1 Chapter 1: Getting Acquainted

2 Chapter 2: Getting Started

2.1 Preparing an Input File

2.2 The Input

2.2.1 Sentences and Paragraphs

Words are separated by one or more spaces. Paragraphs are separated by one or more blank lines. The output is not affected by adding extra spaces or extra blank lines to the input file.

A Warning or Two

If you get too much space after a mid-sentence period—abbreviations like etc. are the common culprits)—then type a backslash followed by a space after the period, as in this sentence.

Quotation Marks

Double quotes are typed like this: “quoted text”. Single quotes are typed like this: ‘single-quoted text’.

Dashes

You can produce three different sizes of dash by typing one, two, or three “-” characters:

Intra-word dash: X-ray

Medium dash: 1–2

A punctuation dash—like this.

Long dashes are typed as three dash characters—like this.

Space After a Period

You tell TeX that a period doesn't end a sentence by using the backslash space command (a backslash followed by a space or end of line).

Tinker et al. made the double play.

Don't leave any space between the period and the backslash.

Special Symbols

Remember, don't type the 10 special characters (such as dollar sign and backslash) except as directed! The following seven are printed by typing a backslash in front of them: $ & # % _ { and }. The manual tells how to make other symbols.

Simple-Text-Generating Commands

Some people use plain TEX, but I prefer LATEX.

This page was produced on June 17, 2025.

The following is an ellipsis example.

If nominated . . . , I will not serve.

Emphasizing Text

Emphasized text is typed like this: *this is emphasized.* Bold text is typed like this: **this is bold**.

You can have *empahsized text* within *emphasized text* too.

Preventing Line Breaks

Typesetters break lines between words when possible and split words only between syllables (inserting a hyphen at the break).

Sometimes a line should not be broken between or within words. Human typesetters recognize these situations, but TEXbust be told about some of them.

Line breaking should be prevented at certain interword spaces.

Typing a ~ (a tilde character) produces ordinary interword space at which TEX will never break a line.

Below are some examples indicating when a ~ should be used.

Mr. Jones

Figure 7

(1) gnats

U. S. Grant

from 1 to 10

The \mbox command tells TEX to print its entire argument on the same line.

Doctor Lamport, I presume?

Footnotes

Footnotes are produced with the footnote command having the text of the footnote as its argument.

Gnus[1] can be quite a gnusance.

A \footnote command cannot be used in the argument of most commands; for example, it can't appear in the argument of an \footnote command.

Formulas

If you're writing a technical document, it's likely to contain mathematical formulas. A formula appearing in the middle of a sentence is enclosed by \(and \) commands.

Any spaces that you type in the formula are ignored.

Does $x + y$ always equal $y + x$?

TEX regards a formula as a word, which may be broken across lines at certain points, and space before the \(or after the \) is treated as an ordinary interword separation.

Subscripts are produced by the _ command and superscripts by the ^ command.

$a_1 > x^{2n}/y^{2n}$

These two commands can only be used inside a mathematical formula.

When used in a formula, the right-quote character ' produces a prime ($'$), two in a row produce a double prime.

[1] A gnu is a big animal.

$x' < x'' - y'_3 < 10x'''z$.

The formula $a < 7$ is a noun in this sentence. It is sometimes used as a clause by writing that $a < 7$.

Beginning a sentence with a formula makes it hard to find the start of the sentence; don't do it.

A variable like x is a formula. To save you some typing, LaTeX treats $...$ the same as \(...\).

Let x be a prime such that $y > 2x$.

Use $...$ only for a short formula, such as a single variable. It's easy to forget one of the $ characters that surrounds along formula. You can also type

\begin{math} ... \end{math}

instead of \(... \(. You might want to use this form for a very long formula.

Ignorable Input

When TeX encounters a % character in the input, it ignores the % and all characters following it on that line – including the space charactr that ends the line. TeX also ignores spaces at the beginning of the next line.

Gnus and armadillos are generally tolerant of one another and seldom quarrel.

The % has two uses: ending a line without producing any space in the output]footnoteHowever, you can't split a command across two lines. and putting a comment (a note to yourself) in the input file.

2.2.2 The Document

The text of every document starts with \begin{document} command and ends with an \end{document}. The part of the input file preceding the \begin{document} command is called the *preamble*.

The Document Class

The preamble begins [2] with a \documentclass command whose argument is one of the predefined classes of document that LaTeX knows about. The file sample2e.tex begings with

\documentclass{article}

which selects the article class. The other standard LaTeX classs used for ordinary documents is the report class. The article class is generally used for shorter documents than the report class. Other standard classes are described in Chapter 5.

In addiction to choosing the class, you can also select from among certain document-class options. The options for the article and report classes include the following.

11pt Specifies a size of type known as *eleven point*, which is ten percent larger than the ten-point type normally used.

12pt Specifies a twelve-point type size, which is twenty percent larger than ten point.

twoside Formats the output for printing on both sides of the page. (LaTeX has no control over the actual printing.)

twocolumn Produces two-column output.

[2] As explained in Section 4.7, the \documentclass command may actually be preceeded by prepended files.

You specify a document-class option by enclosing it in square brackets immediately after the "\documentclass", as in

\documentclass[twoside]{report}

Multiple options are separated by commas.

\documentclass [twocolumn,12pt]{article}

Don't leave any space inside the square brackets.

The \documentclass command can be used either with or without the option-choosing part. The options, enclosed in square brackets are an *optional argument* of the command. It is a LaTeX convention that optional arguments are enclosed in square brackets, while mandatory arguments are enclosed in curly braces. TeX ignores spaces after a command like \documentclass and between command arguments.

The document class defines commands for specifying LaTeXś standard logical structures. Additional structures are defined by *packages*, which are loaded by the \usepackage command. For example, the command

\usepackage{latexsym}

loads the latexsym package, which defines commands to produce certain special math symbols. (See Section 3.3.2). A package can have options, specified by an optional argument of \usepackage just like the one for \documentclass.

You will probably want to define some new commands for the special structures used in your particular document. For example, if you're writing a cookbook you will probably define your own commands for formatting recipes, as explained in Section 3.4. These definitions should go in the preamble, after the \documentclass and \commands. The preamble can also contain commands to change some aspects of the formatting. If you have commands or format changes that you use in several documents, you may want to define your own package, as described in Section 6.1.4.

The Title "Page"

A document usually has a title "page" listing its title, one or more authors, and a date. I write "page" in quotes, because, for a short document, this information may be listed on the first page of text rather than on its own page. The title information consits of the title itself, the author(s), and the date, it is specified by the three declarations \title, \author, and \date. The actual title "page" is generated by a \maketitle command.

\title{Gnus of the World}
\author{R. Dather \and J. Pennings \and B. Talkmore}
\date{4 July 1997}
\maketitle

Note how multiple authors are separated by \and commands.

The \command comes after the \begin{document}, usually before any other text. The \title, \author, and \date commands can come anywhere before the \maketitle. The \date is optional; LaTeX supplies the current date if the declaration is omitted, but the \title and \author must appear if a \make title command is used.

Commands for adding other information, such as the author's address and an acknowledgement of support, are described in Section C.5.4.

2.2.3 Sectioning

Sentences are organized into paragraphs, and paragraphs are in turn organized into a hierarchical *section structure*. You are currently reading Subsection 2.2.3, titled *Sectioning*, which is

part of Section 2.2, titled *The Input*, which in turn is part of Chapter 2, titled *Getting Started*. I will use the term *sectional units* for things like chapters, sections, and subsections.

A sectional unit is begun by a sectioning command with the unit's title as its argument.

\subsection{A Sectioning Command}

produces

4.7 A Sectioning Command

LaTeX automatically generates the section number. Blank lnes before or after a sectioning command have no effect.

The document class determines what sectioning commands are provided; the standard classes have the following ones. [3]

\part
\subsection
\paragraph
\chapter
\subsubsection
\subparagraph
\section

The article document class does not contain a \chapter command, which makes it easy to include an "article" as a chapter of a "report". The example above, like most others in this book, assumes the article document class, the 4.7 indicating that this is the seventh subsection of Section 4. In the report class, this subsection might be numbered 5.4.7, with the "5" being the chapter number.

The \part command is used for major divisions of long documents; it does not affect the numbering of smaller units – in the article class, if the last section of Part 1 is Section 5, then the first section of Part 2 is Section 6.

If there is an appendix, it is begun with an \appendix command uses the same sectioning commands as the main part of the document. The \appendix command does not produce any text; it simply causes sectional units to be numbered properly for an appendix.

The document class determines the appearance of the section title, including whether or not it is numbered. Declarations to control section numbering are described in Section C.4, which also tells you how to make a table of contents.

The argument of a sectioning command may be used for more than just producing the section title; it can generate a table of contents entry and a running head at the top of the page. (Running heads are discussed in Section 6.1.2.)

Fragile commands are rarely used in the argument of a sectioning command. Of the commands introduced so far, the only fragile ones are \(, \), \begin, \end, and \footnote – none of which you're likely to need in a section title. [4] On the rare occasions when you have to put a fragile command in a section title, you simply protect it with a \protect command. The \protect command goes right before every fragile command's name, as in:

\subsection {Is \protect\(x+y \protect\) Prime?}

This is actually a silly example, because $ is not a fragile command, so you can instead type

\subsection {Is \$x+y\$ Prime?}

[3]The names \paragraph and \subparagraph are unfortunate, since they denote units that are often composed of several paragraphs; they have been retained for historical reasons.

[4]Section C.3.3 tells you how to footnote a section title.

/noindent but because the problem is so rare, it's hard to find a good example using the commands described in this chapter.

An argument in which fragile commands need \protect will be called a *moving* argument. Commands that are not fragile will be called *robust*.

2.2.4 Displayed Material

The following is an example of a short displayed quotation.

> ... it's a good idea to make your input file as easy to read as possible.

It is indented at both margins.

This example illustrates a type of LaTeX construction called an *environment*, which is typed

\begin{*name*} ... \end{*name*}

where *name* is the name of the environment. The **quote** environment produces a display suitable for a short quotation. You've already encountered two other examples of environments: the **math** environment and the **document** environment.

The standard LaTeX document classes provide environments for producing several different kinds of displays. Blank lines before or after the environment mark a new paragraph. Thus, a blank line after the \end command means that the following text starts a new paragraph. Blank lines before and after the environment mean that it is a complete paragraph.

It's a bad idea to start a paragraph with displayed material, so you should not have a blank line before a display environment without a blank line after it. Blank lines immediately following a display environment's \begin command and emmediately preceding its \end command are ignored.

Quotations

LaTeX provides two different environments for displaying quotations. The quote environment is used for either a short quotation or a sequence of short quotations separated by blank lines.

Our presidents have been known for their pithy remarks.

> The buck stops here. *Harry Truman*
>
> I am not a crook. *Richard Nixon*
>
> It's no exaggeration to say the undecideds could go one way or another. *George Bush*

The **quotation** environment is used for quotations of more than one paragraph as usual, the paragraphs are separated by blank lines.

Here is some advice to remember when you are using LaTeX.

> Environments for making quotations can be used for other things as well.
>
> Many problems can be solved by novel applications of existing environments.

Lists

LaTeX provides three list-making environments: **itemize**, **enumerate** and **description**. In all three, each new list item is begun with an \item command. Itemized lists are made with the **itemize** environment and enumerated lists with the **enumerate** environment.

- Each lst item is marked with a *label*. The labels in this itemized list are bullets.
- Lists can be nested within one another.
    1. The item labels in an enumerated list are numerals or letters.
    2. A list should have a least two items.

  LaTeX permits at least four levels of nested lists, which is more than enough.
- Blank lnes before an item have no effect.

In the **description** environment, you specify the item labels with an optional argument to the \item command, enclosed in brackets. (Although the argument is optional, the item will look funny if you omit it.)

Three animals you should know about are:

gnat A small animal, found in the North Woods, that causes no end of trouble.

gnu A large animal, found in crossword puzzles, that causes no end of trouble.

armadillo a medium-sized animal, named after a medium sized Texas city.

The characters [and] are used both to delimit and optional argument and to produce square brackets in the output. This can cause some confusion if the text of an item begins with a [or if an \item command's optional argument contains a square bracket. Section C.1.1 explains what to do in these uncommon situations. All commands that have an optional argument are fragile.

Poetry

Poetry is displayed with the **verse** environment. A new stanza is begun with one or more blank lines; lines within the stanza are separated by the \\ command.

> There is an environment for verse
> Whose features some poets will curse.
>
> For isntead of making
> Them do *all* line breaking,
> It allows them to put too many words on a line when
> they'd rather be forced to be terse.

The * command is the same as \\except that it prevents LaTeX from starting a new page at that point. It can be used to prevent a poem from being broken across pages in a distracting way. The commands \\and * are used in all environments in which you tell LaTeX where to break lines; several such environments are described in the next chapter. The * command is called the *-form of the \\ command.

Several other commands also have *-forms–versions of the command that are slightly different from the ordinary one–that are obtained by typing * after the command name.

The \\and * commands have a little-used optional argument described in section C.1.6, so putting a [after them presents the same probelm as for the \item command. Moreover, the * in the * command is somewhat like an optional argument for the \textbackslash command, so following a \\with a * in the text posees a similar problem. See Section C.1.1 for the solutions to these unlikely problems. Almost every command that has a *-form is fragile, and its *-form is also fragile.

Displayed Formulas

A mathematical formula is displayed when either it is too long to fit comfortably in the running text, it is so important that you want it to stand out, or it is to be numbered for future reference.

LaTeX provides the **displaymath** and **equation** environments for displaying formulas; they are the same except that **equation** numers the formula and **displaymath** doesn't.

Because displayed equations are used so frequently in mathematics, LaTeX allows you to type \[... \] instead of

```
\begin{displaymath} ... \end{displaymath}
```

Here is an example of an unnumbered displayed equation:

$$x' + y^2 = z_i^2$$

and here is the same equation numbered:

$$x' + y^2 = z_i^2 \tag{1}$$

The document class determines how equations are numbered. Section 4.2 describes how LaTeX can automatically handle references to equation numbers so you don't have to keep track of the numbers.

A displayed formula, like any displayed text, should not begin with a paragraph. Moreover, it should not form a complete paragraph by itself. These two observations are summed up in a simple rule: in the input, never leave a blank line before a displayed formula.

TeX will not break the formula is a **displaymath** or **equation** environment across lines. See Section 3.3.5 for commands to create a single multiple-line formula or a sequence of displayed formulas.

2.2.5 Declarations

You may want to emphasize a large piece of text, such as a quotation. You can do this with the *c*ommand, but that makes the input file hard to read because you have to search for the closing right brace to see where the argument ends. Moreover, it's easy accidentally to delete the closing brace when you edit the text. Instead, you can use the **\em** command, which tells TeX to start emphasizing text.

This prose is very dull.

> Wait! *Her is an exciting quote.*

Aren't you glad all that excitement is over?

As explained below, the **\end{quote}** cause TEX to stop emphasizing text.

Unlike other commands you've encountered so far, **\em** produces neither text nor space; instead if affects the way LATEX prints the text that follows it. Such a command is called a *declaration*. Most aspects of the way LATEX formats a document–the type style, how wide the margins are, and so on–are determined by declarations. The **\em** declaration instructs LATEX to change the type style to the appropriate one for indicating emphasis. The *scope* of a declaration is ended by the first **\end** command or right brace. In the input, braces and **\begin** and **\end** commands must come in matched pairs. The scope of a declaration is ended by the first **\end** or } whose matching **\begin** or { precedes the declaration.

2.3 Running LATEX