

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 L^AT_EX.

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 *emphasized 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 TeX must 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.

Gnus¹ 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.

¹A 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, L^AT_EX treats $\$...\$$ the same as $\backslash(...\backslash)$.

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

$\backslash\begin{math} \dots \end{math}$

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

Ignorable Input

When T_EX encounters a $\%$ character in the input, it ignores the $\%$ and all characters following it on that line – including the space character that ends the line. T_EX 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 $\backslash\begin{document}$ command and ends with an $\backslash\end{document}$. The part of the input file preceding the $\backslash\begin{document}$ command is called the *preamble*.

The Document Class

The preamble begins ² with a $\backslash\documentclass$ command whose argument is one of the predefined classes of document that L^AT_EX knows about. The file sample2e.tex begins with

$\backslash\documentclass{article}$

which selects the article class. The other standard L^AT_EX classes 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 addition 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. (L^AT_EX has no control over the actual printing.)

twocolumn Produces two-column output.

²As explained in Section 4.7, the $\backslash\documentclass$ command may actually be preceded by prepended files.

You specify a document-class option by enclosing it in square brackets immediately after the “ $\backslash\documentclass$ ”, as in $\backslash\documentclass[twoside]{report}$

Multiple options are separated by commas.

$\backslash\documentclass[twocolumn,12pt]{article}$

Don't leave any space inside the square brackets.

The $\backslash\documentclass$ 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 L^AT_EX convention that optional arguments are enclosed in square brackets, while mandatory arguments are enclosed in curly braces. T_EX ignores spaces after a command like $\backslash\documentclass$ and between command arguments.

The document class defines commands for specifying L^AT_EX's standard logical structures. Additional structures are defined by *packages*, which are loaded by the $\backslash\usepackage$ command. For example, the command

$\backslash\usepackage{latexsym}$

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 $\backslash\usepackage$ just like the one for $\backslash\documentclass$.

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 $\backslash\documentclass$ and $\backslash\commands$. 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 consists of the title itself, the author(s), and the date, it is specified by the three declarations $\backslash\title$, $\backslash\author$, and $\backslash\date$. The actual title “page” is generated by a $\backslash\maketitle$ command.

$\backslash\title{Gnus of the World}$

$\backslash\author{R. Dather \and J. Pennings \and B. Talkmore}$

$\backslash\date{4 July 1997}$

$\backslash\maketitle$

Note how multiple authors are separated by $\backslash\and$ commands.

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

L^AT_EX automatically generates the section number. Blank lines 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?}`

`/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

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

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