

Another Short Tutorial for L^AT_EX 2_ε

Charles Carter

June 12, 2018

Contents

1	Introduction	3
2	Document Basics	4
2.1	Basic document	4
2.2	Basic title	4
2.3	Basic sections	5
2.4	Basic paragraphs	6
2.5	Basic packages	7
2.6	Basic contents	8
2.7	Basic decorations	9
2.8	Basic fontsizes	10
3	Math and Symbols	14
3.1	Special characters	14
3.2	Inline math	14
3.3	Equations	16
3.4	Multiline equations	17
3.5	Higher math	18
3.6	Theorems, etc.	19
	Fundamental Theorem of Calculus	19
	Proof of Fundamental Theorem of Calculus	19
	Definition of Calculus	20
3.7	Matrices	21
3.8	Greek letters	22
4	References	26
4.1	Text references	26
4.2	Footnotes	26
4.3	Endnotes	27
4.4	Margin notes	28
4.5	Bibliography	29
4.6	Citations	31
4.7	Indices	32
4.8	Includes	33

5	Lists	36
5.1	Verbatim	36
5.2	Unnnmbered Lists	36
5.3	Numbered Lists	37
5.4	Dictionary Lists	38
5.5	Nested Lists	38
5.6	Listings Package	41
5.7	Listings Version 1	42
5.8	Listings Version 2	44
6	Tables	47
6.1	Verbatim table	47
6.2	Simple table	47
6.3	Row and column lines	48
6.4	Column spacing	49
6.5	Captions and labels	51
6.6	Multicells	52
6.7	Long tables	55
7	Images	58
7.1	Image types	58
7.2	Including graphics	58
7.3	Image manipulation	59
7.4	59
7.5	Captions and labels	60
7.6	Colors	60
8	Advanced Topics	63
8.1	New commands	63
8.2	Templates	63
8.3	Geometry	64
8.4	Source listing	64
8.5	Presentations	65
8.6	Posters	68
8.7	Books	68
8.8	Auxiliary files	69
9	Conclusion	71
A	Installing L^AT_EX 2_ε	72
B	L^AT_EX 2_εEditors	73
C	Development Environments	73
D	Online L^AT_EX 2_ε	74
E	Command Line Execution	74
	Endnotes	75
	References	75

Index	75
--------------	-----------

List of Figures

1	MiKTeX download	72
2	MiKTeX on Windows start menu	72
3	CLI test	73

List of Tables

1	Greek letters	22
2	P columns	50
3	Names - Caption Above	51
4	Plants - Caption Below	51
5	Multicolumn Example	53
6	Multirow Example	54
7	Longtable example	55

1 Introduction

Let's begin thinking about L^AT_EX 2_ε by considering the concept of *easy*. We don't often think of easiness as having a context, but it certainly does. For example, which is easier to use, a shovel and wheelbarrow, or an excavator and dump truck? If we were digging a hole to plant a bush we would choose the former, but if we were digging a swimming pool we would choose the latter. Again, which is easier to use, a pickup truck or an eighteen wheeler? If we were hauling a few garden supplies we would choose the former, but if we were hauling electric generators to California, we would choose the latter. Which is easier to use, a word processing program (e.g., Microsoft Word), or a professional typesetting program (e.g., L^AT_EX 2_ε)? If we were writing a letter or a grocery list, we would choose the former, but if we were writing a mathematical, scientific, or engineering document, we would choose the latter. I can't say whether one choice is "easier" than the other, but I can say that it's easier to use the right tool for the job than to attempt a task using the wrong tool. It's in this spirit that I offer this tutorial.

This is a short, easy, nontechnical introduction to L^AT_EX 2_ε. It's a tutorial designed for students who do not have a lot of time, who do not need to become overnight experts in L^AT_EX 2_ε, and prefer a shallow learning curve to a boot camp approach. After you have worked through it, you will be able to create professional documents, and have the ability to teach yourself how to extend your L^AT_EX 2_ε skills — to become a L^AT_EX 2_ε guru if you want to or need to.

What about L^AT_EX 2_ε? First, it's *free* in the senses both that you do not have to pay for it and that you have the ability to change it without the permission of anyone. Second, it's *easy*, given that you know how to use it. As always, there are trade-offs, and "easy" to an expert tends to be hard for a beginner, and *vice versa*. Third, it's *stable*, the first version released in 1985, and documents written four decades ago still compile today. Fourth, it's *well documented*, not surprising since its purpose is document preparation. Fifth, it's *professional*, as you will soon come to see. Finally, L^AT_EX 2_ε is *universal*. I will touch on these points from time to time in this tutorial.

How does this introduction work? Each "lesson" consists of the introduction of a few commands, some text to copy, paste, and compile, and a couple of questions. Each lesson should not take more than fifteen minutes to complete. It uses the "baby talk" principle — you imitate what you see and explore it by making slight changes. If you complete one lesson a day, within several weeks you will have a good foundation with L^AT_EX 2_ε, and begin to create professional quality documents. You will also see how the things I wrote in the preceding paragraph are true.

Please note that this tutorial *is not* a L^AT_EX 2_ε reference; you will need to find a good reference that details the options and arguments for each command. It's also not a user guide. Nor is it a full introduction to L^AT_EX 2_ε, although it will introduce it to students not previously familiar with it. It gives an ideosyncratic view of L^AT_EX 2_ε (my own). It promises only to be short, easy, and useful, not long-winded or difficult, even if it does omit some very important details.

2 Document Basics

A `tex` document consists of plain text, and special characters, commands, and environments. I will generally refer to special characters, commands, and environments with the word *commands*; you don't need to know the difference between them now, but you shortly will without being told. You *must* precede commands with a backslash (`\`) for the compiler to know that they are commands. This is easy to forget, so I will remind you the first couple of times.

What should I have? You should have a working $\text{\LaTeX} 2_{\epsilon}$ program. If you do not have one, see Appendix A below on page 72. You will also need a text editor, and you will probably want to get an integrated editor, compiler, and printer. See Appendix C on page 73. You can also use the old fashioned command line, I cover this in Appendix E on page 74.

2.1 Basic document

Commands

- `documentclass`
- `begin/end` document
- plain text
- comments

A basic document begins with a document class, and has a preamble and contents. Type (or copy) the following, save it as a `.tex` document and compile it. The percent signs (`%`) are comments and do not have any effect on the document.

```
\documentclass{article}
    %this is the preamble
\begin{document}
    %this is the contents section
    It works! %plain text prints as is
\end{document}
```

Exercise: $\text{\LaTeX} 2_{\epsilon}$ has a number of different document classes. Name four of them. Hint: use a search engine for “`latex document classes`”.

Exercise: A `documentclass` command can take optional arguments, like this: `documentclass[optional arguments]{document class}`.¹ Name two optional arguments.

2.2 Basic title

Commands

- `title`
- `author`

¹Don't forget to type a backslash before the command, like this: `\documentclass`

- `date`
- `maketitle`

A basic document usually title and author information in the preamble. You specify the title with the `title` command. You specify the author(s) with the `author` command. You may optionally specify a date with the `date` command. In the body of the document, you create the title with the `maketitle` command. Create and compile a second document like this:

```
\documentclass{article}
  \title{Title, Author, and Date}
  \author{Charles Carter}
  \date{July 4, 1776}
\begin{document}
  \maketitle{}
  This document has a title, author, and date.
\end{document}
```

Exercise: What happens if you use the command `today{}`² as the date parameter (replacing July 4, 1776)?

Exercise: What happens if you use the command `thanks{email address}`³ after your name in the `author{}` command?

2.3 Basic sections

Commands

- `section`
- `subsection`
- `subsubsection`
- `label`

$\text{\LaTeX} 2_{\epsilon}$ provides a number of useful section levels, including part and chapter. Two of the most useful are `section` and `subsection`. Create and compile the following document.

The `label{}` is used to create cross-references in documents. It's also very helpful in organizing your thoughts.⁴ The argument to `label{argument}` does not appear in the document. I cover the `ref` and `pageref` commands below in subsection 4.1 on page 26. These are used to create references back to the label.

```
\documentclass{article}
  \title{Basic Sections}
```

²Remember, `\date{}`

³`\thanks{}`

⁴Sometimes, people say that $\text{\LaTeX} 2_{\epsilon}$ is a language for logical marup. As you continue to author documents with $\text{\LaTeX} 2_{\epsilon}$, you will increasingly come to rely on the `label` command to organize your paper and structure your thoughts. In my opinion, one of the chief benefits of using $\text{\LaTeX} 2_{\epsilon}$ is that it enforces this kind of structured process on your writing, and you will become a better writer because of it,

```

\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\section{Introduction}
\label{Introduction}
\section{Body}
\label{Body}
\section{Conclusion}
\label{Conclusion}
\end{document}

```

Exercise: What do the commands `subsection{}` and `subsubsection{}` do?

Exercise: What does `section*{}` do? Note the asterisk (*) after `section`. You can also use this starred version for `subsections` and `subsubsections`.

2.4 Basic paragraphs

Commands

- `paragraph`
- `subparagraph`

We have reached the point where you need some real content. I will use the text of Abraham Lincoln’s Gettysburg Address to illustrate paragraphs. Notice that ordinary paragraphs do not need a special command – the “paragraph command” is simply two blank lines to create an empty new line between the paragraphs, as if they were double spaced. Create and compile the following document.

```

\documentclass{article}
\title{Basic Paragraphs}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\section{Introduction}
\label{Introduction}
\section{Body}
\label{Body}

```

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battlefield of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting

and proper that we should do this.

But, in a larger sense, we can not dedicate, we can not consecrate, we can not hallow this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us, that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion, that we here highly resolve that these dead shall not have died in vain, that this nation, under God, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish from the earth.

```
\section{Conclusion}
\label{Conclusion}
\end{document}
```

Exercise: What happens if you include the `paragraph{}` or `subparagraph{}` commands before each paragraph?

Exercise: What happens if you include arguments with the `paragraph{argument}` or `subparagraph{argument}` commands?

2.5 Basic packages

Commands

- `usepackage`
- `lipsum`

Much of L^AT_EX 2_ε functionality is contained in external packages. To use this functionality, you include the command `usepackage{}` in the preamble. Of course, you first have to install the package on your computer, but the MiKTeX distribution does that automatically. The `lipsum` package generates generic text (in Latin, of course). The `lipsum{}` command generates text. Notice that you can control the number of paragraphs to include. Below, I have included paragraph 1 in the introduction, paragraphs 2 through 4 in the body, and paragraph 5 in the conclusion.

Notice the paragraph indentation. First paragraphs are *not* indented. Following paragraphs *are* indented. This is normal typographic practice.

```
\documentclass{article}
\usepackage{lipsum}
\title{Using Packages}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
```



```

\section{Introduction}
\label{Introduction}
\lipsum[1]{}
\section{Body}
\label{Body}
\lipsum[2-4]{}
\section{Conclusion}
\label{Conclusion}
\lipsum[5]{}
\end{document}

```

Exercise: What is CTAN, the Comprehensive T_EX Archive Network? How many packages are currently on CTAN?

Exercise: What are the most popular L^AT_EX 2_ε packages?

2.6 Basic contents

Commands

- tableofcontents

Creating a table of contents is easy. Just include the `tableofcontents{}` command. You may have to compile the document twice to ensure that the table of contents is generated properly.

```

\documentclass{article}
\usepackage{lipsum}
\title{Table of Contents}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\tableofcontents{}
\section{Introduction}
\label{Introduction}
\lipsum[1]{}
\section{Body}
\label{Body}
\lipsum[2-4]{}
\section{Conclusion}
\label{Conclusion}
\lipsum[5]{}
\end{document}

```

Exercise: The `section[Optional Argument]{Section Title}` command takes an optional argument. How does this argument affect the table of contents?

Exercise: What other kinds of content tables can L^AT_EX 2_ε generate? To start with, look at figures and tables.

2.7 Basic decorations

Commands

- `textit`
- `textsf`
- `texttt`
- `textbf`
- `textsc`
- `underline`

In this section, you will fiddle with the appearance of text. To *create text in italics*, use `textit`. To create text in sans serif, use `textsf`. To create text in monospace font, use `texttt`. To create text in boldface, use `textbf`. To CREATE TEXT USING SMALL CAPS, use `textsc`. You should almost never underline text! If you choose to do so, use `underline`.

Illustration

To *create text in italics*, use `textit`.

To create text in sans serif, use `textsf`.

To create text in monospace font, use `texttt`.

To create text in boldface, use `textbf`.

To CREATE TEXT USING SMALL CAPS, use `textsc`.

You should almost never underline text! If you choose to do so, use `underline`.

```
\documentclass{article}
  \title{Font Appearance}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
    \paragraph{}In this section, you will fiddle with the appearance of text.
    \paragraph{}To \code{textit}{create text in italics}, use \code{texttt}{textit}.
    \paragraph{}To \code{textsf}{create text in sans serif}, use \code{texttt}{textsf}.
    \paragraph{}To \code{texttt}{create text in monospace font}, use \code{texttt}{texttt}.
    \paragraph{}To \code{textbf}{create text in boldface}, use \code{texttt}{textbf}.
    \paragraph{}To \code{textsc}{create text using Small Caps}, use \code{texttt}{textsc}.
    \paragraph{}\code{underline}{You should almost never underline text}!
```

```

        If you choose to do so, use \texttt{underline}.
\section{Conclusion}
\label{Conclusion}
\end{document}

```

Exercise: As with much else in $\text{\LaTeX} 2_{\epsilon}$, there are multiple ways to italicize or bold-face text. Can you find other ways?

2.8 Basic font sizes

Commands

- tiny
- scriptsize
- footnotesize
- small
- normalsize
- large
- Large
- LARGE
- huge
- Huge

$\text{\LaTeX} 2_{\epsilon}$ has several different ways to alter the size of the font. Perhaps the simplest way is to create a *size environment*. You do this by using one of the commands listed above, and this controls the size of all text until it is changed by another command. You would typically use this for sections of text that need to be made smaller, such as tables, block quotes, technical sections not germane to the main discussion, and similar.

Illustration

This paragraph has a normalsize font size.

This paragraph has a tiny font size.

This paragraph has a scriptsize font size.

This paragraph has a footnotesize font size.

This paragraph has a small font size.

This paragraph has a normalsize font size.

This paragraph has a large font size.

This paragraph has a Large font size.

This paragraph has a LARGE font size.

This paragraph has a huge font size.

This paragraph has a Huge font size.

This paragraph has a normalsize font size.

```
\documentclass{article}
  \title{Font Sizes}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \normalsize{}\paragraph{}This paragraph has a normalsize font size.
  \tiny{}\paragraph{}This paragraph has a tiny font size.
  \scriptsize{}\paragraph{}This paragraph has a scriptsize font size.
  \footnotesize{}\paragraph{}This paragraph has a footnotesize font size.
  \small{}\paragraph{}This paragraph has a small font size.
  \normalsize{}\paragraph{}This paragraph has a normalsize font size.
  \large{}\paragraph{}This paragraph has a large font size.
  \Large{}\paragraph{}This paragraph has a Large font size.
  \LARGE{}\paragraph{}This paragraph has a LARGE font size.
  \huge{}\paragraph{}This paragraph has a huge font size.
  \Huge{}\paragraph{}This paragraph has a Huge font size.
  \normalsize{}\paragraph{}This paragraph has a normalsize font size.
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise: The issues of font, font size, and font decoration, are difficult, complicated, and subject to internecine wars. You may want to postpone your exploration of these issues until you have created and compiled several hundred `.tex` documents. If you want, and have discretionary time available and nothing else to do, you may want to delve into the complex and divisive world of fonts, font sizes, and font decorations.

$\text{\LaTeX } 2_{\varepsilon}$ is *Universal*

3 Math and Symbols

Both \TeX and \LaTeX shine when it comes to math. In fact, Donald Knuth originally wrote \TeX just so he could typeset math. In this section, we will dip our toes into math and symbols. This will not be difficult. If you have need for more advanced mathematics, you will know how to find what you need to render your equations.

3.1 Special characters

Most characters are not special. An a is just an a , a Z is just a Z , and a 7 is just a 7 . Sometimes, this isn't the case — an $\&$ is not just an ampersand. \LaTeX has ten special characters. They are listed below.

Commands

- backslash - \backslash
- percent - $\%$
- left curly bracket - $\{$
- right curly bracket - $\}$
- dollar sign - $\$$
- caret - \wedge
- underscore - $_$
- tilde - \sim
- hash - $\#$
- ampersand - $\&$

You already know four of them. “ \backslash ” indicates the beginning of a command, “ $\%$ ” indicates a comment, and the “ $\{ \dots \}$ ” pair (usually) indicates the argument to a command. You will learn about three more in this section, the dollar sign “ $\$$ ”, the underscore “ $_$ ”, and the caret “ \wedge ”. It's worthwhile to stare at these ten characters long enough to become familiar with them. When your document misbehaves, often these characters are the culprit.

Sometimes you will find characters that wish they were special, but are not. These include the cedilla ($\text{\c}[/math>), the degree (\textdegree), and diphthongs (\ae). All these are represented by \LaTeX commands, you will use the command for the character.$

Exercise: Scott Pakin has published the booklet *The Comprehensive \LaTeX Symbol List*. You can find this online in PDF format. Search for it and just look at it. It contains over 300 pages of symbols. You'll be amazed!

3.2 Inline math

Commands

- $\$$

- plus or +
- - (dash or subtraction)
- times or ast or cdot
- frac or div
- sqrt
- ^ (caret or circumflex)
- _ (underscore)

These commands represent the basic arithmetic operations of addition, subtraction, multiplication, and division. These also include the square root and exponents.

Illustration

This is an example of inline math.
 Use the dollar symbol (\$) to set the math.
 Here is how it works.
 Addition: $4 + 5 = 9$.
 Subtraction: $4 - 5 = -1$.
 Multiplication: $4 \times 5 = 20$.
 Multiplication: $4 \cdot 5 = 20$.
 Multiplication: $4 * 5 = 20$.
 Division: $\frac{4}{5} = 0.8$.
 Division: $4 \div 5 = 0.8$.
 Square root: $\sqrt{2} = 1.41421$.
 Higher roots: $\sqrt[4]{81} = 3$.
 Exponents: $2^8 = 256$.
 Subscripts: x_0, x_1, x_2 .

```
\documentclass{article}
\title{Inline Math}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
This is an example of inline math. Use the dollar symbol (\$) to set the math. \\
Here is how it works. \\
Addition: $4 + 5 = 9$. \\
Subtraction: $4 - 5 = -1$. \\
Multiplication: $4 \times 5 = 20$. \\
Multiplication: $4 \cdot 5 = 20$. \\
Multiplication: $4 \ast 5 = 20$. \\
Division: $\frac{4}{5} = 0.8$. \\
Division: $4 \div 5 = 0.8$. \\
Square root: $\sqrt{2} = 1.41421$. \\
Higher roots: $\sqrt[4]{81} = 3$. \\
Exponents: $2^8 = 256$. \\
Subscripts: $x_0, x_1, x_2$
\end{document}
```


Exercise: You can find the *User's Guide for the amsmath Package* in PDF format online. Search for it and start reading through it.

3.3 Equations

Commands

- amsmath
- equation
- equation*

L^AT_EX 2_ε provides the *equation* environment for writing block equations with the *amsmath* package. First, import the package with `\usepackage{amsmath}`. Equations are numbered and can be referenced by means of their labels. The starred version omits the equation from the numbered equations. Here are some examples. Equation 1 is the formula for a straight line. Equation 2 is the formula for the slope of a straight line. The third, unnumbered equation is the formula for a straight line with multiple parameters.

Illustration

$$y = \beta_0 + \beta_1 x_1 \tag{1}$$

$$m = \frac{y_1 - y_0}{x_1 - x_0} \tag{2}$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

```
\documentclass{article}
\usepackage{amsmath}
  \title{Equations}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
This is an example of equations.
\begin{equation}
\label{line}
y = \beta_0 + \beta_1 x_1
\end{equation}
\begin{equation}
\label{slope}
m = \frac{y_1 - y_0}{x_1 - x_0}
\end{equation}
\begin{equation*}
y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3
\end{equation*}
\end{document}
```

Exercise: Continue reading through the *User's Guide for the amsmath Package*.

3.4 Multiline equations

Commands

- align
- gather
- multiline

How do I place several equations together in one equation environment, aligned on a particular character, such as an equal sign (=)? Use the *align* environment, with the ampersand (&) as the tab character, and end each line with two backslashes (\\).

Illustration

$$y = \beta_0 + \beta_1 x_1 \quad (3)$$

$$\text{slope} = \frac{y_1 - y_0}{x_1 - x_0} \quad (4)$$

$$\text{predictedvalue} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \quad (5)$$

How to I center the equations? Use the *gather* environment, with no tab character but ending each line with two backslashes (\\).

Illustration

$$y = \beta_0 + \beta_1 x_1 \quad (6)$$

$$\text{slope} = \frac{y_1 - y_0}{x_1 - x_0} \quad (7)$$

$$\text{predictedvalue} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \quad (8)$$

What if I have a very long equation that won't fit on one line? Use the *multiline* environment, breaking with two backslashes (\\)

Illustration

$$\begin{aligned} y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \\ \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \\ \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 \end{aligned} \quad (9)$$

```
\documentclass{article}
\usepackage{amsmath}
\title{Multiline Equations}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
This is an example of align.
\begin{align}
```

```

y& = \beta_0 + \beta_1 x_1\\
slope& = \frac{y_1 - y_0}{x_1 - x_0}\\
predictedvalue& = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3
\end{align}
This is an example of gather.
\begin{gather}
y = \beta_0 + \beta_1 x_1\\
slope = \frac{y_1 - y_0}{x_1 - x_0}\\
predictedvalue = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3
\end{gather}
This is an example of multiline.
\begin{multiline}
y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 +\\
\beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 +\\
\beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9
\end{multiline}
\end{document}

```

3.5 Higher math

Commands

- sums
- products
- limits
- derivatives
- integrals

Equation 10 states the shorthand for the sum of a series of integers 1 through n . Equation 11 states the shorthand for the product of a series of integers 1 through n . Equation 12 demonstrates the notation for limits. Equation 13 demonstrates the notation for derivatives. Equation 14 demonstrates the notation for integrals.

Illustration

$$\sum_{i=1}^{i=n} i = 1 + 2 + 3 + \cdots + n \quad (10)$$

$$\prod_{i=1}^{i=n} i = 1 \times 2 \times 3 \times \cdots \times n \quad (11)$$

$$\lim_{x \rightarrow \infty} f(x) \quad (12)$$

$$\frac{d}{dx} [e^{\frac{x}{2}} \sin(ax)] \quad (13)$$

$$\int_a^b x^2 dx \quad (14)$$

```
\documentclass{article}
```

```

\usepackage{amsmath}
\title{Higher Math}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\begin{gather}
\sum_{i=1}^{i=n} i = 1 + 2 + 3 + \dots + n \\
\prod_{i=1}^{i=n} i = 1 \times 2 \times 3 \times \dots \times n \\
\lim_{x \rightarrow \infty} f(x) \label{higher:lim} \\
\frac{d}{dx} \left[ e^{\frac{x}{2}} \sin(ax) \right] \\
\int_a^b x^2 dx
\end{gather}
\end{document}

```

Exercise: Continue reading through the *User's Guide for the amsmath Package*.

3.6 Theorems, etc.

Commands

- amsthm package
- newtheorem
- addcontentsline

Stating theorems, lemmas, proofs, axioms, and similar constructions in L^AT_EX 2_ε is particularly easy. For simple applications, just define whatever environment you need in the preamble with `newtheorem`, like this: `newtheorem{first argument}{second argument}`. The *first argument* is the name of the structure: theorem, lemma, definition, etc. The *second argument* is the printed heading that the reader will see in the document.

For more flexibility (and I do mean a *lot* more flexibility) use the `amsthm` package, imported as usual in the preamble with `usepackage{amsthm}`. This package allows many customizations, for example, numbering by section and subsection, and concurrent numbering of theorems and proofs. The documentation of this package is fairly short and not overly complex, so it's not hard to read, understand, and use.

Finally, you might want to add your theorem to the table of contents. Use `addcontentsline{first argument}{second argument}{third argument}`. The *first argument* is the listing where you want the entry to appear: table of contents, list of tables, or list of figures. The *second argument* is the kind of entry you want, such as section, subsection, etc. The *third argument* is the printed text that will appear in the contents section.

Illustration

Theorem 1 *Definite integral of a function is related to its antiderivative, and can be reversed by differentiation.*

Proof 1 *If f is continuous on $[a, b]$, then $\int_a^b f$ exists.*

If f is continuous on $[a, b]$ and $c \in [a, b]$, then $\int_a^c F + \int_c^b F = \int_a^b F$.
 If $m \leq f \leq M$ on $[a, b]$, then $(b - a)m \leq \int_a^b f \leq (b - a)M$.

Definition 1 *Calculus is the branch of mathematics that deals with the finding and properties of derivatives and integrals of functions, by methods originally based on the summation of infinitesimal differences. The two main types are differential calculus and integral calculus.*

For the definition of calculus, see definition 1. For the fundamental theorem of calculus, see theorem 1. For the proof, see proof 1.

```
\documentclass{article}
\usepackage{amsmath}
\newtheorem{theorem}{Theorem}
\newtheorem{proof}{Proof}
\newtheorem{definition}{Definition}
\title{Theorems, Proofs, Definitions, etc.}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
Theorems, proofs, definitions, etc. can easily be defined
\begin{theorem}
\label{thm:first}
\addcontentsline{toc}{subsubsection}{Fundamental Theorem of Calculus}
Definite integral of a function is related to its antiderivative,
and can be reversed by differentiation.
\end{theorem}
\begin{proof}
\label{prf:first}
\addcontentsline{toc}{subsubsection}{Proof of Fundamental Theorem of Calculus}
If  $f$  is continuous on  $[a, b]$ ,
then  $\int_a^b f$  exists. \\
If  $f$  is continuous on  $[a, b]$  and  $c \in [a, b]$ ,
then  $\int_a^c F + \int_c^b F = \int_a^b F$ . \\
If  $m \leq f \leq M$  on  $[a, b]$ ,
then  $(b - a)m \leq \int_a^b f \leq (b - a)M$ .
\end{proof}
\begin{definition}
\label{def:first}
\addcontentsline{toc}{subsubsection}{Definition of Calculus}
Calculus is the branch of mathematics that deals with the
finding and properties of derivatives and integrals of
functions, by methods originally based on the summation
of infinitesimal differences. The two main types are
differential calculus and integral calculus.
\end{definition}
For the definition of calculus, see definition \ref{def:first}.
For the fundamental theorem of calculus, see theorem \ref{thm:first}.
For the proof, see proof \ref{prf:first}.
\end{document}
```

Exercise: Create your own new theorem class, called Problem, perhaps, and write a problem.

Exercise: Access the documentation to the amsthm package and look through it.

3.7 Matrices

Commands

- `\vmatrix`
- `\pmatrix`
- `\bmatrix`
- `\Bmatrix`
- `\Vmatrix`

Matrices are contained within a math environment, either an equation block or inline math pairs (\dots). They are entered by row. Each row ends with two backslash symbols (`\`). Each element on a row is separated by an ampersand (`&`). The different variations create different surrounding brackets. `\vmatrix` uses vertical bars ($|\dots|$), for example, to indicate the determinant of the matrix. `\pmatrix` uses parentheses ((\dots)). `\bmatrix` uses square brackets ($[\dots]$). `\Bmatrix` uses curly braces ($\{\dots\}$). `\Vmatrix` uses double vertical bars ($\| \dots \|$).

Illustration

$$\begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix} \times \begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
\documentclass{article}
\usepackage{amsmath}
\title{Matrices}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
$ \begin{bmatrix}
1 & 3 & 3 \\
1 & 4 & 3 \\
1 & 3 & 4
\end{bmatrix}
\times
\begin{bmatrix}
7 & -3 & -3 \\
-1 & 1 & 0 \\
-1 & 0 & 1
\end{bmatrix}
=
```

```

\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\end{document}

```

Exercise: Continue reading through the *User's Guide for the amsmath Package*.

3.8 Greek letters

Here are the Greek letters commonly used in mathematical and scientific applications, table 3.8 on page 22. The $\text{\LaTeX} 2_{\epsilon}$ commands for the Greek letters are in inline math mode, so you *must* surround the command by a pair of dollar signs (\dots). You must also use the `amsmath` package.

Name	UC command	UC Letter	LC command	LC letter
Alpha	Alpha	A	alpha	α
Beta	B	B	beta	β
Gamma	Gamma	Γ	gamma	γ
Delta	Delta	Δ	delta	δ
Epsilon	E	E	epsilon	ϵ
Zeta	Z	Z	zeta	ζ
Eta	H	H	eta	η
Theta	Theta	Θ	theta	θ
Iota	I	I	iota	ι
Kappa	K	K	kappa	κ
Lambda	Lambda	Λ	lambda	λ
Mu	M	M	mu	μ
Nu	N	N	nu	ν
Xi	Xi	Ξ	xi	ξ
Omicron	O	O	o	o
Pi	Pi	Π	pi	π
Rho	R	R	rho	ρ
Sigma	S	S	sigma	σ
Tau	T	T	tau	τ
Upsilon	Y	Y	upsilon	υ
Phi	Phi	Φ	phi	ϕ
Chi	X	X	chi	χ
Psi	Psi	Ψ	psi	ψ
Omega	Omega	Ω	omega	ω

Table 1: Greek letters

```

\documentclass{article}
\usepackage{amsmath}
\title{Greek Letters}
\author{Charles Carter}
\date{\today}

```

```

\begin{document}
\maketitle
\begin{tabular}{|| 1 || 1 | 1 || 1 | 1 ||}
\hline
Name & UC command & UC Letter & & LC command & LC letter \\
\hline
\hline
Alpha & Alpha &  $A$  & & alpha &  $\alpha$  \\
\hline
Beta & B &  $B$  & & beta &  $\beta$  \\
\hline
Gamma & Gamma &  $\Gamma$  & & gamma &  $\gamma$  \\
\hline
Delta & Delta &  $\Delta$  & & delta &  $\delta$  \\
\hline
Epsilon & E &  $E$  & & epsilon &  $\epsilon$  \\
\hline
Zeta & Z &  $Z$  & & zeta &  $\zeta$  \\
\hline
Eta & H &  $H$  & & eta &  $\eta$  \\
\hline
Theta & Theta &  $\Theta$  & & theta &  $\theta$  \\
\hline
Iota & I &  $I$  & & iota &  $\iota$  \\
\hline
Kappa & K &  $K$  & & kappa &  $\kappa$  \\
\hline
Lambda & Lambda &  $\Lambda$  & & lambda &  $\lambda$  \\
\hline
Mu & M &  $M$  & & mu &  $\mu$  \\
\hline
Nu & N &  $N$  & & nu &  $\nu$  \\
\hline
Xi & Xi &  $\Xi$  & & xi &  $\xi$  \\
\hline
Omicron & O &  $O$  & & o &  $o$  \\
\hline
Pi & Pi &  $\Pi$  & & pi &  $\pi$  \\
\hline
Rho & R &  $R$  & & rho &  $\rho$  \\
\hline
Sigma & S &  $S$  & & sigma &  $\sigma$  \\
\hline
Tau & T &  $T$  & & tau &  $\tau$  \\
\hline
Upsilon & Y &  $Y$  & & upsilon &  $\upsilon$  \\
\hline
Phi & Phi &  $\Phi$  & & phi &  $\phi$  \\
\hline
Chi & X &  $X$  & & chi &  $\chi$  \\
\hline

```



```

        Psi & Psi & $\Psi$      & psi & $\psi$ \\
        \hline
        Omega & Omega & $\Omega$      & omega & $\omega$ \\
        \hline
    \end{tabular}
\end{document}

```

Exercise: Mathematics uses a large number of non-Latin characters. For example, search for the Hebrew character Aleph (\aleph) and the symbol for infinity (∞).

L^AT_EX 2_ε is *Professional*

L^AT_EX 2_ε is flexible to an extreme. You can customize every part of your document. However, its authors have spent much time and effort to ensure that it gives a professional appearance “out of the box.” This short tutorial does not cover customization. After you have created and compiled several dozen `.tex` files, you will begin to see the need for customization.⁵ Unless you have special requirements, such as page margins, paragraph spacing, and page numbers, and you are happy with the appearance of your document, you do not need to think about customization.

⁵The first thing you may want to do is to change the page margins. An easy way to change page margins is to use the **geometry package**, covered in section 8.3 on page 64

4 References

In scientific and research writing, one of the most critical aspects is references. Think about it: can you imagine a research paper without a single reference? These include footnotes, endnotes, marginal notes, and especially citations to sources. References have two components, a target and a source, or a label and a reference to that label. We have used labels before, but now we will explicitly consider them.

4.1 Text references

Commands

- label
- ref
- pageref

A text reference has two parts, a target and a pointer. The pointer points to a target, and a target is pointed to with a pointer. Targets are identified using the `label` command. Pointers are created with the `ref` command for numerical designations, and the page on which the target appears with the `pageref` command. For example, the table of Greek letters on page 22 is table 3.8. Targets can include structural elements (chapters, sections, paragraphs, etc.), tables, and figures.

```
\documentclass{article}
  \title{Footnotes and References}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  This is the introduction; for the conclusion, see
  section \ref{Conclusion} on page \pageref{Conclusion}.
  \section{Conclusion}
  \label{Conclusion}
  This is the conclusion; for the introduction, see
  section \ref{Introduction} on page \pageref{Introduction}.
\end{document}
```

Exercise: Create and compile a document with at least two of each: labels, refs, and pagerefs.

4.2 Footnotes

Commands

- footnote
- label

- `ref`
- `pageref`

To insert a footnote, just use the command `footnote`.⁶ If you label the footnote, you can refer to the footnote by number and page in the text of the document. Please be sure to read footnote 6 on page 27.

```
\documentclass{article}
  \title{Footnotes and References}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  To insert a footnote, just use the command \text{footnote}.
  \footnote{\label{references:fn} Don't forget to put the
    \textbackslash{} before the \texttt{footnote} command.}
  If you label the footnote, you can refer to the footnote
  by number and page in the text of the document. Please
  be sure to read footnote \ref{references:fn} on page
  \pageref{references:fn}.
\end{document}
```

Exercise: Read the L^AT_EX 2_ε documentation on footnotes.

4.3 Endnotes

Commands

- `package endnotes`
- `endnote`
- `theendnotes`
- `addcontentsline`

Endnotes are a little more complicated than footnotes, but not much.¹ Here is an endnote.² In order to actually print the endnotes, use the `theendnotes` command. In order to create an entry for the endnotes in the table of contents, you must use the `addcontentsline`.³ Please see endnote 3 on page 75 for the details.

```
\documentclass{article}
  \usepackage{endnotes}
  \title{Endnotes}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \tableofcontents{}
  \section{Text}
```

⁶Don't forget to put the `\` before the `footnote` command.

Endnotes are a little more complicated than footnotes, but not much. `\endnote{The difference between footnotes and endnotes is that footnotes go at the foot of the page where they appear, while endnotes appear at the end of the document.}` Here is an endnote. `\endnote{This is an endnote.}` In order to actually print the endnotes, use the `\texttt{theendnotes}` command. In order to create an entry for the endnotes in the table of contents, you must use the `\texttt{addcontentsline}`. `\endnote{\label{references:en}}` The `addcontentsline` takes three parameters, where the line should be written, usually `\textit{toc}`, the formatting to be used, usually `\textit{section}`, and the name to be given to the entry, perhaps `\textit{Endnotes}`.} Please see endnote `\ref{references:en}` on page `\pageref{references:en}` for the details.

```
\theendnotes{}
\addcontentsline{toc}{section}{Endnotes}
\end{document}
```

Exercise: Find and read through the documentation of the Endnotes package.

4.4 Margin notes

Commands

- `marginpar`
- `reversemarginpar`
- `normalmarginpar`
- `raggedright`

Marginal notes are really useful to call attention to very important points. They can also be useful in drafting documents to note future edits, insertions, or deletions. Use `marginpar` to create a marginal note. To delete marginal notes during the drafting process, just comment them out. The text remains as a reminder of the modifications in the document.

To permanently reverse the page sides, use `reversemarginpar`. To reverse the reverse page margins, use `normalmarginpar`. To change the paragraph alignment from the default justified text, use the command `raggedright` in a block.

```
\documentclass{article}
\title{Margin Notes}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\marginpar{Important point!}
Marginal notes are really useful to call attention to very important
points. They can also be useful in drafting documents to note
future edits, insertions, or deletions. Use \texttt{marginpar}
to create a marginal note. To delete marginal notes during the
drafting process, just comment them out. The text remains as a
```

Important
point!

This is in
the correct
margin

This is re-
versed text

```

    reminder of the modifications in the document.
    {\raggedright\reversemarginpar{\marginpar{This is reversed text}}}
    To permanently reverse the page sides, use \texttt{reversemarginpar}.
    To reverse the reverse page margins, use \texttt{normalmarginpar}.
    {\raggedright\normalmarginpar\marginpar{This is in the correct margin}}
\end{document}

```

Exercise: If you really need to create sophisticated marginal notes, you will need both the `geometry` and `marginnote` packages. The former is very useful, and you may need to use it in every paper. The latter is useful for marginal notes when you need something more than the default $\text{\LaTeX} 2_{\epsilon}$ commands.

4.5 Bibliography

Commands

- creating an external Bib \TeX database

Needless to say, every research paper requires a bibliography. $\text{\LaTeX} 2_{\epsilon}$ has multiple ways to include bibliographies. I have chosen to use a method called Bib \TeX . There are three reasons for this choice. First, it's simple enough to include in a simple tutorial. Second, it gives good results using its default settings. Third, it's complex enough to be configured for almost any application (given an author's time and patience).

Bib \TeX uses an external file as a bibliographical database. This means that management of the sources is separate from writing and editing the original document. The data from the external database is merged into the original document with a series of commands. We will cover the basics of the external database in this lesson. In the following lesson, we will see how to merge the two files.

The external Bib \TeX database is merely a plain text file with entries in a prescribed format. It *must* have the `.bib` extension for the filename. The following file includes four books and three online sources. The Bib \TeX format file contains specifications for a large number of types of sources, and you should become familiar with the various types of documents and the format for each.

Create a document with the following content. Name it `tut.bib` and place it in the same directory with your original source document, where you compile your PDF output. Each entry has a type identifier preceded by the at symbol (`@`) and followed by a curly brace pair (`{...}`), within which are contained various fields separated by commas. The first field is the identifier you will use to access the entry. The subsequent fields are “key”=“value” pairs, giving the title, author, date etc.

```

\\tut.bib
@book{
  goossens04,
  author   = "Frank Mittelbach and Michel Goossens and
             Johannes Braams and David Carlisle and
             Chris Rowley",
  title    = "The \LaTeX{} Companion (Tools and
             'Techniques for Computer Typesetting)",
  year     = "2004",

```

```

        edition    = "2nd",
        publisher   = "Addison-Wesley",
        address     = "Reading, MA",
        ISBN        = "978-0201362992"
    }
    @book{
        kottwitz11,
        author       = "Stefan Kottwitz ",
        title        = "\LaTeX{} Beginner's Guide",
        year         = "2011",
        publisher     = "Packt Publishing",
        address      = "Birmingham, UK",
        ISBN         = "978-1847199867"
    }
    @book{
        kottwitz15,
        author       = "Stefan Kottwitz ",
        title        = "\LaTeX{} Cookbook",
        year         = "2015",
        publisher     = "Packt Publishing",
        address      = "Birmingham, UK",
        ISBN         = " 978-1784395148"
    }
    @book{
        gratzer14,
        author       = "George Gratzer",
        title        = "Practical \LaTeX{}",
        year         = "2014",
        publisher     = "Springer",
        address      = "New York, NY",
        ISBN         = "978-1847199867"
    }
    @electronic{
        oetiker15,
        author       = "Tobias Oetiker and Hubert Partl and Irene Hyna and Elisabeth Schlegl",
        title        = "The Not So Short Introduction to {LaTeX2e}",
        url          = "http://tug.ctan.org/info/lshort/english/lshort.pdf",
        year         = "2015",
        note         = "accessed August 2, 2016"
    }
    @electronic{
        pakin15,
        author       = "Scott Pakin",
        title        = "The Comprehensive {LATEX} Symbol List",
        url          = "http://tug.ctan.org/info/symbols/comprehensive/symbols-letter.pdf",
        year         = "2015",
        note         = "accessed August 2, 2016"
    }
    @electronic{
        carter16,
        author       = "Charles Carter",

```

```

title = "Another Short Tutorial to {LaTeX2e}",
url = "https://github.com/ccc31807/latex-short-intro",
year = "2016",
note = "accessed August 2, 2016"
}

```

Exercise: Search for the Bib_T_EX format specification. Name four different kinds of source documents and identifies the required elements for each.

4.6 Citations

Commands

- cite
- nocite
- bibliographystyle
- bibliography
- addcontentsline
- bibtex

Preparing the bibliographic database is fairly straightforward and easy. Unfortunately, the details of including the citations in your document is not as straightforward.⁷ To cite to a reference, use the `cite` command, passing the citation reference as an argument. By default, the printed bibliography includes only cited references — to include a particular uncited reference, use the `nocite` command, passing the citation reference as an argument. To include all uncited references, use the command `nocite{*}`. (The asterisk is a typeglob that means “everything”).

To specify a citation style, use the command `bibliographystyle`. As you will see when you study the different styles, there are many of them. Adding to the amount of effort of work potentially involved, you can write your own style specification if you don’t want to use any that are available. In the example below, I have used the *IEEEtranS* style. This will probably not meet your needs. You will have to find an appropriate style for your requirements.

To include the bibliography itself, use the command `bibliography`. To add your references to the table of contents, use the command `addcontentsline`. To compile your bibliography using the command line interface, see appendix E on page 74.⁸

```

\documentclass{article}
\title{Bibliography}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{ }

```

⁷Again, you must pay attention to the reference types and required fields. This is probably the most onerous part of including a bibliography.

⁸If you compile your document using the command line, you must use the `bibtex <document name>` command in order to produce the bibliography itself.


```

\tableofcontents{}
\section{Introduction}
\label{inTroduction}
This section cites \cite{goossens04} and \cite{oetiker15}.
\section{Body}
\label{Body}
This section cites \cite{kottwitz15} and \cite{pakin15}.
\section{Conclusion}
This section contains no citations.
\label{Conclusion}
\bibliographystyle{IEEEtranS}
\bibliography{tut}
\addcontentsline{toc}{section}{References}
\nocite{*}
\end{document}

```

Exercise: Bib_TE_X provides many different kinds of styles. Research three of them.

Exercise: Every style has required fields and optional fields. For the three styles you researched, list the required fields and the optional fields.

4.7 Indices

Commands

- package `imakeidx`
- `makeindex`
- `index`
- `printindex`
- `addcontentsline`

Any paper of a reasonable size should have an index. Fortunately, indices in L^AT_EX 2_ε are easy if a bit tedious. The tedious part comes from marking all the words to be indexed. The rest is pretty easy.

First, in the preamble, you must use an indexing package, the current version of which is `imakeidx` (perhaps for *improved* `makeidx`) followed by the `makeindex` command. At the end of your paper, include the `printindex` command, optionally followed by the `addcontentsline` if you wish to have an entry in our table of contents for the index.

Then, for every term you wish to index, you must make a reference using the `index` command. L^AT_EX 2_ε produces the index automatically.⁹

```

\documentclass{article}
\usepackage{imakeidx}
\makeindex{}
\title{Indices}

```

⁹If you compile your document using the command line, you must use the `makeindex <document name>` command in order to produce the index itself.

```

\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\tableofcontents{}
\section{Introduction}
\label{inTroduction}
This section contains the indexed word introduction.\index{introduction}
\section{Body}
\label{Body}
\paragraph{}This paper concerns the structure\index{structure} of a paper.
\paragraph{}Papers generally consist of an introduction\index{introduction},
    body\index{body|}, and conclusion\index{conclusion}.
\paragraph{}This is the body\index{body} of a paper.
\section{Conclusion}
This section contains the conclusion\index{conclusion}.
\label{Conclusion}
\printindex{}
\addcontentsline{toc}{section}{Index}
\end{document}

```

Exercise: Find the documentation to the `imakeidx` package and read through it.

Exercise: How can you generate multiple indices, perhaps one for people and one for commands?

4.8 Includes

Commands

- `include`
- `input`

It is convenient to break a paper of more than a few sections into several parts. When you write the paper, you can work on each part separately, and then combine them into the finished paper, part by part, as convenient. Using the command `include` and `input` requires each part to be placed into a separate document. Your “main” document will then have a series of include or input commands. When you compile the main document, each part will be included at the appropriate place. Note that the included documents may not contain any preamble commands — all your preamble commands must be placed in the preamble, which will be in your main document.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name includemain.tex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass{article}
\title{This is My Title}
\author{Charles Carter}
\date{\today{}}

```

```

\begin{document}
  \maketitle{}
  \input{includeintro}
  \input{includebody}
  \input{includeconc}
\end{document}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name includeintro.tex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section*{Introduction}
This is the introduction.
The file is named includeintro.tex.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name includebody.tex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section*{Body}
This is the body.
The file is named includebody.tex.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name includeconc.tex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section*{Conclusion}
This is the conclusion.
The file is named includeconc.tex.

```

Illustration

Introduction

This is the introduction. The file is named includeintro.tex.

Body

This is the body. The file is named includebody.tex.

Conclusion

This is the conclusion. The file is named includeconc.tex.

Exercise: Using any longish document you have, create a main document with a series of include statements, and place the content of your document into a series of separate documents, each with their own name.

\LaTeX 2 $_{\epsilon}$ is *Well Documented*

5 Lists

5.1 Verbatim

- verbatim environment
- verbatim

Perhaps the simplest and easiest way to make a list is by using the `verbatim` environment. Everything in the environment is printed as is, that is, $\text{\LaTeX} 2_{\epsilon}$ commands are not evaluated.

```
\documentclass{article}
  \title{Verbtim Environment}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Verbatim Environment}
  \begin{verbatim}
    \texttt{one}
    \textit{two}
    \textsf{three}
    \huge{aye}
    \tiny{bee}
    \section{see}
  \end{verbatim}
\end{document}
```

Exercise: The only command not permitted in the `verbatim` environment is the `\end{verbatim}` command. The reason is that this command ends the verbatim environment. If you examine the source of this tutorial, you will see that I resorted to a trick to include this command. As an exercise, can you figure out the trick?

5.2 Unnnmbered Lists

- itemize
- item

$\text{\LaTeX} 2_{\epsilon}$ unnumbered lists are very easy. Simply enclose the list in the `itemize` environment, and place each list item as an argument to the `item` command.

- Green

- Eggs
- and
- Ham

```
\documentclass{article}
  \title{Unnumbered List}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \begin{itemize}
    \item{Green}
    \item{Eggs}
    \item{and}
    \item{Ham}
  \end{itemize}
\end{document}
```

Exercise: Make your own unnumbered list.

5.3 Numbered Lists

- enumerate
 - item

L^AT_EX 2_ε numbered lists are very easy. Simply enclose the list in the `enumerate` environment, and place each list item as an argument to the `item` command.

1. Washington
2. Adams
3. Jefferson
4. Madison

```
\documentclass{article}
  \title{Unnumbered List}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \begin{enumerate}
    \item{Washington}
    \item{Adams}
    \item{Jefferson}
  \end{enumerate}
\end{document}
```

```

\item{Madison}
\end{enumerate}
\end{document}

```

Exercise: Make your own numbered list.

5.4 Dictionary Lists

- description
- item

$\text{\LaTeX} 2_{\epsilon}$ dictionary lists are very easy. Simply enclose the list in the `description` environment, place the term in square brackets (`[]`), and place each list item as an argument to the `item` command.

C a procedural language

Java an object oriented language

Lisp a functional language

JavaScript an event driven language

Erlang a concurrent language

```

\documentclass{article}
\title{Unnumbered List}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\begin{description}
\item[C]{a procedural language}
\item[Java]{an object oriented language}
\item[Lisp]{a functional language}
\item[JavaScript]{an event driven language}
\item[Erlang]{a concurrent language}
\end{description}
\end{document}

```

Exercise: Make your own dictionary list.

5.5 Nested Lists

And of course, $\text{\LaTeX} 2_{\epsilon}$ lists can be nested. Notice how “smart” the numbered lists are — the number style depends on the heading level.

1. 17th century
 - (a) Thirty Years War
 - Gustavus Adolphus** Sweden
 - Wallenstein** Hapsburg Austria
 - Turenne** France
 - (b) War of the Grand Alliance
 - William III** Dutch Republic
 - Eugene** Hapsburg Austria
 - Vauban** France
2. 18th century
 - (a) War of the Spanish Succession
 - Eugene** Hapsburg Austria
 - Malbourough** England
 - Villeroi** France
 - Maximilian II** Bavaria
 - (b) Seven Years War
 - Frederick II** Prussia
 - Daun** Hapsburg Austria
 - Maximilian III** Bavaria
 - Clive** Great Britain
3. 19th century
 - (a) War of the Sixth Coalition
 - Napoleon I** France
 - Blucher** Prussia
 - Bennigsen** Russia
 - Schwarenborg** Austria
 - (b) Franco-Prussian War
 - Napoleon III** France
 - Moltke** Prussia
4. 20th century
 - (a) World War 1
 - Haig** United Kingdom
 - Foch** France
 - Hindenburg** Germany
 - Pershing** United States
 - (b) World War 2
 - McArthur** United States
 - Montgomery** United Kingdom
 - Manstein** Germany
 - Rossokovsky** Soviet Union


```

\documentclass{article}
  \title{Unnumbered List}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \begin{enumerate}
    \item{17th century}
    \begin{enumerate}
      \item{Thirty Years War}
      \begin{description}
        \item[Gustavus Adolphus]{Sweden}
        \item[Wallenstein]{Hapsburg Austria}
        \item[Turenne]{France}
      \end{description}
      \item{War of the Grand Alliance}
      \begin{description}
        \item[William III]{Dutch Republic}
        \item[Eugene]{Hapsburg Austria}
        \item[Vauban]{France}
      \end{description}
    \end{enumerate}
    \item{18th century}
    \begin{enumerate}
      \item{War of the Spanish Succession}
      \begin{description}
        \item[Eugene]{Hapsburg Austria}
        \item[Malbourough]{England}
        \item[Villeroi]{France}
        \item[Maximilian II]{Bavaria}
      \end{description}
      \item{Seven Years War}
      \begin{description}
        \item[Frederick II]{Prussia}
        \item[Daun]{Hapsburg Austria}
        \item[Maximilian III]{Bavaria}
        \item[Clive]{Great Britain}
      \end{description}
    \end{enumerate}
    \item{19th century}
    \begin{enumerate}
      \item{War of the Sixth Coalition}
      \begin{description}
        \item[Napoleon I]{ }
        \item[Blucher]{Prussia}
        \item[Bennigsen]{Russia}
        \item[Schwarenborg]{Austria}
      \end{description}
      \item{Franco-Prussian War}
      \begin{description}
        \item[Napoleon III]{ }France
      \end{description}
    \end{enumerate}
  \end{enumerate}

```

```

        \item[Moltke]{Prussia}
    \end{description}
\end{enumerate}
\item{20th century}
\begin{enumerate}
    \item{World War 1}
    \begin{description}
        \item[Haig]{United Kingdom}
        \item[Foch]{France}
    \item[Hindenburg]{Germany}
        \item[Pershing]{United States}
    \end{description}
    \item{World War 2}
    \begin{description}
        \item[McArthur]{United States}
        \item[Montgomery]{United Kingdom}
        \item[Manstein]{Germany}
        \item[Rossokovsky]{Soviet Union}
    \end{description}
    \end{enumerate}
\end{enumerate}
\end{document}

```

Exercise: Make your own nested list to at least two levels.

5.6 Listings Package

- listings package
- lstset
- lstlisting

If you really want to customize your listing, use the `listings` package. Import the package as usual in the preamble with `\usepackage{listings}`. Use `lstset` to specify your list settings. This can be either in the preamble or (if you have various lists with different settings) in the body of the document just prior to the list. Use `lstlisting` to set the list. See section 5.7 on page 42 and section 5.8 on page 44 for examples.

Here are some of the parameters allowed for `lstset`.

```

\lstset{ %
backgroundcolor=\color{white},    % choose the background color;
    you must add \usepackage{color} or \usepackage{xcolor}
basicstyle=\footnotesize,        % the size of the fonts that are used for the code
breakatwhitespace=false,         % sets if automatic breaks should only happen at whitespace
breaklines=true,                  % sets automatic line breaking
captionpos=b,                     % sets the caption-position to bottom
frame=single,                     % adds a frame around the code
keepspace=true,                   % keeps spaces in text, useful for keeping indentation
}

```

```

of code (possibly needs columns=flexible)
keywordstyle=\color{blue},      % keyword style
language=Octave,                % the language of the code
numbers=left,                   % where to put the line-numbers;
    possible values are (none, left, right)
numbersep=5pt,                  % how far the line-numbers are from the code
numberstyle=\tiny\color{gray},  % the style that is used for the line-numbers
rulecolor=\color{black},        % if not set, the frame-color may be changed on line-breaks
    within not-black text (e.g. comments (green here))
showspaces=false,               % show spaces everywhere adding particular underscores;
    it overrides 'showstringspaces'
showstringspaces=false,         % underline spaces within strings only
showtabs=false,                 % show tabs within strings adding particular underscores
tabsize=2,                      % sets default tabsize to 2 spaces
}

```

Exercise: Find the documentation for the listings package and look through it.

5.7 Listings Version 1

The programming language Lisp was created in the late 1950s as a LISP Processing language — it might not be inappropriate to use Lisp to illustrate lists. Here is the listing.

```

1  ;;; add-test.lisp
2  (print "This is add-lisp. Evaluate (start-test) to start the test
   .")
3
4  (defun start-test ()
5    (defparameter number-of-questions 10)
6    (defparameter number-correct 0)
7    (defparameter question-counter 1)
8    (format t "Starting the addition test, you have ~a questions.~%"
   " number-of-questions)
9    (run-test))
10
11 (defun addition-problem ()
12   (let* ((a (random 11))
13         (b (random 11))
14         (c (+ a b))
15         (d (read (format t "What is ~a + ~a? " a b))))
16     (cond ((= c d)
17            (format t "Correct~%")
18            1)
19          (t (format t "The answer is ~a~%" c)
20             0))))
21
22 (defun run-test ()
23   (cond
24     ((zerop number-of-questions)
25      (format t "You got ~a correct and made a ~a.~%"
26              number-correct (* 100 (/ number-correct 10.0))))
27     (t (format t "Question ~a. " question-counter)
28         (decf number-of-questions)
29         (incf number-correct (addition-problem))
30         (incf question-counter)
31         (run-test))))

```

Here is the code.

```

\documentclass{article}
  \title{Lists, Version 1}
  \author{Charles Carter}
  \date{\today{}}
  \usepackage{listings}
  \usepackage{xcolor}
\begin{document}
  \maketitle{}
\lstset{language=Lisp,numbers=left,keepspaces=true,
  basicstyle=\small,numberstyle=\tiny,
  showstringspaces=false,breaklines=true}
\begin{lstlisting}
;;;add-test.lisp
(print "This is add-lisp. Evaluate (start-test) to start the test.")

(defun start-test ()
  (defparameter number-of-questions 10)
  (defparameter number-correct 0)
  (defparameter question-counter 1)
  (format t "Starting the addition test, you have ~a questions.~%"
    number-of-questions)
  (run-test))

(defun addition-problem ()
  (let* ((a (random 11))
        (b (random 11))
        (c (+ a b))
        (d (read (format t "What is ~a + ~a? " a b)))))
    (cond ((= c d)
      (format t "Correct~%"
        1)
      (t (format t "The answer is ~a~%" c)
        0))))))

(defun run-test ()
  (cond
    ((zerop number-of-questions)
      (format t "You got ~a correct and made a ~a.~%"
        number-correct (* 100 (/ number-correct 10.0))))
    (t (format t "Question ~a. " question-counter)
      (decf number-of-questions)
      (incf number-correct (addition-problem))
      (incf question-counter)
      (run-test))))
\end{lstlisting}
\end{document}

```

Exercise: What programming languages does the listings package know about?

5.8 Listings Version 2

Here is a listing you have see before. To create this, use this code. The language is set to “TeX”, the dialect is “LaTeX”. The line numbers are on the right. Keywords are blue bolded. Comments are colored teal. Note the use of packate *xcolor*, which I will cover in section 7.6 on page 60.

```
\lstset{
  language=[LaTeX]TeX
  numbers=right
  keywordstyle=\color{blue}\textbf
  keepspaces=true
  basicstyle=\footnotesize
  numberstyle=\scriptsize
  showstringspaces=false
  breaklines=true
  commentstyle=\color{teal}}
\begin{lstlisting}
  %code goes here
\end{lstlisting}
```

Here is what the listing looks like with these settings.

```
\documentclass{article}
%this is the preamble
%author and title information
\usepackage{xcolor}
\usepackage{listings}
\title{Font Sizes}
\author{Charles Carter}
\date{\today{}}
\begin{document}
%create the title
\maketitle{}
%create the table of contents
\tableofcontents{}
\section{Introduction}
\label{Introduction}
\section{Body}
\label{Body}
%start with a normal sized font
\normalsize{}\paragraph{}This paragraph has a normalsize font size
.
%font sizes in forder from smallest to largest
\tiny{}\paragraph{}This paragraph has a tiny font size.
\scriptsize{}\paragraph{}This paragraph has a scriptsize font size
.
\footnotesize{}\paragraph{}This paragraph has a footnotesize font
size.
\small{}\paragraph{}This paragraph has a small font size.
\normalsize{}\paragraph{}This paragraph has a normalsize font size
.
\large{}\paragraph{}This paragraph has a large font size.
\Large{}\paragraph{}This paragraph has a Large font size.
\LARGE{}\paragraph{}This paragraph has a LARGE font size.
\huge{}\paragraph{}This paragraph has a huge font size.
\Huge{}\paragraph{}This paragraph has a Huge font size.
%end with a normal sized font
\normalsize{}\paragraph{}This paragraph has a normalsize font size
.
```

<code>\section{Conclusion}</code>	33
<code>\label{Conclusion}</code>	34
<code>\end{document}</code>	35

Here is the code for the listing.

```

\documentclass{article}
  %this is the preamble
  %author and title information
  \usepackage{xcolor}
  \usepackage{listings}
  \lstset{language=[LaTeX]TeX,numbers=right,keywordstyle=\color{blue}\textbf,
    keepspaces=true,basicstyle=\footnotesize,numberstyle=\scriptsize,
    kshowstringspaces=false,breaklines=true,commentstyle=\color{teal}}
  \title{Font Sizes}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  %create the title
  \maketitle{}
  %create the table of contents
  \tableofcontents{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \begin{lstlisting}
    %start with a normal sized font
    \normalsize{}\paragraph{}This paragraph has a normalsize font size.
    %font sizes in forder from smallest to largest
    \tiny{}\paragraph{}This paragraph has a tiny font size.
    \scriptsize{}\paragraph{}This paragraph has a scriptsize font size.
    \footnotesize{}\paragraph{}This paragraph has a footnotesize font size.
    \small{}\paragraph{}This paragraph has a small font size.
    \normalsize{}\paragraph{}This paragraph has a normalsize font size.
    \large{}\paragraph{}This paragraph has a large font size.
    \Large{}\paragraph{}This paragraph has a Large font size.
    \LARGE{}\paragraph{}This paragraph has a LARGE font size.
    \huge{}\paragraph{}This paragraph has a huge font size.
    \Huge{}\paragraph{}This paragraph has a Huge font size.
    %end with a normal sized font
    \normalsize{}\paragraph{}This paragraph has a normalsize font size.
  \end{lstlisting}
  \section{Conclusion}
  \label{Conclusion}
\end{document}

```

Exercise: Try listing your own favorite programming language.

$\text{\LaTeX 2}_{\varepsilon}$ is *Stable*

6 Tables

Unfortunately, tables in L^AT_EX 2_ε tend to be a bit cumbersome. Tables are specified row by row, and require meticulous attention to detail. If sufficient care is taken, composing tables is not very difficult. Fortunately, simple tables are not too hard.

6.1 Verbatim table

- verbatim
- verbatim environment

As with lists, this simplest, most brain-dead way to make a table is by using the `verbatim` environment. The hardest part is making sure that your cells align properly using correct spacing. Hint: use a monospace font in your text editor rather than a variable spaced font.

Abigail	Derek	Gladiola
Brenda	Edgar	Hibiscus
Claudia	Frank	Impatiens

```
\documentclass{article}
  \title{Verbatim Tables}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \begin{verbatim}
Abigail    Derek      Gladiola
Brenda     Edgar      Hibiscus
Claudia    Frank      Impatiens
  \end{verbatim}
\end{document}
```

Exercise: Create your own verbatim table.

6.2 Simple table

- tabular environment
- l (column specifier)
- c (column specifier)
- r (column specifier)

- `\\` (row terminator)
- `&` (cell separator)

A simple $\text{\LaTeX 2}_{\epsilon}$ table is created within a `tabular` environment. The column specifications are passed as a parameter to the environment — `r` (right), `c` (center), `l` (left). Table rows are terminated by two backslashes (`\\`). Cells are separated by an ampersand (`&`).

Ann	Derek	Gladiola
Brenda	Ed	Hibiscus
Claudette	Frederick	Impatiens

```
\documentclass{article}
\title{Simple Tables}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\begin{tabular}{l c r}
Ann      & Derek      & Gladiola \\
Brenda   & Ed         & Hibiscus \\
Claudette & Frederick  & Impatiens \\
\end{tabular}
\end{document}
```

Exercise: Make your own simple table.

6.3 Row and column lines

- `|`
- `hline`

Table rules are specified by the vertical bar (`|`) and by `hline`. Column rules are set in the parameter of the column specifications passed to the `tabular` environment. Row rules are created by the `hline` command. Both of these can be doubled to produce double rules.

Girls	Boys	Flowers	Animals
Abigail	Derek	Gladiola	Jackel
Brenda	Edgar	Hibiscus	Koala
Claudia	Frank	Impatiens	Lynx

```
\documentclass{article}
\title{Table Rules}
\author{Charles Carter}
```

```

\date{\today{}}
\begin{document}
\maketitle{}
\begin{tabular}{| l | r || l | r |}
\hline
Girls & Boys & Flowers & Animals\\
\hline
\hline
Abigail & Derek & Gladiola & Jackel \\
\hline
Brenda & Edgar & Hibiscus & Koala \\
\hline
Claudia & Frank & Impatiens & Lynx \\
\hline
\end{tabular}
\end{document}

```

Exercise: Create your own simple table with vertical and horizontal rules.

6.4 Column spacing

- p (column specifier)

The column specifiers l (left), c (center), and r (right) mostly do what you want, but their capacity is limited and frequently you need to use the column specifier p (paragraph) to make the content of the cells behave. p takes as a parameter a length recognized by L^AT_EX 2_ε, frequently a common English measure expressed in inches, for example, p{2in}.

```

\documentclass{article}
\title{P Columns}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\begin{tabular}{p{1in} p{4in}}
\hline{}
C & C is a general-purpose, imperative computer programming
language, supporting structured programming, lexical variable
scope and recursion, while a static type system prevents many
unintended operations. \\
Java & Java is a general-purpose computer programming language
that is concurrent, class-based, object-oriented, and
specifically designed to have as few implementation
dependencies as possible. It is intended to let application
developers "write once, run anywhere" meaning that compiled

```

C	C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations.
Java	Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers “write once, run anywhere” meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.
Common Lisp	Common Lisp (historically, LISP) is a family of computer programming languages with a long history and a distinctive, fully parenthesized prefix notation. Originally specified in 1958, Lisp is the second-oldest high-level programming language in widespread use today. Only Fortran is older, by one year.
JavaScript	JavaScript is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification.
Erlang	Erlang is a programming language designed for developing robust systems of programs that can be distributed among different computers in a network. Named for the Danish mathematician Agner Krarup Erlang, the language was developed by the Ericsson Computer Sciences Lab to build software for its own telecommunication products.

Table 2: P columns

```

Java code can run on all platforms that support Java without
the need for recompilation. \\
Common Lisp & Common Lisp (historically, LISP) is a family
of computer programming languages with a long history and a
distinctive, fully parenthesized prefix notation. Originally
specified in 1958, Lisp is the second-oldest high-level
programming language in widespread use today. Only Fortran
is older, by one year. \\
JavaScript & JavaScript is a high-level, dynamic, untyped,
and interpreted programming language. It has been
standardized in the ECMAScript language specification. \\
Erlang & Erlang is a programming language designed for
developing robust systems of programs that can be
distributed among different computers in a network.
Named for the Danish mathematician Agner Krarup Erlang,
the language was developed by the Ericsson Computer
Sciences Lab to build software for its own tele
communication products.  \\
\hline{}
\end{tabular}
\end{document}

```

Table 3: Names - Caption Above

A	Akin	Allie
B	Bobby	Bonita
C	Chad	Carly

D	Date	Dogwood
E	Eggplant	Elm
F	Fig	Frasier Fir

Table 4: Plants - Caption Below

Exercise: Create your own table with paragraph spacing.

6.5 Captions and labels

- table environment
- caption
- label
- listoftables

The *table environment* is created with the `\begin{table}` command, and ended with the `\end{table}` command. Table environments contain tables (inside the *tabular environment*), and optional labels and captions. A caption is created with the `\caption` command. Notice that in the example below the label command is placed after the caption command. Table 3 is on 51 and table 4 is on page 51.

The `\listoftables` command, which should be placed just after the `\tableofcontents` command, generates a list of the tables in the document. For an example, see the table of contents of this tutorial and the listing below.

```
\documentclass{article}
\title{Table Captions and Labels}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\listoftables{}
Reference to table \ref{Plants} on page \pageref{Plants}.
\begin{table}
\caption{Names - Caption Above}
\label{Names}
\begin{tabular}{r | l l}
\hline
A & Akin & Allie \\
B & Bobby & Bonita \\
C & Chad & Carly
\end{tabular}
\end{table}
```

```

        \hline
    \end{tabular}
\end{table}
\newpage
\begin{table}
    \begin{tabular}{r | l l}
        \hline
        D & Date & Dogwood \\
        E & Eggplant & Elm \\
        F & Fig & Frasier Fir \\
        \hline
    \end{tabular}
    \caption{Plants - Caption Below }
    \label{Plants}
\end{table}
Reference to table \ref{Names} on page \pageref{Names}.
\end{document}

```

Exercise: You can refer to a table, and the page number of the table, by using the `ref` and `pageref` commands. See section 4.1 on page 26.

Exercise: How would you center a table horizontally on the page?

6.6 Multicells

- multicolumn
- multirow package
- multirow

L^AT_EX 2_ε tables can be nested, but for multi-column and multi-row spans, there are separate commands that are much easier to use. Multicolumn cells are created by the `multicolumn{num-of-cols}{col-alignment}{cell-content}` as illustrated in table 5 on page 53.

```

\documentclass{article}
\title{Multicolumns}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\begin{table}
    \begin{tabular}{| l | p{2.5in} | c |}
        \hline
        \multicolumn{3}{| c |}{Measures of center} \\
        \hline
        mean &
    \end{tabular}
\end{table}

```

Measures of center		
mean	the sum of all values divided by the number of values	$\frac{\sum_{i=1}^n x_i}{n}$
median	the middle value in a vector of sorted value	$\frac{\text{length}(V)+1}{n} \text{sorted}(V)$
mode	the most frequent value or values if more than one	(no formula)
Measures of dispersion		
variance	the sum of the difference between each value of the vector and the mean of the vector divided by the number of values in the vector	$\sum_{i=1}^n \frac{V_i - \mu}{n}$
standard deviation	the square root of the sum of the difference between each value of the vector and the mean of the vector divided by the number of values in the vector	$\sqrt{\sum_{i=1}^n \frac{V_i - \mu}{n}}$
median absolute deviation	the average distance between each data value and the mean	$\frac{\sum_{i=1}^n V_i - \mu }{n}$

Table 5: Multicolumn Example

```

        the sum of all values divided by the number of values &
        $\frac{\sum_{i=1}^n x_i}{n}$ \\
\hline
median &
        the middle value in a vector of sorted value &
        $\frac{\text{length}(V) + 1}{n} | \text{sorted}(V)$ \\
\hline
mode &
        the most frequent value or values if more than one
        & (no formula) \\
\hline
\multicolumn{3}{| c |}{Measures of dispersion} \\
\hline
variance &
        the sum of the difference between each value of the vector
        and the mean of the vector divided by the number of values
        in the vector &
        $\sum_{i=1}^n \frac{V_i - \mu}{n}$ \\
\hline
standard deviation &
        the square root of the sum of the difference between each value
        of the vector and the mean of the vector divided by the number
        of values in the vector &
        $\sqrt{\sum_{i=1}^n \frac{V_i - \mu}{n}}$ \\
\hline
median absolute deviation &
        the average distance between each data value and the mean &
        $\frac{\sum_{i=1}^n |V_i - \mu|}{n}$ \\

```

Confederate songs	Dixie Yellow Rose of Texas Bonnie Blue Flag When Johnny Comes Marching Home
Union songs	Battle Hymn of the Republic Marching Through Georgia Tramp, Tramp, Tramp Battle Cry of Freedom
Civilian songs	Aura Lee Tenting Tonight Lorena Weeping Sad and Lonely

Table 6: Multirow Example

```

\hline
\end{tabular}
\caption{Multicolumn Example}
\label{multicol}
\end{table}
\end{document}

```

Tables with multiple rows for a column need the package *multirow*, import as usual with `\usepackage{multirow}`. The syntax is `\multirow{num-of-rows}{row-width}{content}`. In the table, make sure that the appropriate rows for the multirow structure are undefined. Here is an example in table 6 on page 54.

```

\documentclass{article}
\title{Multirows}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\begin{table}
\begin{tabular}{| l | l |}
\hline
\multirow{5}{*}{Confederate songs} & Dixie \\
& & Yellow Rose of Texas \\
& & Bonnie Blue Flag \\
& & When Johnny Comes Marching Home \\
\hline
\multirow{5}{*}{Union songs} & Battle Hymn of the Republic \\
& & Marching Through Georgia \\
& & Tramp, Tramp, Tramp \\
& & Battle Cry of Freedom \\
\hline
\multirow{5}{*}{Civilian songs} & Aura Lee \\
& & Tenting Tonight \\
& & Lorena \\
& & Weeping Sad and Lonely \\
\hline
\end{tabular}
\end{table}

```

```

\end{tabular}
\caption{Multirow Example}
\label{multirow}
\end{table}
\end{document}

```

Exercise: Tables with multiple row spans and multiple column spans require much thought and debugging. Try your own very simple example

6.7 Long tables

- longtable package
- longtable

What do you do if the length of your table exceed the page length? Use the `longtable` command after first importing the *longtable package* the usual way with `\usepackage{longtable}`. The `longtable` also accepts the `caption` and `label` commands. See table 7 on page 55.

Table 7: Longtable example

State	Capital	Population
California	Sacramento	37,000,000
Texas	Austin	25,000,000
New York	Albany	19,000,000
Florida	Tallahassee	18,000,000

```

\documentclass{article}
\usepackage{longtable}
\title{Long Tables}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\listoftables{}
\begin{longtable}{|l|c|r|}
\caption{Longtable example}
\label{longtable}
\hline
State & Capital & Population \\
\hline
\hline
California & Sacramento & 37,000,000 \\
\hline
Texas & Austin & 25,000,000 \\

```



```

\hline
New York & Albany & 19,000,000 \\\
\hline
Florida & Tallahassee & 18,000,000 \\\
\hline
\end{longtable}
\end{document}

```

\LaTeX 2 $_{\epsilon}$ is *Easy*

7 Images

7.1 Image types

-

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

7.2 Including graphics

-

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

7.3 Image manipulation

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

7.4

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

7.5 Captions and labels

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

7.6 Colors

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

$\text{\LaTeX 2}_{\epsilon}$ is *Free*

8 Advanced Topics

This part of the tutorial covers features that, while not really advanced, illustrate useful applications of $\text{\LaTeX} 2_{\epsilon}$. Section 8.1 shows how to create new $\text{\LaTeX} 2_{\epsilon}$ commands that can be useful in automating certain tasks. Section 8.2 addresses templates that allow authors to conform to certain style conventions, such as MLA, APA, and IEEE. Section 8.4 concerns the reproduction of this document using the source listing. Sections 8.5 and 8.7 extend the $\text{\LaTeX} 2_{\epsilon}$ document classes to books and presentations. Section 8.3 explores a very useful package for adjusting page margins. Finally, section 8.8 discusses certain files $\text{\LaTeX} 2_{\epsilon}$ produces during the compilation process.

8.1 New commands

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}
```

Exercise:

Exercise:

8.2 Templates

•

```
\documentclass{article}
  \title{This is My Title}
  \author{Charles Carter}
  \date{\today{}}
\begin{document}
  \maketitle{}
```



```

\section{Introduction}
\label{Introduction}
\section{Body}
\label{Body}
\section{Conclusion}
\label{Conclusion}
\end{document}

```

Exercise:

Exercise:

8.3 Geometry

•

```

\documentclass{article}
\title{This is My Title}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\maketitle{}
\section{Introduction}
\label{Introduction}
\section{Body}
\label{Body}
\section{Conclusion}
\label{Conclusion}
\end{document}

```

Exercise:

Exercise:

8.4 Source listing

•

```

\documentclass{article}
\title{This is My Title}
\author{Charles Carter}
\date{\today{}}

```

```

\begin{document}
  \maketitle{}
  \section{Introduction}
  \label{Introduction}
  \section{Body}
  \label{Body}
  \section{Conclusion}
  \label{Conclusion}
\end{document}

```

Exercise:

Exercise:

8.5 Presentations

- beamer documentclass
- frame
- frametitle

$\text{\LaTeX} 2_{\epsilon}$ creates excellent slide presentations. The following example illustrates the use of the *beamer* package. Beamer has a great deal of flexibility, and it has the advantage that you can recycle your $\text{\LaTeX} 2_{\epsilon}$ code that you used for your document to create your presentation. This short tutorial cannot cover beamer at all but it can suggest how beamer can be used to produce a presentation.

```

\documentclass{beamer}
  \title{Beamer Example}
  \author{Charles Carter}
  \date{\today}
\begin{document}
  \begin{frame}
    \maketitle
  \end{frame}
  \begin{frame}
    \frametitle{Table of Contents}
    \tableofcontents{}
  \end{frame}
  \section{Lists}
  \begin{frame}
    \frametitle{Lists}
    This section covers lists, unnumbered, numbered, and dictionary lists.
  \end{frame}
  \subsection{Unnumbered lists}
  \begin{frame}

```

```

\frametitle{Unnumbered lists}
\begin{itemize}
  \item{apples}
  \item{bananas}
  \item{cranberries}
  \item{dates}
\end{itemize}
\end{frame}
\subsection{Paused unnumbered lists}
\begin{frame}
  \frametitle{Unnumbered lists with pause}
  \begin{itemize}
    \item{endive}
    \pause
    \item{figs}
    \pause
    \item{garlic}
    \pause
    \item{honeydew melon}
  \end{itemize}
\end{frame}
\subsection{Numbered lists}
\begin{frame}
  \frametitle{Numbered lists}
  \begin{enumerate}
    \item{iceberg lettuce}
    \item{Jerusalem artichoke}
    \item{kiwi}
    \item{lime}
  \end{enumerate}
\end{frame}
\subsection{Paused numbered lists}
\begin{frame}
  \frametitle{Numbered lists with pause}
  \begin{enumerate}
    \item{mango}
    \pause
    \item{nectarine}
    \pause
    \item{olive}
    \pause
    \item{paw paw}
  \end{enumerate}
\end{frame}
\subsection{Dictionary lists}
\begin{frame}
  \frametitle{Dictionary lists}
  \begin{description}
    \item[quince]{cooked in pies}
    \item[radish]{raw in salads}
    \item[strawberry]{fresh as a dessert}
  \end{description}
\end{frame}

```

```

        \item[turnip]{raw in salad or cooked with greens}
    \end{description}
\end{frame}
\subsection{Paused dictionary lists}
\begin{frame}
    \frametitle{Numbered lists with pause}
    \begin{description}
        \item[ugli fruit]{fresh at breakfast}
        \pause
        \item[watermelon]{at picnics}
        \pause
        \item[yam]{baked with turkey or ham}
        \pause
        \item[zucchini]{boiled or stir fried}
    \end{description}
\end{frame}
\section{Tables, blocks, and verbatim}
\begin{frame}
    \frametitle{Tables, blocks, and verbatim}
    This section covers tables, blocks, and listings using the \texttt{verbatim} environm
\end{frame}
\subsection{Tables}
\begin{frame}
    \frametitle{Tables}
    \begin{tabular}{|l|l|l|l|}
        \hline
        Beef & steak & pot roast & \\
        \hline
        Pork & bacon & spare ribs & \\
        \hline
        Fowl & fried chicken & chicken livers & \\
        \hline
    \end{tabular}
\end{frame}
\subsection{Blocks}
\begin{frame}
    \frametitle{Blocks}
    \begin{block}{Beef stroganoff}
        contains both milk and meat and is not kosher.
    \end{block}
    \begin{block}{Pork products}
        such as wieners and BBQ, are not kosher
    \end{block}
    \begin{block}{Shellfish}
        such as oysters, shrimp, and clams, is not kosher.
    \end{block}
\end{frame}
\subsection{Verbatim}
\begin{frame}[containsverbatim]
    \frametitle{Verbatim}
    Here is the listing from the previous slide.

```

```

\begin{verbatim}
\begin{block}{Beef stroganoff}
  contains both milk and meat and is not kosher.
\end{block}
\begin{block}{Pork products}
  such as wieners and BBQ, are not kosher
\end{block}
\begin{block}{Shellfish}
  such as oysters, shrimp, and clams, is not kosher.
\end{block}
\end{verbatim}
\end{frame}
\section{Afterword}
\begin{frame}
  \frametitle{Afterword}
  This just scratches the surface. Beamer has much, much more functionality than illust
\end{frame}
\end{document}

```

Exercise: Write your own, very simple, presentation, compile it, and present it.

8.6 Posters

- book documentclass

Poster text

poster code

Exercise: Write your own book. Just kidding! However, you can make the skeleton of a very simple, short book and you may even find this useful for a very long paper.

8.7 Books

- book documentclass

You can and should use $\text{\LaTeX} 2_{\epsilon}$ for writing book-sized documents. Use the documentclass `book`. Here is a very brief example. If you compile this and open it, you will have the skeleton of a “real book.” As always, the `book documentclass` contains a number of particular commands and environments, and if you go to the effort of authoring a book, you should also go to the effort of familiarizing yourself

with this document class. There are also other document classes, such as memoir, which are current more popular, and you may benefit from exploring these as well.

```
\documentclass[letterpaper]{book}
\title{My First Book}
\author{Charles Carter}
\date{\today{}}
\begin{document}
\frontmatter
\maketitle
\tableofcontents{}
\chapter{Preface}
This is the preface
\mainmatter
\chapter{A Story}
Let me tell you a story.
\chapter{Moral}
This is the moral of the story.
\appendix
\chapter{First Appendix}
Here is an appendix.
\backmatter
\chapter{Last note}
The last word.
\end{document}
```

Exercise: Write your own book. Just kidding! However, you can make the skeleton of a very simple, short book and you may even find this useful for a very long paper.

8.8 Auxiliary files

When you compile a L^AT_EX 2_ε document, the compilation process produces a number of auxiliary files. I have listed those produced by this tutorial below. You can open and read these in any text processor. You *should* open and read the log file. Often, when you have a warning or error in the compilation process, you can find valuable information here. Also, if you know what you are doing, you can edit an appropriate auxiliary file to fine tune your document. In any case, you should not regard these auxiliary files as a mystery, but should become comfortable with the process that produces them.

08/12/2016	01:10 PM	1,718	latex-tutorial.aux
08/04/2016	11:06 AM	2,388	latex-tutorial.bbl
08/04/2016	11:06 AM	662	latex-tutorial.blg
08/12/2016	01:10 PM	611	latex-tutorial.ent
08/12/2016	01:10 PM	3,732	latex-tutorial.idx
08/12/2016	01:10 PM	362	latex-tutorial.ilg
08/12/2016	01:10 PM	3,071	latex-tutorial.ind
08/12/2016	01:10 PM	0	latex-tutorial.lof
08/12/2016	01:10 PM	50,555	latex-tutorial.log
08/12/2016	01:10 PM	586	latex-tutorial.lot

08/12/2016	01:10 PM	3,988	latex-tutorial.out
08/12/2016	01:10 PM	626,626	latex-tutorial.pdf
08/10/2016	07:35 PM	6,768	latex-tutorial.tex
08/12/2016	01:10 PM	9,723	latex-tutorial.toc

Exercise: Read the log file, all of it. Understand as much of it as you can.

9 Conclusion

A Installing L^AT_EX 2_ε

L^AT_EX 2_ε formally consists of macros built on T_EX, a typesetting program created by Donald Knuth in the late 1970s. L^AT_EX 2_ε itself was created by Leslie Lamport in the early 1980s. Both T_EX and L^AT_EX 2_ε are open technologies, which means that anyone can implement them without restriction. Several implementations of L^AT_EX 2_ε are available, and opinions differ as to which one is “best.” I have tried several different implementations, and on Microsoft Windows, MiKTeX works best for me. Your mileage may vary.

This tutorial uses the MiKTeX implementation of L^AT_EX 2_ε. MiKTeX’s home (as of August, 2016) is <http://miktex.org/>. It can be downloaded by following the Download link (figure 1). Typically, accepting the usual defaults will lead to a successful installation. In particular, the installation process will add the executable to your path, so you can invoke it without jumping through any hoops.

If all goes well (and it should) you will see MiKTeX in your Windows start menu, as in figure 2. Note the program **TeXworks** — you can use this as your L^AT_EX 2_ε development environment if you want, see appendix C. You can also test your installation by executing the command `pdflatex --help` using your command line interface, see figure 3.



Figure 1: MiKTeX download

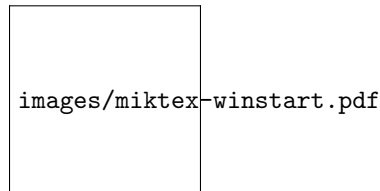


Figure 2: MiKTeX on Windows start menu

B L^AT_EX 2_ε Editors

A `tex` file is just ordinary plain text, without any special codes or binary commands. You write your L^AT_EX 2_ε document using an ordinary text editor. You can use your favorite text editor with no restrictions, from Windows Notepad (the most minimal text editor I know) on up.¹⁰

Notepad++ is a very popular, full featured, free text editor, and I recommend it for a large number of applications, including L^AT_EX 2_ε. You can find Notepad++ at <https://notepad-plus-plus.org/>. Notepad++ is just a text editor, so it doesn't come with any special bells and whistles.¹¹ Personally, I use `gvim` as my text editor, as it works for virtually everything, and works on Windows, Mac, Linux, and UNIX. However, `gvim` requires a substantial investment of time and effort to achieve proficiency, and I don't cover it here.

Typically, a new user of L^AT_EX 2_ε will choose to use an integrated development which combines a text editor and compiler (see appendix C) or an online resource (see appendix D). Use of a text editor requires separate editing and compilation cycles, which does offer some advantages, which is why some people choose to do it this way.

C Development Environments

MiKTeX comes with TeXwriter, which combines a text editor with a compiler. It's pretty minimal and crude, but that means that there's less to learn and less to break. Many other L^AT_EX 2_ε programs can be had, both free and commercial. Look

¹⁰But be careful of Notepad. Notepad by default saves documents with the `txt` extension, i.e., as a plain text file. You *must* save your L^AT_EX 2_ε file with a `tex` extension. Be sure to select the All Files option in the Save As box, and explicitly add `.tex` to the name of your file

¹¹If you really want to add a compilation feature to Notepad++, search the internet. It's not hard, but I don't cover it here.



Figure 3: CLI test

at the Wikipedia entry on $\text{\LaTeX} 2_{\epsilon}$ or \TeX editors. \TeX maker is also popular. The advantage to a development environment is that you can do your writing, editing, compilation, and publishing with one application. These also offer nice features, such as syntax coloring, autocomplete, and builtin help systems. The top end development environments also have point and click, drag and drop, features. This results in something close to a blend of a word processor (like Microsoft Word) and $\text{\LaTeX} 2_{\epsilon}$. Whether or not this is a good thing is a matter of opinion. You will find that people who use $\text{\LaTeX} 2_{\epsilon}$ professionally do not use these kinds of graphical interfaces — there are very good reasons for this avoidance that you will come to see if you use $\text{\LaTeX} 2_{\epsilon}$ frequently.

D Online $\text{\LaTeX} 2_{\epsilon}$

You can also use online $\text{\LaTeX} 2_{\epsilon}$ editors. There are many equation editors that translate equations into $\text{\LaTeX} 2_{\epsilon}$ source code. There are also full fledged online $\text{\LaTeX} 2_{\epsilon}$ systems, which allow collaboration between colleagues. Many (virtually all) are commercial. I have never used any of these so I can't offer an opinion about them. If you do not have an adequate computer (such as a Chromebook, perhaps), online document writing is an option.

E Command Line Execution

Both \TeX and $\text{\LaTeX} 2_{\epsilon}$ consist of executable computer programs that can be invoked and executed on the command line. If you automate your processes (perhaps you have to create lengthy reports several times a day or a week) you will write programs that call these executables to produce your documents. The three commands you will use most often are `pdflatex`, `makeindex`, and `bibtex`. Power users commonly do their work on the command line, and again, there are good reasons for this.

If the thought of the command line terrifies you, you never have to see it. However, it's always there lurking in the background. And sometimes, use of the command line is the only way to perform some tasks that your GUI program does not implement.

Notes

¹The difference between footnotes and endnotes is that footnotes go at the foot of the page where they appear, while endnotes appear at the end of the document.

²This is an endnote.

³The `addcontentsline` takes three parameters, where the line should be written, usually *toc*, the formatting to be used, usually *section*, and the name to be given to the entry, perhaps *Endnotes*.

Index

- $+$, 15
- $-$ (dash or subtraction), 15
- $\#$, 15
- $\$$, 15
- $\%$, 15
- $\&$, 15
- $\&$ (cell separator), 48
- \wedge , 15
- \wedge (caret or circumflex), 15
- \rightarrow , 15
- $_$ (underscore), 15
- \backslash , 15
- $\backslash\backslash$ (row terminator), 48
- $|$, 49
- $\{ \}$, 15

- addcontentsline, 20, 28, 32, 33
- align, 18
- amsmath, 17
- amsthm package, 20
- ast, 15
- author, 5

- beamer documentclass, 66
- begin/end document, 5
- bibliography, 32
- bibliographystyle, 32
- bibtex, 32
- Bmatrix, 22
- bmatrix, 22
- book documentclass, 69

- c (column specifier), 48
- caption, 52
- \cdot , 15
- cite, 32
- comments, 5
- creating an external BibTeX database, 30

- date, 5
- derivatives, 19
- description, 39
- documentclass, 5
- documentclass beamer, 66
- documentclass book, 69

- endnote, 28
- enumerate, 38
- equation, 17
- equation*, 17

- footnote, 28
- footnotesize, 11
- frac or div, 15
- frame, 66
- frametitle, 66

- gather, 18

- hline, 49
- Huge, 11
- huge, 11

- index, 33
- integrals, 19
- item, 37–39
- itemize, 37

- l (column specifier), 48
- label, 6, 27, 28, 52
- LARGE, 11
- Large, 11
- large, 11
- limits, 19
- lipsum, 8
- listings package, 42
- listoftables, 52
- longtable, 56
- longtable package, 56
- lstlisting, 42
- lstset, 42

- makeindex, 33
- maketitle, 5
- marginpar, 29
- multicolumn, 53
- multirow, 53
- multirow package, 53
- multline, 18

- newtheorem, 20
- nocite, 32
- normalmarginpar, 29
- normalsize, 11

- p (column specifier), 50
- package amsthm, 20
- package endnotes, 28
- package imakeidx, 33
- pageref, 27, 28

paragraph, 7	table environment, 52
plain text, 5	tableofcontents, 9
plus, 15	tabular environment, 48
pmatrix, 22	textbf, 10
printindex, 33	textit, 10
products, 19	textsc, 10
	textsf, 10
r (column specifier), 48	texttt, 10
raggedright, 29	theendnotes, 28
ref, 27, 28	times, 15
reversemarginpar, 29	tiny, 11
	title, 5
scriptsize, 11	
section, 6	underline, 10
small, 11	usepackage, 8
sqrt, 15	
subparagraph, 7	verbatim, 37, 48
subsection, 6	verbatim environment, 37, 48
subsubsection, 6	Vmatrix, 22
sums, 19	vmatrix, 22