

# L<sup>A</sup>T<sub>E</sub>X Tutorial

Dailon Dolojan  
ddolojan@ucsc.edu

January 27, 2018

## 1 Introduction

L<sup>A</sup>T<sub>E</sub>X or more formally known as Lamport TeX is a document preparation system which allows users to write word documents in plain text.

L<sup>A</sup>T<sub>E</sub>X implements the usage of markup tagging in order to create the overall structure of a given document. L<sup>A</sup>T<sub>E</sub>X is often used to create documents such as letters, articles, books, and other stylized documents. Within this manual, common conventions of L<sup>A</sup>T<sub>E</sub>X and their usage will be explained in detail.

## 2 Document classes

When writing a document within L<sup>A</sup>T<sub>E</sub>X there are various document classes which determines the overall layout of the document. There are several document classes within L<sup>A</sup>T<sub>E</sub>X including *article*, *letter*, *beamer*, *slides*, *books*, and other user defined document classes.

In defining a document class of a specific document, a user can utilize the following structure for establishing a document class.

### Example 2.0.1:

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

The user can specify various attributes of the document such as size of font, paper size, and other aspects within the *options* parameter. The *class* parameter is used to specify the class of a document. Within **Example 2.0.2**, a user has specified to used 11pt font and will have a paper size of a4 paper with a document class of “letter”.

### Example 2.0.2:

```
\documentclass[11pt,a4paper]{letter}
```

Within this manual we will go over three prominent classes: *article*, *letter*, and *beamer*.

### 2.1 Article

The document class *article* is often used to write articles within scientific journals, reports, presentations, and other applied uses. Article is often used to present information within a structured report and is one of the most common document classes used within L<sup>A</sup>T<sub>E</sub>X. **Example 2.1** specifies how to write an article with 12pt font.

### Example 2.1:

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

## 2.2 Letter

The document class *letter* is used to write a structured, formal letter to a specific correspondent. The document class *letter* has specific commands that can be utilized to create the overall layout of a stereotypical letter. **Example 2.2** implements a typical structure of a formal letter.

### Example 2.2:

```
\address{Street \ City \ Country}
\begin{document}
\begin{letter}{Recipient \ Street\
City\ Country}
\opening{Dear _}
Insert Body of Letter here
\closing{Sincerely,}
\signature{Your Name}
\ps{Insert P.S.}
\end{letter}
```

The command `\address{}` inputs the sender's street, city and country on the top right hand corner of the document which is separated by double backslashes. The command `\begin{letter}{}` specifies the recipient's address within the second parameter separated by backslashes. The command `\opening{}` allows for the user to insert the opening address to the recipient such as "Dear Mr. Greenwidge" or "Hello Ms. Smith" whereupon the user may write the body of the letter following the opening. Lastly the commands `\closing{}`, `\signature{}`, and `\ps{}` implements the closing of the letter within the given parameter, types your name, and writes any post scriptum (p.s.) notes at the end of your letter with each respective command.

## 2.3 Beamer

*Beamer* is a document class within  $\text{\LaTeX}$  that allows a user to create presentations with various slides. Individuals often use the document class *Beamer* to further customize a slide show to highlight key points during a presentation. **Example 2.3** illustrates how a user would be able to create a slide with a title and subtitle.

### Example 2.3:

```
\frametitle{Insert Title of Frame}
\framesubtitle{Insert Subtitle of
Frame}
Insert Content of slide
\end{frame}
```

## 3 Creating a Title and Author

To create a simple title and author listing on a given document, a  $\text{\LaTeX}$  user can simply insert a title and author by first specifying the title and author with the following commands `\title{}` and `\author{}`. The user can then create the title and author with the command `\maketitle{}`. **Example 3.0.1** illustrates the creation of a simple centered titled and author.

### Example 3.0.1:

```
\title{Insert Title of Document}
\author{Insert Author's Name}
\maketitle{}
```

In order to implement more complex functionality of creating titles, a user can specify a subtitle using the command `\subtitle{}` where the user can insert the subtitle within

the curly brackets. An individual may even change the overall format of creating a title and author by utilizing commands that change the alignment of text such as `\centering{}`, `\raggedleft{}`, and `\raggedright{}` which would be inserted into the curly bracket of `\title{}` or `\author{}` as seen in **Example 3.0.2**.

**Example 3.0.2:**

```
\title{\raggedright Insert Title of
      Document}
\author{\raggedright Insert Author's
      Name}
\maketitle{}
```

An individual could also create a title page as well by using the command `\pagebreak{}` after implementing the command `\maketitle{}` or even utilizing the environment `\begin{titlepage}`.

## 4 Sections and Labeling

A user can create various sections, sub-sections, sub-subsections, and paragraphs to create a more structured document through built in  $\text{\LaTeX}$  commands as well.

### 4.1 Sections

In order to create a section a user can implement the command `\section{}` where the user would insert the name of the section within the curly brackets.

### 4.2 Subsections

A subsection can be created using the command `\subsection{}` where the name of a

given subsection would be typed within the curly brackets.

### 4.3 Sub-subsections

Sub-subsections are used to categorize a category within a given section and can be implemented using the command `\subsubsection{}` with the name of the subsection listed within the curly bracket.

### 4.4 Paragraphs

A user may create separate paragraphs utilizing the command `\par{}` before a block of text. A user can even specify if the paragraph should start with an indentation by utilizing the command `\setlength{\parindent}{}` where in the empty curly brackets a user can specify the size of the indent such as “1cm” or “0.5in”.

## 5 Environments

Environments within  $\text{\LaTeX}$  are used to create the formatting of text blocks within a  $\text{\LaTeX}$  document. Environments can range from alignment, numbering, creating matrices, or other user defined formatting. The syntax of an environment starts with an opening tag `\begin{}` and a closing tag `\end{}`.

### 5.1 Alignment

A user can utilize environments to format the alignment of a given text. An individual can choose alignments such as left justified, right justified, or center for a given block. **Example 5.1** showcases how to implement the respective alignments.

### Example 5.1

```
\begin{\raggedright}
  Insert right justified text
\end{\raggedright}

\begin{\raggedleft}
  Insert left justified text
\end{\raggedleft}

\begin{\center}
  Insert centered text
\end{\center}
```

## 5.2 Tabular

The *Tabular* environment is used to format a text block into a matrix. The *Tabular* environment uses a parameter to specify the number of items within a column where the text between the opening and closing tags specifies the entries of a column separated by & and \\ separates rows. **Example 5.2** illustrates this below.

### Example 5.2

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

## 5.3 Itemize

The *Itemize* environment is an environment used to create lists within L<sup>A</sup>T<sub>E</sub>X. A user can create a bullet point list within a document as seen in **Example 5.3**.

### Example 5.3

```
\begin{itemize}
\item Item 1
\item Item 2
\item Item 3
\end{itemize}
```

## 6 Mathematical Formulas

To create Mathematical Formulas within L<sup>A</sup>T<sub>E</sub>X a plethora of tools can be implemented to create complex formulas.

### 6.1 Exponents and Subscripts

To create an exponent of a number simply type in the base number followed by the caret symbol and the exponent number. In addition to create a subscript simply type the base number followed by the underscore symbol and the number to be sub-scripted in order to produce a subscript.

### 6.2 Symbols

A user can invoke various math symbols utilizing L<sup>A</sup>T<sub>E</sub>X code. Some basic math symbols include implementing a fraction using `\frac{ }{ }` where the first parameter would be the numerator and the second parameter would be the denominator. The summation symbol can be created using the command `\sum{ }`. Greek letters such as alpha, beta, gamma, rho, sigma, delta, and epsilon can be inserted by the following commands respectively `\alpha`, `\beta`, `\gamma`, `\rho`, `\sigma`, `\delta`, and `\epsilon`. For more math symbols, please refer to the L<sup>A</sup>T<sub>E</sub>X documentation.

## 6.3 Math Modes

L<sup>A</sup>T<sub>E</sub>X has a wide range of math modes to display equations in a multitude of ways.

### 6.3.1 Inline Math Mode

For an equation to be inserted within a line of text, a user can simply type in an equation between two dollar signs as shown in **Example 6.3.1**.

#### Example 6.3.1

text  $F=n^2$  text

### 6.3.2 Displayed Math Mode

To create a separate line with the equation centered a user can simply input the equation in between double dollar signs as illustrated below in **Example 6.3.2.1**.

#### Example 6.3.2.1

text 
$$F=n^2$$
 text

If a user wanted to have the equation numbered in order to reference the equation throughout the paper, the user simply needs to utilize the *equation* environment as shown in **Example 6.3.2.2**.

#### Example 6.3.2.2

```
\begin{equation}
F=n^2
\end{equation}
```

## 7 User-defined Marcos

A user can create new commands to further specialize the formatting of a given text by implementing a new command. The format of a macro can be seen in **Example 7.0.1** where a user specifies the name of the macro, the number of parameters, and the definition of the command.

#### Example 7.0.1

```
\newcommand{\macroname}[number of
parameters]{command definition}
```

The following examples illustrate the usage of creating user-defined macro commands where **Example 7.0.2** displays three unique user defined macros. The first command displays the Pythagorean theorem, the second displays the name of a textbook with the title italicized, and the third creates the “ $\times 10$ ” which is used in scientific notation.

#### Example 7.0.2

```
\newcommand{\pythag}{a^2+b^2=c^2}

\newcommand{\textBook}{\textit{
Analysis of Algorithms} by
Henry Peters}

\newcommand{\scin}{\${\times 10$}}
```

## 8 Bibliography and Citation

A L<sup>A</sup>T<sub>E</sub>X user can create a bibliography to host all their citations by creating a “*bib*” file that will contain all the bibliographic information of a given L<sup>A</sup>T<sub>E</sub>X file. Once a “*bib*”

file is created, a user can easily reference the citation using the `\cite{}` command.

The “.bib” file will contain all the bibliographic information necessary which the given L<sup>A</sup>T<sub>E</sub>X document will reference when using the `\cite` command. The format of creating a reference within a “.bib” file can be seen in **Example 8.0**.

### Example 8.0

```
@TypeOfDocument{citation_name,
AUTHOR="name of author",
TITLE="title of book",
PUBLISHER="name of publisher",
YEAR="Year Of Publishing",
}
```

Once a reference is created within a “.bib” file, a user can simply type in the command `\cite{citation_name}` and the text will note a number to reference that specific text.

## 9 Interesting L<sup>A</sup>T<sub>E</sub>X Feature: Creating Images

To import images within L<sup>A</sup>T<sub>E</sub>X a user simply needs to use the *graphicx* package within L<sup>A</sup>T<sub>E</sub>X by invoking the command `\usepackage{graphicx}`. Once the package is implemented, a user needs to write a command that tells L<sup>A</sup>T<sub>E</sub>X where the images are kept within the current directory using the command `\graphicspath{}`. The user can then insert the image into the document by the command `\includegraphics{}`. An example of this can be seen in **Example 9.0.1**.

### Example 9.0.1

```
\usepackage{graphicx}
\graphicspath{imagepath/}
\includegraphics{ImageName}
```

A user can then specify the image location by implementing the *figure* environment where a user can specify the location and image size. Utilizing the command `\begin{figure}[h]`, a user can specify where exactly to place the figure within the brackets. Within Example 9.0.1, the figure is placed within the line of text with the *h* parameter, the image will be centered, and have a width and height of 0.5 with both a label and caption.

### Example 9.0.1

```
\begin{figure}[h]
\centering
\includegraphics[width=0.5, height=0.5]
{ImageName}
\caption{Insert Caption here}
\label{Insert Label Here}
\end{figure}
```

## References

- [1] LaTeX Documentation, *LaTeX Documentation*, <https://www.sharelatex.com/learn>. Feb 1. 2018.
- [2] Phokion G. Kolaitis, *Lecture 2*, “CMPS 185: Technical Writing for Computer Science”, <https://piazza.com/cmeps185> Feb 1. 2018.