

# Basic L<sup>A</sup>T<sub>E</sub>X: Features Explained

James M. Vriónis  
CMPS-185  
University of California, Santa Cruz

1-26-2018

[1]

<http://www.tug.org/texlive/>

## 1 Introduction

Pronounced either LAH-tek or LAY-tek, L<sup>A</sup>T<sub>E</sub>X is a *free* typesetting language developed by *Leslie Lamport* based on *Donald E. Knuth's* T<sub>E</sub>X.

Markup Language Key Features: [2]

Typesetting journal articles.

Technical reports.

Bibliographies and indexes.

and many more.

## 2 Install

### 2.0.1 Ubuntu Linux

Open Terminal to install **TEX LIVE** via the *apt-get* command:

```
sudo apt-get install texlive-full
```

*NOTE: it's always good to perform a "sudo apt-get upgrade && sudo apt-get update" before installing anything new.*

now ".tex files" can be compiled using the pdf<sub>l</sub>atex <your file>.tex as long as you in the same directory as the file.

Alternatively, you may install from the link below

### 2.0.2 MacOS

Open Terminal to install using *Homebrew Cast*.

```
macbk-usr$ brew tap caskroom/cask
macbk-usr$ brew cask install mactex
```

now ".tex files" can be compiled using the pdf<sub>l</sub>atex <your file>.tex as long as you in the same directory as the file.

Alternatively, you may install from the link below

<http://www.tug.org/mactex/>

### 2.0.3 Windows

To install and use a L<sup>A</sup>T<sub>E</sub>X distribution on Windows operating system one should install **T<sub>E</sub>X Live** via the installer.

Install via link below  
<http://www.tug.org/texlive/>

## 3 Compilation

All three Operating Systems have very similar terminal command line arguments necessary to compile you ".tex" file → the most commonly desired ".pdf".

In an open terminal simply type the following:

'pdflatex you\_file\_name.tex' and within the current working directory '.pdf' file will be created.

Linking '.bib' files take a little more work. I will explain the steps as follows for a file named blah.tex.

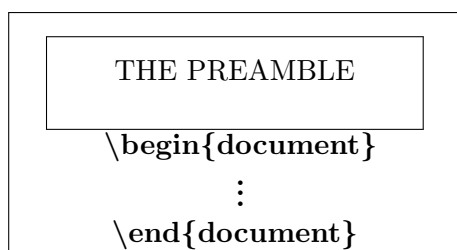
If you have not already done so, change the current directory of terminal to that which contains blah.tex as well as another file created by you called blah.bib. As long as there are not any errors within your latex code 4 commands make this possible:

```
pdflatex blah.tex
bibtex blah.aux
pdflatex blah.tex
pdflatex blah.tex
```

It is important to execute the pdflatex command twice after bibtex blah.aux has been run. This will create your desired pdf-file with an added references section (assuming you have entries in the .bib file). The last section of this document will contain more information about bibliographies.

## 4 Document Classes

The **Preamble** in a L<sup>A</sup>T<sub>E</sub>X document is defined as the area prior to any invocation of begin/end {document} tags. This space is used to define the type of document class you will be using, the page layout desired, and user defined commands.



This is where your *Document type* is to be defined.

### Types of Document Classes

There are nine main L<sup>A</sup>T<sub>E</sub>X defined textitDocument Classes. They are as follows:

[article](#), [IEEEtran](#), [proc](#), [report](#), [book](#), [slides](#), [memior](#), [letter](#), and [beamer](#).

In order to use these one of these *Document Classes* one must use the following command: `\documentclass{Document-Class}`.

where *Document-class* must be replaced with one of the nine blue labeled document classes (show above).

Each of these have their own unique advantages and one package may be already optimized for your particular use, but to use any of these environments they must be placed with-in the **preamble** of your document.

### 4.1 Article

- **Description:**

Quite possibly the most popular L<sup>A</sup>T<sub>E</sub>X class used is textbfarticle.

It is mainly used for short documents and journal articles it can also be used for the following: articles in scientific journals, presentations, short reports, program documentation, invitations, homework, note-taking, and many more.

- **Commands:**

```

\part{}
\chapter{}
\section{}
\paragraph{}
\subsection{}

```

[3]

- **How to use:**

- To use this particular document class within your L<sup>A</sup>T<sub>E</sub>X document, the `\documentclass{article}` in the area defined as the preamble.

[4]

## 4.2 IEEEtran

- **Description:**

- Institute of Electrical and Electronics Engineers (IEEE) transactions journals and conferences are what is meant by IEEEtran document class name. This document class is particularly useful for (IEEE) authors and has many useful commands and layout required by this organization.

- **Commands:**

```
\thanks{}  
\abstract{}  
\bibliography{}  
\Theorem{}  
\cite{}
```

- **How to use:**

- To use this particular document class within your  $\text{\LaTeX}$  document, the `\documentclass{IEEEtran}` in the area defined as the preamble.

## 4.3 book

- **Description of Book:**

- The structure of this document class contains "chapter" which a header automatically made with pages and the name of section on odd pages. Pages are always 2-sided by default.

- **Commands:**

```
\part{}  
\chapter{}  
\section{}  
\paragraph{} [3]  
\subsection{}
```

- **How to use:**

- To use this particular document class within your  $\text{\LaTeX}$  document, the `\documentclass{book}` in the area defined as the preamble.

## 5 Title, Listing Multiple Authors and Affiliations

Titles in  $\text{\LaTeX}$  can be made using `\maketitle` or by using the **titlepage** environment. As a little example, here is the code I wrote to make the title of this document:

```
\title{Basic\LaTeX{}: Features Explained }  
\author { James M. Vronis \\  
CMPS-185, \\  
University of California, Santa Cruz \\  
}  
\maketitle  
My title was set up completely using the  
maketitle as you can see it outputs a very  
professional title.
```

### 5.1 Title

Titles in  $\text{\LaTeX}$  can be made using `\maketitle` or by using the **titlepage** environment. As a little example, here is the code I wrote to make the title of this document:

```
\title{Basic \LaTeX{}: Features Explained }  
\author { James M. Vronis \\  
CMPS-185, \\  
University of California, Santa Cruz \\  
}  
\maketitle  
This code would be placed immediately after  
\begin{document}.
```

#### 5.1.1 Title Page

Another way to accomplish a **title** would be to make a **title page**. Another example should make the use of this clearer, so first we need to define attributes within the preamble portion of your document.

```
\title{Basic \LaTeX{}: Features Explained }  
\author{James M. Vronis Inspired by  
CMPS-185  
January 27, 2018  
}  
Next, immediately after our \begin{document}  
tag we can put the do the following:
```

```

\title{Basic \LaTeX{}:
  Features Explained}
\author{James M. Vrionis}
\thanks{Inspired by
  CMPS-185}}
\date{January 27, 2018}

```

```

\begin{document}
\begin{titlepage}
\maketitle
\end{titlepage}

```

All attributes of the **Title** should be defined within the preamble and like we have seen before

## 5.2 Multiple Authors and Affiliations

To list multiple Authors and Affiliations the `\author` command will suffice. Let me show you another nice example of how this can be accomplished:

```

\author{
  Author 1 \\\
  Company or School \\\
  ... etc. ... \\\
\and
  Author 2 \thanks{...} \\\
  ... etc. ... \\\
}
\date \\\
\maketitle

```

## 6 Sections

These are the  $\text{\LaTeX}$  commands very specific to the organization of document types. A section creates a numbered title above text its associated block of material. The numbering system is set by default but is easily turned off by placing an asterisk before the first opening curly brace.<sup>1</sup>

<sup>1</sup> Using the asterisk will remove this section from a Table of Contents and you must re-add, via `\addcontentsline` placed to the right of `\section{SoMe.SecTioN}`

## 6.1 Subsections

A `\{subsection\}` is a child of its parent section. If you are in section 4.0, the following subsections would be 4.1, 4.2,..., 4.x where x can be any number larger than its predecessor.

If you are in section 5.0, the following subsections would be 5.1, 5.2,...,5. x where x can be any number larger than its predecessor. An easy interpretation would be **Section.Subsection**

## 6.2 Sub-subsections

A `\{subsubsection\}` is the grandchild its some preceding parent section. 5.1.1 would read: section 5, subsection 1 of subsubsection 1. Possible sub-subsections are 5.1.1, 5.1.2, ...,5.1.x, with x to any number larger than its predecessor.

An easy interpretation would be **Section.Subsection.Sub-subsection**

---

By default, sections are numbered and in bold font. Say we are looking some text and we come across bolded text **5 Sections** above the text we are reading, which means section 5 of our text and anything below it, whether it be a subsection, subsubsection or paragraph should be related to this main idea.

```

Section 1
  Paragraphs
  Subsection 1
    Paragraphs
    Sub-subsection 1
      Paragraphs
      Sub-subsection 2
        Paragraphs
  Subsection 2
Section 2

```

and so on...

`\section{ Title of Section}`

## 6.3 Paragraphs

[3] When using the `\begin{paragraph}` command text is broken down into lines, and lines are broken down into pages. To end a paragraph there must be one or more blank lines following the text inside the paragraph. Your text should take place of the vertical dots.

```
\begin{paragraph}
      ⋮
\end{paragraph}
```

## 6.4 Labeling

[5] Labels in  $\text{\LaTeX}$  are references to previous place in your document enabling a much better way to point back to previous marked objects. Objects are marked with `\label{some\_name}` and can be retrieved using `\ref{some\_name}` command.

```
\{section}
\label{labbbel}
      ⋮
\ref{labbbel}
```

# 7 Environments

[1]  $\text{\LaTeX}$  comes equip with many different environments. Three environments we will look at are **Tabular**, **Itemize/Enumerate**, and **Description**. [1]

## 7.1 Tabular

**Tabular** is an alignment environment with vertical and horizontal line separating capabilities. When one or both of these capabilities are enabled many different arrangements become available with different column orientations, justifications, and unique line drawing instructions. [6]

**Syntax:**

```
\begin{tabular}[pos]{table spec}
\begin{tabular} {A}
& & \\
B & & \\
& & \\
\end{tabular}
```

Here we have a place to define columns.

A l for left-justified  
c for centered  
r for right-justified

B Here we have the number of entries in the table.

**No line Example:**

this is	table using	mbox
L-just col	center col	R-just col
using	tabular	envir

**Same example but with an added |:**

this is	table using	mbox
L-just col	center col	R-just col
using	tabular	envir

**Added another | to previous:**

this is	table using	mbox
L-just col	center col	R-just col
using	tabular	envir

**Adding a \hline to previous:**

this is	table using	mbox
L-just col	center col	R-just col
using	tabular	envir

**Added another \hline to previous:**

this is	table using	mbox
L-just col	center col	R-just col
using	tabular	envir

## 7.2 Itemize and Enumerate

**itemize** is unordered list making environment whereas the **Enumerate** environment is numerically ordered.

```
\begin{itemize}
\item "Text goes here"
:
\end{itemize}
```

```
\begin{enumerate}
\item "Text goes here"
:
\end{enumerate}
```

## 7.3 Description

**Description** is an ideal list making environment for definitions.

```
\begin{description}
\item[Word]
Here is the definition
\item[Word2]
Here is the definition 2
\end{description}
```

## 8 Mathematical Formulas

L<sup>A</sup>T<sub>E</sub>X has many different ways to write mathematical expressions. Using either **inline** or one of the many different *display* modes. Displays are separate from paragraphs and used as its own code block whereas inline expressions can be used with-in writing environments.

### Inline:

- $\text{math equation inline}$
- $\text{[ math equation inline ]}$
- $\text{( math equation inline )}$

### Display:

- $\text{\begin{equation} \end{equation}}$

- $\text{\begin{displaymath} \end{displaymath}}$

### inline vs display mode:

$$\text{\begin{equation} \sum_{i=0}^{\infty} \frac{n}{n+1} \end{equation}} \quad (1)$$

vs. This is inline:  $\sum_{i=0}^{\infty} \frac{n}{n+1}^2$

**code:**

```
\sum_{i=0}^{\infty} \frac{n}{n+1}
```

## 9 User-defined macros

[7]

Creating your own *commands* via chaining together different commands. The definition of which, must be in the preamble section of your .tex file.

Another possible use would be to redefine a command already defined by latex. This may lead to serious problems unlike our final safer *provide command* that defines the command if and only if (IFF) the command doesn't exist, otherwise no meaning is changed.

### 9.1 New Commands

```
\newcommand{\DALGO}{\vspace{.10in}
\textbf{Description of Algorithm:}}
```

I made this simple example while in cmps102 class. The **new command** defined here is  $\text{\DALGO}$ .

- $\text{\DALGO}$  produces the text "Description of Algorithm:"
- $\text{\newcommand}$  says we are creating a new command,
- $\text{\DALGO}$  named "DALGO",
- $\text{\vspace{.10in}}$  .10inches of vertical space between this command and anything following an instance of  $\text{\DALGO}$ ,
- $\text{\{Description of Algorithm:\}}$  which prints "Description of Algorithm:"

---

<sup>2</sup>for inline place sound eq. in  $\text{\$}$

This can be really useful when using the same material over and over again.  $\LaTeX$  also gives users the ability to redefine existing commands via the `\renewcommand`. An example of how this can be accomplished is as follows:

Original command is `\part{a}`. the output of the unmodified command is

### Part I

**a**

Now, our after redefining this command we get the following output: **(a)** "with text capabilities" immediately following the command.

#### Original Command

`\part{a}` with output

### Part I

**a**

Now, our after redefining this command we get the following output:

**(a)** "with text capabilities" immediately following the command.

`\providecommand{ }{ }` which is similar to `\newcommand{ }{ }` without a generated error message for the existence of a command.

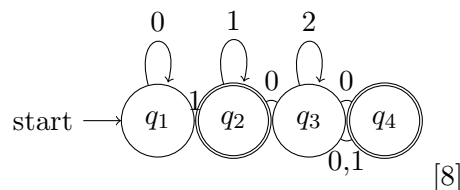
```
\renewcommand\part[1]
{\vspace{.10in}
\textbf{(#1)}}}
```

## 10 NFA's and DFA's

Of Coase!!  $\LaTeX$  has a library for almost everything. The specific library needed to draw Finite State Machines (FSM) is the *tikzautomata* library. Just like we have been doing, `\usepackage{tikz}` must be defined in the preamble. Now, for the correct positioning and arrows to show direction we must also use the `\usetikzlibrary{automata, positioning, arrows}`.

Nodes are what we are going to be working

with, so we want to connect these nodes in such a way that we get some FSM. Here is a Deterministic Finite Automata for some fun:



Here is code used:

```
\begin{tikzpicture}
\node[state, initial] (q1) {$q_1$};
\node[state, accepting, right of q1] (q2)
{$q_2$};
\node[state, accepting, right of q1] (q2)
{$q_2$};
\node[state, right of q2] (q3) {$q_3$};
\node[state, accepting, right of q3] (q4)
{$q_4$};
\draw (q1) edge[loop above] node{0} (q1)
(q1) edge[above] node{1} (q2)
(q2) edge[loop above] node{1} (q2)
(q2) edge[bend left, above] node{0} (q3)
(q3) edge[loop above] node{2} (q3)
(q3) edge[bend left, above] node{0} (q4)
(q4) edge[bend left, below] node{0,1} (q3);
\end{tikzpicture}
```

## 11 Bibliography

[9] The ability to cite different sources, inspirations and influence can be very important if you want to be respected in your field.

Of coarse,  $\LaTeX$  does have a bibliography environment called "the bibliography" but after doing much research it is much more flexible to create a .bib file, also known as a file that can dynamically receive entries, after we identify the type of source we are referencing. [10]

All references must begin with a "@" followed by type of reference. There are many different types of predefined *standard templates*. To list a few: @book{ }, @article{ }, and @misc{ }.

These are the entries in your .bib file that you will be inserting in-between

```
\begin{thebibliography}
\end{thebibliography}
```

Lastly, for  $\LaTeX$  to be able to find your .bib

file, the following two lines of code should be placed within your .tex file preceding `\end{document}`: `\bibliography{path_to_your_.bib}` and `\bibliographystyle{plain}`. Now, in order to cite your source within your article this command should follow any instance of another's work `\cite{ref1,ref2,...,refn}`.

## References

- [1] ShareLaTeX, "Latex table creation," <https://en.wikibooks.org/wiki/LaTeX/Tables/>.
- [2] Rjgunning and R. Gunning, "Latex: a scientist's trade secret to writing," <http://historytothepublic.org/latex-a-scientists-trade-secret-to-writing/>.
- [3] latex<sub>wikia</sub>, "*Paragraphmode*," .
- [4] G. Murray and S. Balemi, "Using the document class ieeetran.cls," <https://koclab.cs.ucsb.edu/teaching/ecc/project/paper/IEEEsample2e.pdf>.
- [5] A. S. . Unported, "Latex/labels and cross-referencing," [https://en.wikibooks.org/wiki/LaTeX/Labels\\_and\\_Cross-referencing#Examples](https://en.wikibooks.org/wiki/LaTeX/Labels_and_Cross-referencing#Examples).
- [6] P. A. J. Roberts, "Latex: Environments," <http://www.maths.adelaide.edu.au/anthony.roberts/LaTeX/ltxenviron.php>.
- [7] "Latex macros," [https://www.math.uh.edu/~torok/math\\_6298/latex/macros.html](https://www.math.uh.edu/~torok/math_6298/latex/macros.html).
- [8] S. Sikdar, "Drawing finite state machines in latex using tikz," [https://www3.nd.edu/~kogge/courses/cse30151-fa17/Public/other/tikz\\_tutorial.pdf](https://www3.nd.edu/~kogge/courses/cse30151-fa17/Public/other/tikz_tutorial.pdf).
- [9] L. F. Mori, "Managing bibliographies in latex," <https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf>.
- [10] ShareLaTeX, "Latextitle creation," [https://en.wikibooks.org/wiki/LaTeX/Title\\_Creation/](https://en.wikibooks.org/wiki/LaTeX/Title_Creation/).