
\centering
\begin{tabular}{|l|r|r|}
\hline
\textbf{Element} & \textbf{Atomic Weight} & \textbf{Atomic Number} \\
Hydrogen & 1.008 & 1 \\
Helium & 4.003 & 2 \\
Lithium & 6.941 & 3 \\
\hline
\caption{First three elements.}
\end{tabular}
\end{figure}

produces:

<table>
<thead>
<tr>
<th>Element</th>
<th>Atomic Weight</th>
<th>Atomic Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydrogen</td>
<td>1.008</td>
<td>1</td>
</tr>
<tr>
<td>Helium</td>
<td>4.003</td>
<td>2</td>
</tr>
<tr>
<td>Lithium</td>
<td>6.941</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 4: First three elements.

8 Bibliographies

\LaTeX has many fancy bibliography packages. In this workshop, we are only
going to explore a simple one. We will use the \texttt{thebibliography} environment.
To do this, at the end of your document, just before the \texttt{\end{document}} com-
mand, put the following code:

\begin{thebibliography}{99}
\end{thebibliography}

This places you in the bibliography environment. The 99 parameter that
follows the \texttt{\begin{thebibliography}} command simply tells the program how
wide to make the citations. If you have 1-9 sources, this can be any 1 digit
number. If you have 10-99 sources, this can be any two digit number, and so
on. You can now create a citation using the command

\bibitem{Tag}
(Source citation in any format.)
This citation can now be referenced anywhere in your paper by using the \cite{tag} command. The tag is just any word or phrase that lets you know what the source is - it will never appear in your final document. Author’s last name followed by year of publication is a good one to use.

Now, using the following code, we will generate the “References” section.

\begin{thebibliography}{9}
\bibitem{Copernicus}
Nicholas Copernicus

De revolutionibus orbium coelestium (On the Revolutions of the Heavenly Spheres)

Nuremburg, Holy Roman Empire

1543
\end{thebibliography}

This references section (seen below) will have a single entry, denoted by the above information, with the tag “Copernicus” as designated by the \bibitem command. We can now create an in-text citation by typing the line:

The sun is the center of the solar system, and all planets revolve around it. \cite{Copernicus}

Using this, we obtain:
The sun is the center of the solar system, and all planets revolve around it. [1]

NOTE: Like the table of contents, producing the bibliography requires you to compile the document twice.

References

[1] Nicholas Copernicus

De revolutionibus orbium coelestium (On the Revolutions of the Heavenly Spheres)

Nuremburg, Holy Roman Empire

1543

9 \LaTeX{} Code for This Document

To serve as an additional example as well as for reference, attached is the \LaTeX{} code that was used to generate the manual above. The code is in a .tex file and can be opened with any standard text editor, such as Notepad++, TextEdit, or gedit.
10 Appendices

Appendix 1: Putting LaTeX on your personal computer

If you would prefer to put LaTeX on your own computer instead of having to use a TCC machine, you will need to install what’s known as a TeX distribution. It’s a big package that includes the compilers (to actually create the documents) and a library of standard typesetting commands.

Windows:
Visit
http://www.howtotex.com/howto/installing-latex-on-windows/
for a simple tutorial on how to install the TCC’s LaTeX editor on your PC.

Mac OS:
Visit
http://www.howtotex.com/howto/installing-latex-on-mac-os-x/
for a simple tutorial on how to install something similar to the TCC’s LaTeX editor on your Mac.

Linux:
If you’re brave enough to be a Linux user, you are hopefully familiar with your distro’s program installation methods. A list of standard terminal commands can be found at
http://latex-community.org/know-how/466-texlive-linux

Appendix 2: Using LaTeX in Linux/UNIX with Terminal

If you are accustomed to quicker work involving fewer mouse clicks in a Unix environment, you may want to avoid the clunkiness of a GUI for your document creation. The alternative method is to open a terminal and move into whatever directory you are going to have your .tex file in. Open up your favorite text editor (TCC machines come with the standard gedit program; the author recommends Sublime Text) and begin composing your document. When you are ready to compile, in the terminal enter the command

```
pdflatex filename.tex
```
replacing filename with the name of your file. If you are working through this tutorial and would like to use this method of document composition, just enter in the above command any place the instructions say “compile.”