Introduction to the LaTeX Document Preparation System

This document will teach you the fundamentals of LaTeX and help you start creating documents

Table of Contents

Overview

What is LaTeX?

LaTeX is a document preparation system. You create a file (using the text editor on your system) that contains your text and the appropriate LaTeX commands to format the text. This file can then be processed by the LaTeX software application to produce a file that can be printed or a file that can be viewed at your workstation (if your computer has this ability).

LaTeX was developed on UNIX[1] systems, but it is available for a variety of computing platforms including microcomputers. How the file is processed through LaTeX varies from system to system. However, the basic LaTeX formatting commands are the same. At Rice, several computer systems have LaTeX installed including Owlnet (Rice University's educational network).

Why would I use LaTeX?

If you want to create a document with any type of formatting, which includes selecting a font, centering lines, creating a table of contents, or including complex mathematical formulas, you need to use some type of formatting software. LaTeX can suit a wide variety of formatting needs from the very simple to the extremely complex.

How to Get Started

To learn enough LaTeX commands to create a very simple document, read the sections, *LaTeX Basics* and *Starting a Document*. When you decide you would like to take advantage of some more of LaTeX's formatting capabilities, read the remaining sections of this document. Examples of a letter, resume, and report can be found in the section *Starting a Document* to aid you as well. Additional information about some of the more complex features of LaTeX can be found in the books listed in the *For More Information* section.

You will also need to read the appropriate section on *System Specifics* so that you learn how to process your files with LaTeX on the computer you are using. If the computer platform you are using is not covered in this document, you will need to check the documentation that came with the software.

Accessing the Software

In order to use LaTeX, you need to be familiar with creating a file with a text editor on your system. Documents explaining how to use certain text editors on Owlnet can be obtained in the Consulting Center (103 Mudd Lab). You also need to make sure that you have access to the software. Access methods vary from system to system, so read the appropriate section below.

UNIX Note: Initially accounts on Owlnet are set up to have the correct path to access the LaTeX software. (Note: If you changed your path, you will need to make sure that

1. UNIX is a registered trademark of AT&T Bell Laboratories.

/**usr/local/bin** is in it to use LaTeX.) Using the editor of your choice, create a file that has the extension **.tex**. For example, your file might be named **paper.tex**.

LaTeX Basics

How to Give LaTeX Commands

The Backslash Character

LaTeX commands are preceded by the backslash character (\). LaTeX recognizes two varieties of control sequences:

- *control symbols*, which consist of a backslash character preceding a single character that is not a letter of the alphabet. The control symbol **%**, for example, tells LaTeX to print a percent sign. Without the backslash, the % character would be interpreted by LaTeX as a comment.
- *control words*, which are a backslash character preceding one or more letters of the alphabet. When LaTeX reads a character after the backslash that is not a letter of the alphabet, it knows that it has reached the end of the control word. Examples of control words are **center** (centers text on a line), **it** (puts words in italics), and **rm** (puts words in roman type).

LaTeX will not print the blank space or spaces you type after a control word (unlike control symbols) because it understands that a blank space or number or punctuation mark signals the end of a control word.

Groups

Groups make working with control words much easier. Whenever you enclose text in your file within curly braces, you create a separate environment for those words. Any control words you invoke inside a pair of these braces have no effect on the outside text. Instead of typing:

```
Words in \it italics \rm look better than those in \bf bold \rm.
```

to get the sentence

Words in *italics* look better than those in **bold**.

you can just enter:

```
Words in {\it italics} look better than those in {\bf bold}.
```

and receive the exact same sentence on the printed page.

Minimum Requirements

The two standard requirements for every LaTeX document are:

- defining the style
- indicating the beginning and ending of the text

Every LaTeX document that you create with your text editor must begin with a line like this one:

```
\documentstyle{article}
```

The **documentstyle** command calls a document template that determines document characteristics such as headers, margins, and so on. Other valid document styles besides article include report, letter, and book.

To indicate the body of the text, you must enclose it within two more control words, **\begin{document}** and **\end{document}**, so one of the smallest possible documents you can process with the LaTeX command would look something like this:

```
\documentstyle{article}
\begin{document}
The text of the document
\end{document}
```

All of the words, sentences, and paragraphs that you want in your document must lie between these **\begin** and **\end** control words.

Spaces, Line Breaks, and Paragraphs

When you are entering your text, your right margin can be ragged. LaTeX will print your text as pages of neatly spaced sentences and paragraphs. This is because LaTeX has its own rules regarding the use of spaces.

As you might expect, LaTeX recognizes a word as a group of characters separated from others by a blank space. The number of times you hit the space bar between words has no effect on how much space actually appears between them, it will only be one space. LaTeX identifies the ends of sentences by their punctuation and again separates them by its own system regardless of how many spaces you type. This is also true of the extra spaces at the end of your lines. Once your document has been processed by LaTeX, it will justify the right margin. To tell LaTeX that you have reached the end of a paragraph, enter a blank line after the paragraph's last line.

Controlling Line Breaks

If you want to prevent LaTeX from ending a line within a certain set of words, such as "Dr. Samuel Johnson," you can place a tilde character (~) between these words instead of a space, and LaTeX will keep those words together. To have LaTeX treat the doctor's whole name as one entity, you would type it like this:

```
Dr.~Samuel~Johnson
```

On the other hand, you may want to end a line before LaTeX breaks it at a margin, and in these cases you can use the special control symbol '\\'.

To break a line of text here

you would type this line in your file:

```
To break a line of text here\\you would type this line.
```

Special Characters

Quotation Marks

You have probably noticed that the single and double quotation marks in books go in both directions, as in this example: “Character is simply habit long continued.” Most terminals have only one kind of double-quote mark key, but most let you type two different kinds of single-quote marks–the standard apostrophe (the ’ on your keyboard) and the accent gràve (the ‘ key). If you place these left and right single quote marks around a word or phrase, LaTeX prints them as typeset quality quotes. If you want double-quote marks, just type two of these single-quote marks together. For example, to print ‘To be, or not to be,’ and “That is the question.”, your text on the screen must look like this:

```
‘To be, or not to be,’ ‘‘That is the question.’’
```

Hyphens and Dashes

Your terminal probably has only one kind of dash (or hyphen) key, but LaTeX allows you to print dashes of three different lengths, which professional publishers use in different circumstances. You can put the shortest dash, which appears in compound words like user-friendly, onto your printed page with the regular hyphen (or dash) key on your keyboard. Publishers place what they call an en-dash (so named because it is about as wide as the letter “N”) in number ranges like chapters 12–24. You can have LaTeX print this en-dash by typing two regular dashes together on your screen. For the longest dash, called the em-dash (it’s about as wide as an “M”), type three regular dashes together. This last kind of dash usually sets apart phrases in published prose. You can see all three dashes at work in this sentence: “The near-sighted professor read sections 3.1–3.9 of his new paper in the seminar—we all fell asleep.” To print this sentence, you would enter these lines:

```
The near-sighted professor read sections 3.1--3.9 of his new paper in the seminar---we all fell
asleep.
```

Using the Characters # $ % & { }

Besides the backslash character, LaTeX also has these characters in its special notation:

```
# $ % & { }
```

Because people commonly use these characters when they write, the designers of LaTeX made circumventing the LaTeX functions of these characters easy. To produce one of these characters you need only to precede the character with a backslash. To produce a % sign, enter **\%** and to produce a # sign, enter **\#**, and so on.

Accent Marks

Other simple control words allow you to put accents on letters. If you want to print the Spanish word for bathtub, ‘Baño,’ you can type it as **Ba\~no**. You can have LaTeX print the French verb ‘êcrivê’ by typing **\^ecrive\^e**. To put an accent mark over an i or j, however, you must tell LaTeX to print these letters without their dots by typing **\i** or **\j**. If, for example, you want to write ‘Leîa un libro mañana’ (I am going to read a book tomorrow), you would type it on your screen like this:

```
Le\^\i a un libro ma\~nana.
```

While you are thinking about printing Spanish prose, you should know that !‘ and ?‘ will give you the characters ¡ and ¿. See the *LaTeX Class Notes* for more symbols.

Type Styles and Sizes

Type Styles

You will occasionally wish to put a word or phrase in a different type style for special emphasis. With the use of control words, you can make words come out in *italics* and **bold** or the words can appear `typewritten`.

\rm	Roman (normal) type
\bf	bold face
\it	italics
\sl	slanted text
\sc	small caps
\tt	typewriter type
\sf	sans serif type
\em	emphatic style; if the style around it is Roman, it changes the style to italic, but in any other case changes the print style to plain Roman

You can change from one type style to another as often as you like, but when you switch from slanted words to unslanted, you should put the control symbol \ / before the control word that makes the text normal. The control symbol \ / adds just a tiny amount of extra space between adjoining letters of these different styles, so that you get ‘*semi*sweet’ instead of ‘*semi*sweet.’ You can print ‘*semi*sweet’ by typing:

```
\it semi\rm\/sweet
```

or

```
{\it semi}\/sweet
```

Type Sizes

LaTeX not only lets you change the style of your printed text, it also allows you to make your letters large or small. Printers measure type sizes in points (a point is 1/72 of an inch, so 72-point text would print letters one inch high). Most of the words in this document, for example, are in ten-point type. All of the control words that alter the type size gauge their changes against the ‘normal’ type size, which in most papers (and in the case of this list) is ten points.

\normalsize	normal type size (ten points)
\large	twelve point type
\Large	fourteen point type
\LARGE	seventeen point type
\huge	twenty point type

\Huge	twenty-five point type
\small	nine point type
\footnotesize	same size as a footnote (eight point type)
\scriptsize	same size as mathematical subscripts (seven point type)
\tiny	five point type

When you change from one type size to another, LaTeX automatically switches back into roman lettering. If you want to print something huge in boldface, you should type the \bf after the \huge, like this:

```
{\huge\bf something small in italics}
```

instead of the opposite order.

Starting a Document

Perhaps the best way to quickly create a LaTeX document is to work from a template. Once you have created one or two documents, you can alter them to suit other needs of a similar style. The section *Minimum Requirements* explained the two elements you must have in a document: a **\documentstyle** command, a **\begin{document}**, and an **\end{document}** command.

To help you choose a document style and to give you some examples to look at, samples of the article, report, and letter document styles are below.

Example of the Article Document Style

The article document style works well for short, simple documents. The title, author, and date appear on the same page as the main text of the document.

```
\documentstyle{article}
\title{The Benefits of Playing Video Games}
\author{Cody Walden \\ Darren Mihelick}
\date{December 22, 1994}
\begin{document}
\maketitle

Much has been written on the evil effects of children being allowed to play video games unsupervised. This article looks at the reverse viewpoint and examines the beneficial consequences of video game playing.

Although cost was once a factor against game playing, the advent of home video games has reduced the cost...

\end{document}
```

Example of the Report Document Style

The report document style works well for longer, more complex documents. A title page contains the title, author, and date fields. The next page contains the abstract and is followed by the page(s) containing the table of contents. In this example below, we put in the \newpage command after the table of contents to force the beginning of the main text to be on a separate page.

```
\documentstyle{report}
\title{Impressionism: the Fleeting Moment}
\author{Cole Buchanan}
\date{January 15, 1995}
\begin{document}
\maketitle
\begin{center}
{\bf Abstract}
\end{center}

This paper examines the brief period of Impressionism and whether it affected later styles.

\tableofcontents
\newpage
\section{In the Beginning}
\subsection{Critics Reaction}
In its day, impressionism was not well received. The critics ...
\section{Artistic Principles}
\subsection{Monet}
Monet’s treatment of light...
\subsection{Renoir}
The essence of light and color combined...
\subsection{Cassette}
Carrying a new subject matter one step further, Cassette...
\end{document}
```

Example of the Letter Document Style

The letter document style was designed to make the creation of letters easy.

```
\documentstyle{letter}
\begin{document}
\address{1 Outgoing Address\\Houston, TX 77005}
\begin{letter} {Janet Baxter\\11 Receiving Address\\Peabody, SD 99777}
\opening{Dear Aunt Jan,}
Thank you for the snowshoes that you sent for Christmas. I am looking forward to using
them, but I think I will need to leave the state of Texas. Plan to use them during our ski trip
in March...
```

```
\closing{Regards,}
\signature{Ben Baxter\\Your favorite nephew}
\cc{Mom Baxter\\Grandma Baxter}
\encl{Christmas photos}
\ps{P.S. Tell Uncle Josh hello.}
\end{letter}
\end{document}
```

LaTeX Environments

How Nested Environments Work

In the section, *LaTeX Basics*, you learned that the **\begin{document}** and **\end{document}** control words must surround the body of a document. LaTeX also allows you to place more sets of **\begin** and **\end** statements within these outer boundaries. Each pair of these control words establishes a separate environment different from the surrounding text. All of the example statements in this document lie within **\begin** and **\end** environments, and most of them sit inside two nested environments. To produce this output:

```
Isak Dinesen opened her novel with “I had a farm in Africa, at
the foot of the Ngong Hills.”
```

The LaTeX code is:

```
\begin{quote}
\begin{verbatim}

Isak Dinesen opened her novel with “I had a farm in Africa, at the foot of the Ngong Hills.”

\end{verbatim}
\end{quote}
```

You can see that the verbatim environment lies within the quote environment. If the boundaries of these different environments cross, LaTeX sends you an error message; LaTeX will not accept a set of environments like these:

```
\begin{quote}
\begin{verbatim}
\end{quote}
\end{verbatim}
```

The principle at work here resembles that behind nested loops in computer programming: in both loops and environments, boundaries may not overlap.

Environments used to Produce Examples

Quote

Having seen how environments work, you probably now want to know what the different environments have to offer. The quote environment places its text within smaller margins and puts a small blank space between paragraphs instead of indenting them. If you type a quote environment like this one:

```
\begin{quote}
\small
Lieutenant Dunbar wasn't really swallowed. But that was the first word that stuck in his head.

Everything was immense.

The great, cloudless sky. The rolling ocean of grass. Nothing else, no matter where he put his eyes. No road. No trace of ruts for the big wagon to follow. Just sheer, empty space.

He was adrift. It made his heart jump in a strange and profound way.

\end{quote}
```

this will appear on your printed page:

> Lieutenant Dunbar wasn't really swallowed. But that was the first word that stuck in his head.
>
> Everything was immense.
>
> The great, cloudless sky. The rolling ocean of grass. Nothing else, no matter where he put his eyes. No road. No trace of ruts for the big wagon to follow. Just sheer, empty space.
>
> He was adrift. It made his heart jump in a strange and profound way.[1]

If you like the look of having your quoted passages printed small, use the **\small** within the quote environment as in the example above. You should also put a blank line after your last paragraph in a quote environment, especially if you have changed the type size, to prevent LaTeX from becoming confused about the proper paragraph spacing. The **\small** type size only functions inside the quote environment, so you do not need to insert a **\normalsize** command to make the rest of your text print at full size again.

Quotation

The quotation environment works much like the quote environment, except that it indents each paragraph instead of placing a space between them. If you sent the passage just quoted to LaTeX as a quotation, it would look like this on your page:

> Lieutenant Dunbar wasn't really swallowed. But that was the first word that stuck in his head.

1. from Michael Blake's *Dances with Wolves* (New York: Fawcett Gold Medal, 1988), p. 1.

Everything was immense.

The great, cloudless sky. The rolling ocean of grass. Nothing else, no matter where he put his eyes. No road. No trace of ruts for the big wagon to follow. Just sheer, empty space.

He was adrift. It made his heart jump in a strange and profound way.

Verse

You will find the verse environment helpful if you need to print poetry. Like the quote and quotation environments, a verse environment contains its text within smaller margins, but indents any part of a line that stretches beyond the right margin. If, for example, you typed the following portion of Lewis Carroll's poem, *Jabberwocky* in your file:

```
\begin{verse}
{\large Jabberwocky}

'Twas brillig, and the slithy toves\\
Did gyre and gimble in the wabe:\\
All mimsy were the borogoves,\\
And the mome raths outgrabe.\\
"Beware the Jaberwock, my son!\\
The jaws that bite, the claws that catch!\\
Beware the Jubjub bird and shun\\
The frumious Bandersnatch!"\\
\end{verse}
```

you would get a printed result that looks like this:

Jabberwocky

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.
"Beware the Jaberwock, my son!
The jaws that bite, the claws that catch!
Beware the Jubjub bird and shun
The frumious Bandersnatch!"

Verbatim

Within a verbatim environment, all of the text prints in typewriter type, all line breaks and blanks appear just as you have typed them, and all LaTeX control sequences print like regular text. You can not place an **\end{verbatim}** command inside the verbatim environment, however, since this command closes the environment. If you want to indent the lines in your verbatim environment (a common need since most people use this environment to print examples), remember to nest it within a quote environment.

At other times, you need the text to be printed with the words positioned as you typed them. For instance, the following excerpt from A.A. Milne's *Winnie the Pooh* would be produced by enclosing it in a verbatim environment.

```
   this                          take
If    is          shall    really   to
         flyingI     never             it.
```

If you only need part of a line in the verbatim environment, you can use the **\verb** command. For instance to produce `this is an example`, you would type:

\verb+this is an example+

Note: Do not use curly braces to designate the argument, use other characters such as the plus sign. Also, the verbatim environment or the **\verb** command can not be arguments for other commands. However, they can be inside other environments.

Environments for Aligning Text

Center

If you need to center a title or other information, use the center environment. Instead of spreading the words across each line, LaTeX centers the words on each line. If you type some short lines (broken with the \\ control symbol) like these:

\begin{center}
{\large\bf Why Being is Nothing}\\
{\bf (Though Perhaps It Isn't)}\\
{\small\it An Analysis of Hegel's Theory of Being}
\end{center}

you can get a high-quality title like this one on your printed page:

Why Being is Nothing

(Though Perhaps It Isn't)

An Analysis of Hegel's Theory of Being

Flushleft and Flushright

If you need a portion of text aligned with the left or right margin, use the *flushleft* or *flushright* environments. The flushright environment lets you put a header on a letter with ease. Each line of text runs against the right margin in the flushright environment, so if you wanted to send a letter to J. L. (a talk show host), you could type these lines:

\begin{flushright}
Typical Rice Student\\
Rice University\\
6100 South Main St.\\
Houston, Texas 77005\\

```
\end{flushright}

J. L.\\
c/o The Tonight Show\\
NBC Studios\\
Burbank, California
```

at the top of your page. When it prints it will appear as:

Typical Rice Student
Rice University
6100 South Main St.
Houston, Texas 77005

J. L.
c/o The Tonight Show
NBC Studios
Burbank, California

The flushleft environment aligns the lines of text against the left margin, letting the lines run ragged on the right side instead of spreading the words to fill each one. If you do not want your lines to be justified then use the flushleft environment. Compare the two styles:

With the Flushleft Environment

One of hardest things about using LaTeX is deciding how to pronounce it. This is also one of the few things I'm not going to tell you about LaTeX, since pronunciation is best determined by usage, not fiat. TeX is usually pronounced teck, making *lah*-teck, lah-*teck*, and *lay*-teck the logical choices; but language is not always logical, so *lay*-tecks is also possible.[1]

Normal

One of hardest things about using LaTeX is deciding how to pronounce it. This is also one of the few things I'm not going to tell you about LaTeX, since pronunciation is best determined by usage, not fiat. TeX is usually pronounced teck, making *lah*-teck, lah-*teck*, and *lay*-teck the logical choices; but language is not always logical, so *lay*-tecks is also possible.

Environments for Creating Lists

If you need to put a list into what you write, LaTeX makes your task simple by providing three special environments for:

- itemized lists
- numbered lists

1. From Leslie Lamport's *LaTeX User's Guide and Reference Manual* (Reading, Mass.:Addison-Wesley, 1986), p.5.

- description lists

All of these list environments automatically indent an item's text if it runs longer than a single line, and a single item can contain several paragraphs.

Itemize

With the itemize environment, you can make itemized lists that look like the one in the last paragraph. Lines to create the list above would look like this:

```
\begin{itemize}
\item itemized lists
\item numbered lists
\item description lists
\end{itemize}
```

You must place the **\item** control word before each entry in any of the list environments, and no text may precede the first **\item** control word in a list environment.

Enumerate

The enumerate environment creates a numbered list, which differs from an itemized list only in that LaTeX puts a number instead of a bullet in front of each item. If you substituted the word enumerate for itemize in the example lines that appear in the preceding paragraph, you could print that list to look like this:

1. itemized lists
2. numbered lists
3. description lists

Description

The description environment simplifies printing definition lists. You can label each item in a description environment with a different optional parameter, which writes whatever you type as the parameter just before the text. You must surround these optional parameters with square brackets instead of curly ones for LaTeX to recognize them. With lines of text like these:

```
\begin{description}
\item[Houston] a perfect demonstration of what results when a city populated by automobile drivers has no zoning laws.
\item[New York City] the most (only?) cosmopolitan, most densely populated, and most expensive city in the United States.
\item[San Francisco] a city of people so odd or tolerant of oddness that they build more skyscrapers while waiting for the next major earthquake.
\end{description}
```

you can print a list that looks like this:

Houston a perfect demonstration of what results when a city populated by automobile drivers has no zoning laws.

New York City the most (only?) cosmopolitan, most densely populated, and most expensive city in the United States.

San Francisco a city of people so odd or tolerant of oddness that they build more skyscrapers while waiting for the next major earthquake.

LaTeX prints the words of each optional parameter in bold face type, but you can change the type style or size by putting a control word inside the brackets. If you want Houston to appear in italics in this list, for example, you could change the optional parameter to read **\item[\it Houston]**. If you place no optional parameters in your description list, LaTeX will print it like the other sorts of lists, but without any kind of marker before each new item.

Nested Lists

Should you ever need to print a more complex list with many subheadings, you can nest lists inside one another just like any other environment. LaTeX automatically indents each deeper level and gives it a different marking system so your list will be clear and easy to read. For example, to print an outline for a speech like this one:

1. “I am not against movies...”
   (a) They can evoke entire worlds.
      i. Deep Space
      ii. Vietnam
      iii. Camelot
      iv. And so on...
   (b) Big Hollywood money buys big production values and big talent.
2. “Well, maybe not always talent...”
   (a) Yes, filming on location almost always looks better than sets...
   (b) Big budget does not always bring you big talent!
      i. Goldie Hawn vs. Meryl Streep
      ii. Patrick Swayze vs. Dustin Hoffman
   (c) Commerce of pretty faces and fast rewrites for the lowest common denominator.
3. The Advantages of Theater
   (a) Theater aspires to art.
      i. Long rehearsals to master the material
      ii. Classic plays (survive because of their high quality)
   (b) Power of watching people portray characters in the flesh.
      i. Energy between audience and actors on a stage.
      ii. Actors respond to a live audience.

you only need to enter this set of nested enumerate environments:

```
\begin{enumerate}
    \item “I am not against movies...”
\begin{enumerate}
    \item They can evoke entire worlds.
\begin{enumerate}
    \item Deep Space
```

```
        \item Vietnam
        \item Camelot
        \item And so on...
\end{enumerate}
        \item Big Hollywood money buys big production values and big talent.
\end{enumerate}
        \item “Well, maybe not always talent...”
\begin{enumerate}
        \item Yes, filming on location almost always looks better than sets...
        \item Big budget does not always bring you big talent!
\begin{enumerate}
        \item Goldie Hawn vs. Meryl Streep
        \item Patrick Swayze vs. Dustin Hoffman
\end{enumerate}
        \item Commerce of pretty faces and fast rewrites for the lowest common denomina-
tor.
\end{enumerate}
        \item The Advantages of Theater
\begin{enumerate}
        \item Theater aspires to art.
\begin{enumerate}
        \item Long rehearsals to master the material
        \item Classic plays (survived because of their high quality)
\end{enumerate}
        \item Power of watching people portray characters in the flesh.
\begin{enumerate}
        \item Energy between audience and actors on a stage.
        \item Actors respond to a live audience.
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

Special Features

Chapters and Sections

LaTeX breaks a document into different kinds of sections that depend on the **\documentstyle** you define. In the report style, you can have Chapters, Sections, Subsections, Subsubsections, Paragraphs, and Subparagraphs. Though much like the report style, the article style does not have chapters.

To set one of these breaks in your document, just enter the appropriate control word. The following example would have to appear in a document using the report style since it contains the **\chapter** command.

```
\chapter{Investigations of a Dog}
\section{Following my Nose}
\subsection{The Cat}
\subsection{The Fire Hydrant}
```

LaTeX numbers all of these divisions for you automatically. If this example was the first section of the document, it would be numbered:

1 Investigations of a Dog

1.1 Following my Nose

1.1.1 The Cat

1.1.2 The Fire Hydrant

Tables of Contents

If you make the effort to break your document into sections and chapters, you will probably also want a table of contents at the front of your paper to help you find them all. LaTeX makes it easy by automatically printing one where you enter the **\tableofcontents** control word.

Note: If you are using article style and do not want the rest of your document's text to begin right under your table of contents, place the **\newpage** control word after it. The **\newpage** control word is also useful if you would like to begin your chapters or sections at the top of a new page. Simply place the **\newpage** control word on the line preceding the **\chapter** control line.

Footnotes

If you are writing a research or technical paper, you may need to cite other sources. LaTeX makes footnotes as simple as tables of contents. You can insert a footnote in your text right after the word you want to note with the **\footnote** control word. On the screen, an example looks like this:

```
consolation for one's failures. \footnote{from Walter Kaufmann's {\it From Shakespeare to
Existentialism} (Princeton, New Jersey: Princeton University Press, 1980), p. 3.}
```

LaTeX numbers all your footnotes automatically and adjusts the order if you decide to delete or add one.

Note: With LaTeX you can also generate an index. For information on this function, see any of the references listed at the end of this document.

Comments

While typing your document on the screen, you may want to leave yourself (or someone else) editing notes or reminders. You do not have to bother with scratch paper, however, because LaTeX lets you put comments, which will not appear on the final printed page, just in your file. Any text you place after a percent sign (%) up to the end of that input line will not appear in your printed file.

Advanced Features

Graphics

Leaving Space to Manually Paste in Graphics

If you need to reserve an area of white space so that you can paste a graphic into the final product, you should follow the example below. This example places 3.5 inches of vertical space and prints the caption “Imaginary Part of log(*Z*).”

```
\begin{figure}
\vspace{3.5in}
\caption[]{ Imaginary Part of $\log(Z)$ }
\end{figure}
```

Note: You can use other units of measurement besides inches with the **\vspace** command. You can use: mm (millimeters), cm (centimeters), pt (point size), pc (picas), and ex (height of letter x in current font).

Merging in Postscript Graphics Files

To merge graphics directly into a LaTeX document, the graphics should be in a postscript file. The macro package, “epsffile” can be used to include PostScript graphics. Near the beginning of a document (after “\begin{document}”) you must tell LaTeX to include the epsf package by putting it in the square brackets in the documentstyle command. By default, epsf will use the natural size of the PostScript, but you can specify a different width and height. For example, to merge the postscript file **myfile.ps**, which requires 3.5 inches of vertical space, and add the caption “Imaginary Part of log(*Z*),” the LaTeX commands look like this:

```
\input{psfig}
\begin{figure}
\centerline{\psfig{myfile.ps, height=3.5in}}
\caption[]{ Imaginary Part of $\log(Z)$ }
\begin{center}
This line was merged using {\em epsffile}.
\end{center}
\end{figure}
```

Tables and Charts

LaTeX has the capacity to do tables and charts, however, these abilities extend beyond the scope of this document. Check one of the manuals listed in the *For More Information* section for more infor-

mation on the tabular and array environment. An example of a table and the LaTeX code that produced it is given below to give you an indication of how the process works.

Software Summary	
Name:	SAS
Version:	6.06
Brief description:	data management, statistical analysis, and report generation
System:	mainframe
Level of support:	Information Technology provides program maintenance and consulting.
Other references:	See the bibliography. If you have just begun learning SAS, you should consult the SAS Introductory Guides available in the Computing Reference Area. Other SAS manuals are also available.

This table would be produced with the following code:

```
\begin{tabular}{| l |p{4.25in}|} \hline
\multicolumn{2}{|c|}{\bf Software Summary} \\ \hline
{\bf Name:} & SAS\\
{\bf Version:} & 6.06 \\
{\bf Brief description:} & data management, statistical analysis, and report generation \\
{\bf System:} & mainframe \\
{\bf Level of support:} & Information Technology provides program maintenance and consulting. \\
{\bf Other references:} & See the bibliography. If you have just begun learning SAS, you should consult the SAS Introductory Guides available in the Computing Reference Area. Other SAS manuals are also available. \\
\hline
\end{tabular}
```

Mathematical Formulas

LaTeX can produce quite complicated mathematical formulas. This document only covers some very basic features. To learn about the advanced features refer to one of the books listed in the *For More Information* section.

To produce mathematical formulas in LaTeX, you need to be in math mode. How you enter math mode depends on how your formula will be displayed. A formula can appear in a sentence, called an in-text formula, or by itself, called a display formula. In-text formulas are produced by the math environment, which you can evoke in 3 ways as the following example illustrates. x + y - z can be produced by **$x+y-z$** or **\(x+y-z\)** or **\begin{math}x+y-z\end{math}**.

Display formulas are produced by the displaymath and equation environments with the difference being that equation numbers the formulas. The displaymath environment can be called in two ways with **\[. . .]** or **\begin{displaymath}** . . . **\end{displaymath}**. The equation environment must be called with **\begin{equation}** . . . **\end{equation}**.

Superscripts

Superscripts are produced using the ^ command. For example, $4x^{3y}$ is produced by **$4x^{3y}$**.

Subscripts

Subscripts are produced by using the **_** command. For example, x_{3y} is produced by **x_{3y}**.

Fractions

To produce fractions in running text use the **/** command and for displayed formulas use the **/frac** command. Examples of fractions in text are x/3 and (a-b)/c which are produced by the commands **$x/3$** and **$(a-b)/c$**.

To produce a displayed formula, use the **/frac** command with the numerator and the denominator delimited by brackets.

$$a = \frac{b + c/4}{d^2 - 1}$$

is produced by the code **\[a= \frac{b+c/4}{d^{2}-1}\]**.

Roots

To display a square root or other root use the **\sqrt** command. For example $\sqrt{a+b}$ and $\sqrt[3]{4x}$ are produced by the commands **$\sqrt{a+b}$** and **$\sqrt[3]{4x}$**.

Processing and Printing with LaTeX

UNIX Systems

If you are working on Owlnet, you can process your file with LaTeX and then either preview it on your screen or print it.

Processing

LaTeX can recognize any text file containing its command words, though it is helpful to designate LaTeX files by adding the suffix **.tex**.. You can use the **mv** command to rename your files. For instance, to rename **paper** to **paper.tex**, enter:

```
mv paper paper.tex
```

To process your file with LaTeX, enter the command

latex *filename*

As it processes the file, LaTeX will display messages on your screen. (This process creates several new files. See the section on LaTeX filetypes below.) Error messages can be intimidating due to the complexity of the messages. However, knowing which part to look at and which part to ignore helps. Generally the first two lines of a LaTeX error are:

```
LaTeX error. See LaTeX manual for explanation.

Type H <return> for immediate help.
```

The next line that begins with a `!` is important. This explains the problem such as an environment variable is not understood (due to a typo) or an environment is not closed with an **\end**. After these lines are some messages that explain where the problem is in terms of the typesetting commands. You can ignore these lines. Finally if LaTeX wants a response from you it will type a `?` and wait for your input. You can respond in various ways:

- enter command – if a typo is encountered you can enter the correct command and LaTeX will continue to process your file
- RETURN – press the RETURN key and LaTeX will continue to run
- I\stop – entering this sequence of a capital I and \stop will try to exit you from the process
- x – if I\stop fails to exit you from LaTeX, x should always work

Write down any error messages during this process and then go back into the file to fix any problems then repeat the process again. Occasionally, you will receive the message:

```
* (Please type a command or say ‘end’)
```

Enter the command

\stop

If this command does not have an effect, you will have to type **^Z**, then **kill %latex**.

Previewing

If you are working in an X environment, you can preview your **.dvi** file on the screen by entering:

xdvi *filename*

Your cursor will be attached to the upper left corner of a new window frame, just click the cursor where you want the upper left hand corner of the window to be placed. You can use the buttons on the right side of the window to control the size of the page and the page that is being examined. With your cursor in the window, you can also type an “n” to move to the next page or a “p” to move to the previous page.

Printing

The process of printing your file actually involves converting the dvi file to postscript, which is what the utility **dvips** does. The command **lpr** sends the file to be printed and so it needs to know the filename and the printer where you want your file to go. To send the file **document.dvi** to the printer ryonps, for example, you would enter the following line:

dvips document | lpr -Pryonps

and your file should come out of the printer in crisp black and white.

LaTeX Filetypes

LaTeX creates several new files during the processing. For instance, in running the file **george.tex**, which has a table of contents, four more files will be created: **george.log**, **george.dvi**, **george.aux**, and **george.toc**.

.tex – the source file with the text and LaTeX commands

.log – the transcript of the LaTeX session, contains all the error messages, etc.

.dvi – the formatted output; this is the file you send to the printer

.aux – the auxiliary reference file for page breaks, titles, index entries, etc.

.toc – contains the table of contents information description

For More Information

Though you now know how to write and print a document with LaTeX, you have only seen a fraction of LaTeX's capabilities. To learn how to make tables, print complex math functions, or just use the multitude of other LaTeX control words, refer to one of the books in the following list. You can find all of them in the Computing Reference Collection (B40 Fondren Library) and some of them in the labs.

David J. Buerger, *LaTeX for Scientists and Engineers*— A guide to LaTeX's more sophisticated features, especially those helpful in preparing scientific papers.

Leslie Lamport, *LaTeX – A Document Preparation System*— The best LaTeX general reference by the creator of LaTeX.

Michael Urban, *An Introduction to LaTeX* — An excellent introduction to the LaTeX system.

Problems or Questions

If you have problems or questions, you can contact the Consulting Center (103 Mudd Lab, or at 527-4983). You can also submit it on the Web at http://problem.rice.edu or you can send e-mail to problem@rice.edu.