Character-Level Conventions in \LaTeX

CSC 341, “Automata, Formal Languages, and Computational Complexity”
Department of Computer Science · Grinnell College
August 31, 2020

\LaTeX{} input files are plain text and use a subset of the characters in the American Standard Code for Information Interchange (ASCII). Characters represented by Unicode codepoints greater than 127 should not appear in these files. \LaTeX{} supports some such characters, but only through commands (such as \texttt{\textbackslash dag} for ‘†’, which is Unicode codepoint \texttt{U+2020}), not directly as input characters.

All letters, upper- and lower-case, and all ten digit characters are normally commands telling the virtual typesetting machines to set those same characters. This is also true of the following punctuation marks and mathematical symbols:

```
. : ; , ? ! ‘ ’ ( ) [ ] - / * @ + = | < >
```

\LaTeX{} interprets any sequence of two or more newlines in the input text as a paragraph break. Otherwise, it regards any non-empty sequence of whitespace characters as generic “space” — no matter how many successive spaces the author puts into the input file, \LaTeX{} treats them as equivalent to one space. The software itself normally determines how much space to leave between words, sentences, and paragraphs and even makes its own decisions about hyphenating words at the ends of lines to make it possible to achieve more uniform spacing in justified text.

The other printable ASCII characters (in other words, those that aren’t control characters) have special, conventional meanings in \LaTeX{}:

A percentage sign, \texttt{%}, indicates the beginning of a “wing comment”: \LaTeX{} discards everything from the percentage sign up to and including the newline character that terminates the line without attempting to interpret any of it. Using this convention, the author/programmer can annotate the document’s “source code” without changing the document as it is actually typeset.

Curly braces, \texttt{\{ and \}}, are used as grouping symbols, most commonly to enclose strings that are arguments to \LaTeX{} commands that take such arguments.

A backslash character, \texttt{\textbackslash{}}, indicates the beginning of a \textit{control sequence}, which is the \LaTeX{} term for the name of a command. A control sequence can be either a \textit{control word}, in which the backslash is followed by one or more letters, or a \textit{control symbol}, in which the backslash is followed by exactly one character that is not a letter.

A control sequence is terminated only when some non-letter character is reached; conventionally, \LaTeX{} ignores a space character that immediately follows a control sequence so that the author can invoke the command that the control sequence expresses and then immediately proceed to typesetting the next letter. (Enclosing the entire control sequence in curly braces will also work.)

The virtual typesetter that \LaTeX{} simulates has several operating modes, including one called \textit{math mode} that is intended mainly for mathematical expressions, equations, and displays. \LaTeX{} interprets its input differently when it is in math mode. (For example, a letter like \texttt{x} is now recognized as the name of a variable and italicized, thus: \texttt{x}). Some \LaTeX{} commands can be used only in math mode. A dollar sign, \texttt{\$}, directs \LaTeX{} to switch into or out of math mode.

A tilde character, \texttt{\textbackslash{}~}, directs \LaTeX{} to insert a space while forbidding it to end the current line and start a new one. It is used to tie together text elements that should be typeset as a unit, because a line break in the middle would be disruptive or confusing
to the reader. For example, one might type in ‘9:45 a.m.’ in order to insert a space but prevent a line break between ‘9:45’ and ‘a.m.’

The remaining ASCII characters (‘#’, ‘&’, ‘‘, and ‘’’) have specialized uses in command definitions or in mathematical expressions as displays.

Most of these special characters have corresponding command symbols directing \LaTeX just to typeset the character instead of giving it any special treatment: ‘\#’, ‘\$’, ‘\%', ‘\&’, ‘\’, ‘\{’, and ‘\}’. There is also a more general mechanism: The characters that \LaTeX is capable of typesetting are arranged in fonts, and each character in a font has a code in the range from 0 to 255. The control sequence ‘\symbol’ takes one argument, which must be a numeral in the range from 0 to 255, and directs the typesetter to insert the character from the current font that happens to have the corresponding code number. So, for instance, one could (usually) write ‘\symbol{94}’ to get the circumflex character ‘^’ from the current font.

Two other input conventions come up frequently enough to be worth learning immediately:

\LaTeX interprets a sequence of two hyphens as an en dash, ‘–’, and a sequence of three hyphens as an em dash, ‘—’. When you’re trying to draw a horizontal line, a sequence of four or more hyphens never gives you what you want in \LaTeX — there’s always a better way to do it using commands.

\LaTeX distinguishes between open- and close-quotation marks, both singles and doubles. The backquote character at the extreme upper left of your keyboard is the single open-quotation mark; the apostrophe character (next to the Enter key) is the single close-quotation mark. To get a double open-quotation mark, use two backquotes; to get a double close-quotation mark, use two apostrophes. Do not use the ASCII double quotation mark to open quotations — \LaTeX sets it as a double close-quotation mark regardless of context.