character **1**

`character.` TEX works with *characters* in two contexts: as input characters, which it reads, and as output characters, which it typesets. TEX transforms most input characters into the output characters that depict them. For example, it normally transforms the input letter '`h`' into the letter 'h' typeset in the current font. That is not the case, however, for an input character such as '`$`' that has a special meaning.

TEX gets its input characters by reading them from input files (or from your terminal) and by expanding macros. These are the *only* ways that TEX can acquire an input character. Each input character has a code number corresponding to its position in the ASCII code table. For instance, the letter '`T`' has ASCII code 84.

When TEX reads a character, it attaches a category code to it. The category code affects how TEX interprets the character once it has been read in. TEX determines (and remembers) the category codes of the characters in a macro when it reads the macro's definition. As TEX reads characters with its eyes (see "anatomy of TEX", p. '`\anatomy`') it does some "filtering", such as condensing sequences of spaces to a single space. See pages 46–48 of *The TEXbook* for the details of this filtering.

The ASCII "control characters" have codes 0–31 and 127–255. They either don't show up or cause strange behavior on most terminals if you try to display them. Nonetheless they are sometimes needed in TEX input, so TEX has a special notation for them. If you type '`^^`*c*', where *c* is any character, you get the character whose ASCII code is either 64 greater or 64 less than *c*'s ASCII code. The largest acceptable code value using this notation is 127, so the notation is unambiguous. Three particularly common instances of this notation are '`^^M`' (the ASCII ⟨return⟩ character), '`^^J`' (the ASCII ⟨line feed⟩ character) and '`^^I`' (the ASCII ⟨horizontal tab⟩ character).

TEX also has another notation for indicating ASCII code values that works for all character codes from 0 to 255. If you type '`^^`*xy*', where *x* and *y* are any of the hexadecimal digits '`0123456789abcdef`', you get the single character with the specified code. (Lowercase letters are required here.) TEX opts for the "hexadecimal digits" interpretation whenever it has a choice, so you must not follow a character like '`^^a`' with a lowercase hexadecimal digit—if you do, you'll get the wrong interpretation. If you need to use this notation you'll find it handy to have a table of ASCII codes.

An output character is a character to be typeset. A command for producing an output character has the meaning "Typeset character number *n* from the current font", where *n* is determined by the command. TEX produces your typeset document by combining such characters with other typographical elements in boxes, and arranging them on the page.

An input character whose category code is 11 (letter) or 12 (other) acts as a command to produce the corresponding output character. In addition you can get TEX to produce character *n* by issuing the command

‘`\char` n’ (p. ‘`\char`’), where n is a number between 0 and 255. The commands ‘`h`’, ‘`\char‘h`’, and `\char104` all have the same effect. (104 is the ASCII code for ‘h’.)