LaTeX 101*

John Gardner and Alex Yuffa

May 21, 2008

Contents

*This tutorial is a modified version of "LaTeX: from beginner to TeXpert" written by John Gardner and is available online at http://generaldisarray.wordpress.com

1 Introduction

This document introduces the LaTeX typesetting system. After digesting the information below, you'll be able to:

- Download and install LaTeX on your PC or Mac
- Create basic documents using LaTeX
- Install new LaTeX packages
- Insert tables and figures into a LaTeX document
- Use LaTeX's cross-referencing, footnote and basic bibliography features
- Insert equations into a LaTeX document

These topics cover the majority of tasks that most people need to do when writing a document. However, please note that while the LaTeX system makes it very easy to create professional-looking documents, it is both comprehensive and extensible. There are many topics that are not covered by this basic tutorial. Fortunately, LaTeX is very well documented. If you come across something that you can't figure out how to do, ask your old friend Google for help.

2 What is LaTeX?

At its core, LaTeX is a typesetting system that allows authors to create highly polished documents without having to worry about formatting, page breaks, object positioning, or any other style concerns that distract them from focusing on writing. LaTeX is pronounced "lay-tech," as it is an extension of TeX ("tech"), the original typesetting system. You can read all about the history of TeX and LaTeX on Wikipedia. LaTeX is used widely in a variety of professions. Mathematicians, physicists, economists, statisticians and other academics and professionals that regularly use mathematical notation in their documents often use LaTeX because of the ease with which it handles such notation. Many publishers use TeX-based systems for typesetting documents.

3 How does LaTeX work?

LaTeX differs from traditional word processors in two fundamental ways:

1. Generally, LaTeX documents are written using the easy-to-learn LaTeX markup language, rather than by using a graphical interface to apply styles[1]
2. LaTeX processes your document after you have entered your text. So unlike word processors, it can use information about the total length of your document, number of tables, etc. to find the optimal places for tables, figures, page breaks, etc. to format your text

The following is an example of a very basic LaTeX document.

Input

```
\documentclass{article}
\title{Test Document}
\author{Your Name}
\begin{document}
\maketitle
This is a test document
\end{document}
```

Output

Test Document

Your Name

April 21, 2006

This is a test document.

With any LaTeX distribution, saving the above Input text as a `.tex` file and running `pdflatex` on that file will produce the above Output. LaTeX is designed to create the same output on any system. As a result, if you distributed the above text to anyone with a working LaTeX distribution, regardless of their particular system, they would get the exact same result. LaTeX outputs compiled documents in several formats, but the most popular is PDF.

[1] Graphical editors, such as Scientific Word (a commercial application) and LyX (an open-source application), are available; these applications are easier to use if you know how LaTeX works, so it's a good idea to learn it even if you don't plan to write LaTeX markup by hand.

3.1 Exercise

Repeat the above example using your real name. Which date appears on your LaTeX document?

Save your TeX file as `username1.tex`, where `username` is your username.

4 Getting LaTeX

All you technically need to create LaTeX documents is a LaTeX engine – the binary files and libraries that will convert plain text TeX files to polished PDF files. LaTeX can be run from the command line, so *nix and DOS aficionados will feel right at home. However, using a front-end for LaTeX can make things much easier. Most front-ends are essentially text editors with functions to

- Compile documents with LaTeX without using the command line
- Facilitate writing in the LaTeX language (wizards for table creation, code completion, syntax highlighting, etc.)

There are many engines and front-ends to choose from on every operating system. LaTeX tools have different configuration requirements and operating instructions on different operating systems, but almost every working environment involves

1. editing raw `.tex` files using a front-end
2. compiling the LaTeX document to a `.pdf`, generally using buttons or menu commands in the front-end rather than the command line

4.1 On Windows

Engine: MikTeX is a popular open-source distribution. To install, visit www.miktex.org, download the executable, and follow the dialog. Additional installation instructions are on the download page.

Front-end: TeXnic Center, available from toolscenter.org, is an open-source front-end with many helpful features. Installation is standard, just download and open the executable, which opens a wizard. TeXnic center is automatically configured to work with MikTeX. To test out your setup, save the sample document above as a `.tex` file using TeXnic Center and select Build ⇒ Current file. If everything is set up properly, a new PDF file (along with a log file) will be created in the directory where your document is saved.

4.2 On Mac OS X

Engine: gwTeX is a free and open-source LaTeX distribution for OS X that comes with a graphical installer. To install, download the i-Installer application, select a mirror, then select the TeX package. Additional installation instructions are available ii2.sourceforge.net/tex-index.html. Once installation is complete, all you need is a front-end.

Front-end: TeXShop (www.uoregon.edu/~koch/texshop/) is a very popular LaTeX front-end for OS X. Installation requires a simple drag and drop to the ~/Applications folder. TeXShop is automatically configured to work with gwTeX, so if that's the engine that you're using, you're set. To test out your distribution, try saving the sample document above as a TEX file and running LaTeX on your document by pressing command-t. If everything is configured properly, a window will appear similar to the example output above, and a new PDF file (as well as a log file) will appear in the directory where your file is saved.

4.3 On Linux:

Different Linux systems have their own application management utilities (apt-get or rpm, for example), and installation will depend on your particular Linux distribution. Ubuntu users can use the Adept Package Manager. Kile is a popular and easy-to-use front-end that works with both KDE and Gnome.

5 LaTeX basics

5.1 LaTeX commands

LaTeX commands generally begin with a backslash and take the form:

```
\command[options]{argument}
```

For example,

```
\section{Introduction}
```

would define a new section, named "Introduction." The "%" character defines a comment, and everything from that character to the end of the line is commented out and will be ignored by LaTeX. For example,

```
% This text is ignored by \LaTeX{}
```

To insert the "%" character into a document, escape it with a backslash: `\%`. Other single characters that require \ are

```
# $ & ~ _ ^ { }
```

To insert a backslash , "\", use `$\backslash$`. Quotes work a bit differently in LaTeX. To insert quote marks, use the form ``` ``text'' ```. That is, the ` character (top left of the keyboard) twice, followed by the single quote character, ', twice. Here is an example using escaped characters and quotes.

Input	Output
`Raising the number, % Ignored ABC's` `\# \^ { } \{ ``3'' \}`	Raising the number, # ^ { "3" }

5.1.1 Exercise

Append to your `username1.tex` file a new section which contains `\# \$` characters and a properly quoted word. For example,

Output

@$#*%!

I'm "bored."

Save your TEX file as `username1.tex`, where `username` is your username.

5.2 The preamble

Everything before the line `\begin{document}` is part of the preamble. A typical preamble might look like this:

```
\documentclass{article}
\usepackage{graphicx}
\usepackage{amsmath,amssymb}
\title{Test}
\author{Test}
\date{\today}
```

In the example above:

- `\documentclass{article}` tells LATEX that the document is an article. Other classes include report, book, letter and slides
- `\usepackage{graphicx}` tells LATEX to use the graphicx package, which allows users to include many types of graphics in their documents. Packages are covered later on. The `\usepackage{amsmath,amsfonts}` command invokes packages from the American Mathematical Society that extent the functionality of LATEX
- `\title{}` and `\author{}` obviously define the title and author
- `\date{\today}` tells LATEX to use today's date. `\date{April 2006}` would print "April 2006" as the date. The `\date{}` without an argument would cause LATEX to leave the date blank.

The `\documentclass{}` command has options. For example,

```
\documentclass[11pt,twocolumn]{article}
```

would organize body of the document into two columns. Note that options are separated by a comma. Other options include:

- `oneside` or `twoside`: change margins for a one or two-sided document
- `landscape`: change the document from portrait to landscape
- `titlepage` or `notitlepage`: define whether there is a separate title page, or if the title, author and date info are presented at the top of the article

We will use the following preamble in during Field Session:

Input

```
\documentclass[12pt,letterpaper]{article}
% letterpaper tells LaTeX to use 8.5 x 11 inches paper size
% 12pt tells LaTeX to use 12 point font
%
\usepackage[margin=1in]{geometry} % Set all margins to 1 inch
\usepackage[tight,nice]{units}
\usepackage{graphicx,color,float,amsmath,amssymb}
% float package is included so we can place figures/tables
% exactly where we want them via capital H flag.
%
\title{Put Your Title Here}
\author{Put Your Name Here}
\date{\today}
```

5.2.1 Excersise

What does `units` package do? How about the `color` package? Your answers should be typeset with above preamble and include at least one example on how `units, color` packages are used. For example:

Output

Packages

The `units` packages does blah, blah. We can use it via `command` to produce 13.6 eV or 123 m/s. The `color` packages does blah, blah. We can use it via `command` to produce text.

Save your TEX file as `username2.tex`, where `username` is your username.

5.3 The document body

Everything after the preamble and between `\begin{document}` and `\end{document}` is part of the document body. Most of a LATEX document is simply plain text. To start a new paragraph, insert two carriage returns (one blank line). To force a line break, use `\\`.

5.4 Document structure

A document's structure is defined using `\section{}` commands. LaTeX is strongly based on well-structured documents. The structure tags include:

- `\section{Name}`
- `\subsection{Name}`
- `\subsubsection{Name}`
- `\paragraph{Name}`

To insert an unnumbered section, use the command `\section*{Name}`. The section numbering will continue as normal with the next section, subsection, etc. The `\paragraph{}` command doesn't need to be included unless you want to insert a heading for a paragraph. For example,

Input

```
\section{Section command}
\section*{Section star command}
This section is not numbered.
\section{Section command}
Text here.  The numbering continues normally.
\subsection{Subsection command}
Text here
\subsubsection{Subsubsection}
\paragraph{Paragraph command} This paragraph has a title.
```

Output

1 Section command

Section star command

This section is not numbered.

2 Section command

Text here. The numbering continues normally.

2.1 Subsection command

Text here.

2.1.1 Subsubsection

Paragraph command This paragraph has a title.

6 Environments

Environments are special blocks of text. For example, the `itemize` and `enumerate` environments create bulleted and numbered lists, respectively. Here is an example of `itemize` environment:

Input

```
\begin{itemize}
\item{First thing}
\item{Second thing}
\item{Third thing}
\end{itemize}
```

Output

- First thing
- Second thing
- Third thing

Here is an example of `enumerate` environment:

Input

```
\begin{enumerate}
\item{First numbered thing}
\item{Second numbered thing}
\item{Third numbered thing}
\end{enumerate}
```

Output

1. First numbered thing
2. Second numbered thing
3. Third numbered thing

Note that environments always begin with `\begin{name}` and end with `\end{name}`. They can be nested, so one item of a bulleted list might contain another bulleted list, or a numbered list, etc. For example:

Input

```
\begin{itemize}
\begin{itemize}
\item{Main list: item one}
  \begin{enumerate}
  \item{Numbered subitem one}
    \begin{itemize}
    \item{Deep}
      \begin{itemize}
      \item{Very Deep}
      \end{itemize}
    \end{itemize}
  \item{Numbered subitem two}
  \end{enumerate}
\item{Mail list: item two}
\end{itemize}
```

Output

- Main list: item one
  1. Numbered subitem one
     - Deep
       - Very Deep
  2. Numbered subitem two
- Mail list: item two

6.1 Exercise

Append appropriate LaTeX code to your `username2.tex` file to produce:

Output

- You
  - are
    - number
      1. Put your name here

Save your TeX file as `username2.tex`, where `username` is your username.

7 Modifying text styles

The basic idea behind LaTeX is to absolve the author of formatting duties. Nevertheless, it's still occasionally necessary to manually format certain text styles. For example:

Input

```
\textbf{bold text} not bold text\\
\textit{italic text} not italic text\\
\texttt{typewriter text} not typewriter text
```

Output

bold text not bold text
italic text not italic text
`typewriter text` not typewriter text

7.1 Exercise

Append appropriate LaTeX code to your `username2.tex` file to produce the following "Woodisms" (extra credit for those who know what it is):

Output

"Where's the other *half* of the **damn class**, by the way? You're not `bailing in real time`, are you?"

Save your TeX file as `username2.tex`, where `username` is your username.

8 The graphicx package

The `graphicx` package allows you to insert images into a LaTeX document. To use it, the command `\usepackage{graphicx}` must be in your document preamble. Then, to insert a graphic, use the command:

```
\includegraphics[options]{filename.png}
```

The `pdflatex` with `graphicx` package supports `PDF`, `JPG`, and `PNG` graphics formats. The `options` include: `width=Xin`, `height=Xin`, and `scale=X`, where `X` denotes a number. For example, `\includegraphics[width=1.5in]{filename.pdf}` will produce a graphic that's 1.5 in wide.

8.1 Other packages

For just about every modification that you might want to make to a standard LaTeX document, there is a pre-made package to help you do so. To learn more about the packages described, or to download new packages, visit the Comprehensive TeX Archive Network (CTAN).

9 Figures and tables

Figures and tables are LaTeX environments, however they have special attributes, such as the `\caption{}` command, which gives them titles within the document. They are called float elements, because their position in the final compiled document depends on LaTeX's style algorithm.

9.1 Figures

To insert a figure, use

Input

```
\begin{figure}[hbtp]
\begin{center}
\includegraphics[width=Xin]{filename.pdf}
\end{center}
\caption{Description of the figure. \label{your-reference-key}}
\end{figure}
```

In the above markup,

- `\begin{figure}` simply tells LaTeX that there is a figure environment
- `[hbtp]` determines how LaTeX will place the figure (here (h), bottom (b), top(t), page(p)). LaTeX will first attempt to insert the figure at its insertion point in the TeX file. If this is not possible due to space or other aesthetic considerations, it will

try to place it at the bottom of the page, then at the top of the page, then on a special page reserved just for float elements. The order in which h, b, t and p are specified determines where LaTeX tries to place the float first. To force the graphic to appear in its original place, for example, you could put `\begin{figure}[h]`, omitting b, p and t

  - Sometimes even `\begin{figure}[h]` won't force the graphic to appear in its original place and in this case we must use the `float` package[2]. To 100% force the graphic to appear in its original place overriding all space and aesthetic considerations use capital "H" flag, i.e., `\begin{figure}[H]`

- `\begin{center}` simply tells LaTeX to center the figure on the page. Don't forget to end the centering environment before you end the figure environment
- `\includegraphics[...]{...}` specifies the location of the file that is being inserted as a figure
- `\caption{Description of the figure.}` specifies the name of the figure
- `\label{your-reference-key}` is a label that you can use to refer to the figure in the text. For example, if you label your figure "fig1" then you can reference it later on by typing `\ref{fig1}`

Here is an example of "H" flag at work:

Input

```
A blue wire carrying current $I=I_o t^3/3$ is wound evenly on a torus of%
rectangular cross section.  There are $N$ turns of the blue wire in all.%
A red wire is thrown over the torus and is connected to a resistor, $R$,%
see Fig.~\ref{torus}.
%
\begin{figure}[H]
\begin{center}
\includegraphics[width=4cm]{torus.pdf}
\end{center}
\caption{A blue wire carrying current $I=I_o t^3$ is wound evenly on%
a torus of rectangular cross section, with inner radius $r_1$ and%
outer radius $r_2$.  There are $N$ turns of the blue wire in all.%
A red wire is thrown over the torus and is connected%
to a resistor, $R$.  \label{torus}}
\end{figure}
```

[2] Old LaTeX systems use the `here` package instead of the `float` package.

Output

A blue wire carrying current $I = I_o t^3/3$ is wound evenly on a torus of rectangular cross section. There are N turns of the blue wire in all. A red wire is thrown over the torus and is connected to a resistor, R, see Fig. 1.

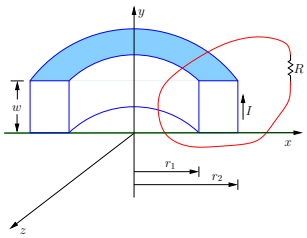

Figure 1: A blue wire carrying current $I = I_o t^3$ is wound evenly on a torus of rectangular cross section, with inner radius r_1 and outer radius r_2. There are N turns of the blue wire in all. A red wire is thrown over the torus and is connected to a resistor, R.

9.1.1 Exercise

Answer the following questions.

1. What is the filename and format of the image in Figure 1?
2. Where is the Figure 1 placed?
3. How large is the image in Figure 1?
4. How can we refer to the Figure 1 via `\ref{...}` command?

Your answers should be typeset in LaTeX using `enumerate` environment.

Save your TeX file as `username2.tex`, where `username` is your username.

9.1.2 Exercise

Reproduce the following output. Your may download the image here. Hint: The width of the image in Figure 2 is 1 in.

Output

Typical geometry for Ampere's Law type problem.

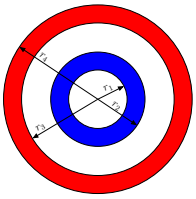

Figure 2: Ampere's Law

Use Ampere's Law to find current in Figure 2.

Save your TEX file as `username2.tex`, where `username` is your username.

9.2 Tables

A floated table in LATEX consists of two environments: `table`, the actual floated entity in the text, and `tabular`, the data contained in the table. For example:

Input

```
\begin{table}[H]
\caption{This table is an example. \label{exampleTable}}
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
row 1, column 1 & row 1, column 2 &  row 1, column 3 \\ \hline
row 2, column 1 & row 2, column 2 &  row 2, column  3 \\
row 3, column 1 & row 3, column 2 &  row 3, column 3 \\ \hline
\multicolumn{2}{|c|}{row 4, two columns} & row 4, column 4 \\ \hline
\end{tabular}
\end{center}
\end{table}
```

Output

Table 1: This table is an example.

row 1, column 1	row 1, column 2	row 1, column 3
row 2, column 1	row 2, column 2	row 2, column 3
row 3, column 1	row 3, column 2	row 3, column 3
row 4, two columns		row 4, column 4

Everything except the code between `\begin{tabular} ... \end{tabular}` is the same as the `figure` environment described Section 9.1. Here's how the `tabular` environment works:

- `\begin{tabular}{|c|c|c|}` tells LaTeX to start a new tabular environment with three centered columns. The bar "|" before/after the "c", tells LaTeX that there is a vertical border before/after the column. Using `{lcrr}` would create four columns, the first left aligned, the second centered, and the third and fourth right aligned
- Table cells are separated by "&" and table rows are separated by "\\"
- `\hline` creates a horizontal line
- `\multicolumn{2}{|c|}{Text here}` creates a row that spans all two columns, is centered, and contains the text "Text here"

There are more complicated options for creating and inserting tables, but the rules above cover the commands needed to create most basic to intermediate tables.[3]

9.2.1 Exercise

Reproduce the Table 2, labeling it as `\label{myTable}`.

Output

Table 2: My very own table labeled as Table 2.

row 1, column 1	row 1, column 2	row 1, column 3	row 1, column 4
row 2, column 1	row 2, two columns		row 2, column 4

Save your TeX file as `username2.tex`, where `username` is your username.

[3] OpenOffice users can use Calc2LaTeX to convert between Calc spreadsheets and LaTeX tables. MS Office users can try Excel2LaTeX, which does the same thing using Excel spreadsheets. Both utilities are cross-platform.

10 Annotations

LaTeX is capable of automatically creating important annotations, such as footnotes, cross references, tables of contents and bibliographies. Note that, since the following commands require LaTeX to automatically number text elements, LaTeX must be run on your document at least twice for proper display.

10.1 Footnotes and Endnotes

To insert a footnote, simply type `\footnote{text here}`. LaTeX will automatically insert the footnote number and text.[4]

10.2 Cross references

To reference a labeled Table or Figure, use `\ref{your-reference-key}` where `your-reference-key` is the argument to the `\label{your-reference-key}` command in the `table` or `figure` environments.

10.3 Table of contents

To insert a table of contents, simply put `\tableofcontents` at the beginning of your document. To insert a list of of figures, simply put `\listoffigures` at the beginning of your document. To insert a list of tables, simply put `\listoftables` at the beginning of your document. For example:

Input

```
\documentclasss[12pt,letterpaper]{article}
% Preamble
\begin{document}
\tableofcontents
\listoffigures
\listoftables
% Different sections, text, etc.
\end{document}
```

10.4 Bibliography

To create a bibliography, insert a list of the citations at the end of your document, using the form:

[4] My footnote.

Input

```
\begin{thebibliography}{99}
\bibitem{key1}H.B.~Phillips, \textit{Vector Analysis} (Wiley and%
 Sons, 1933), p. 206.
%
\bibitem{key2}P.M.~Morse and P.J. Rubenstein, Phys. Rev.%
\textbf{54}, 895 (1938).
%
\bibitem{key3}J.A.~Stratton and L.J.~Chu, Phys. Rev. \textbf{56},%
99 (1939).
\end{thebibliography}
```

Output

References

[1] H.B. Phillips, *Vector Analysis* (Wiley and Sons, 1933), p. 206.

[2] P.M. Morse and P.J. Rubenstein, Phys. Rev. **54**, 895 (1938).

[3] J.A. Stratton and L.J. Chu, Phys. Rev. **56**, 99 (1939).

You must manually type the bibliography entries. To refer to an item within the text, use `\cite{key}`[1]. The `{99}` tells LaTeX that there a maximum of 99 entries in the bibliography. LaTeX needs to know this so it can correctly justify the bibliography entries with their numbering on the left. A more efficient way to create bibliographies is to use BibTeX, which allows you to maintain a database of citations and call them as needed in your bibliography. There are also graphical tools for managing your reference databases, so you don't have to hard code the citations, and can easily change them to different formats. However, BibTeX is too complicated to explain in this document. For an introduction, see this page.

10.4.1 Exercise

Reproduce the above example and use `\cite{...}` command somewhere inside your LaTeX document.

Save your TeX file as `username2.tex`, where `username` is your username.

11 Inserting mathematics

There are several ways to include mathematical notation in LaTeX documents. The most common are inline notation and the displaymath environment.

11.1 Inline

To include some mathematical notation within a paragraph, without offsetting from the rest of the text, enclose the notation between dollar signs. For example, `$a^2+b^2=c^2$` will produce $a^2 + b^2 = c^2$, which is the Pythagorean theorem.

11.2 Display math

The displaymath environment lets you offset some mathematical notation from the rest of the document. For example:

Input

```
Notice how the equation is offset,
\[
a^2+b^2=c^2
\]
but we don't have an equation number.
```

Output

Notice how the equation is offset,

$$a^2 + b^2 = c^2$$

but we don't have an equation number.

11.3 Equation

The equation environment can be used to place numbered equations in the text. For example:

Input

```
An offset equation
\begin{equation}
a^2+b^2=c^2
\label{pythag}
\end{equation}
with equation number and label.
```

Output

An offset equation

$$a^2 + b^2 = c^2 \tag{1}$$

with equation number and label.

In the example above, we can refer to the equation via `\eqref{pythag}` to produce (1), e.g.

Input

```
Pythagorean theorem is given %
by eqn.~\eqref{pythag}.
```

Output

Pythagorean theorem is given by eqn. (1).

11.4 Exercise

Using the Field Session preamble, see Section 5.2, type set the following:

Output

Generic relativistic energy-momentum relationship is given by

$$E^2 = \left(mc^2\right)^2 + \left(pc\right)^2, \tag{2}$$

where p is the momentum. If $p = 0$ then eqn. (2) reduces to

$$E = mc^2,$$

where we have taken the positive square root.

Hint: Use `\left(` and `\right)` for ().

Save your TEX file as `username3.tex`, where `username` is your username.

11.5 Exercise

Refer to your **labeled** equation in Exercise 11.4 via `\ref{}` command and via `\eqref{}`; do you see any differences in the output of the two commands?

Save your TEX file as `username3.tex`, where `username` is your username.