

number

1

number. In T_EX, a *number* is a positive or negative integer. You can write a number in T_EX in four different ways:

- 1) as an ordinary decimal integer, e.g., 52
- 2) as an octal number, e.g., '14
- 3) as a hexadecimal number, e.g., "FF0
- 4) as the code for an ASCII character, e.g., ') or '\)

Any of these forms can be preceded by '+' or '-'.

An octal number can have only the digits 0–7. A hexadecimal number can have digits 0–9 and A–F, representing values from 0 to 15. You can't, alas, use lowercase letters when you write a hexadecimal number. If you need an explanation of octal and hexadecimal numbers, you'll find one on pages 43–44 of *The T_EXbook*.

A decimal, octal, or hexadecimal number ends at the first character that can't be part of the number. Thus a decimal number ends when T_EX sees, say, a letter, even though a letter between 'A' and 'F' would not end a hexadecimal number. You can end a number with one or more spaces and T_EX will ordinarily ignore them.¹

The fourth form above specifies a number as the ASCII code for a character. T_EX ignores spaces after this form of number also. You can write a number in this form either as 'c or as '\c. The second form, though longer, has the advantage that you can use it with *any* character, even '\', '%', or '^~M'. It does have one rather technical disadvantage: when T_EX is expanding a token sequence for a command such as \edef or \write, occurrences of '\c' within numbers will also be expanded if they can be. That's rarely the effect you want.

The following are all valid representations of the decimal number 78:

78 +078 "4E '116 'N '\N

You can't use a number in text by itself since a number isn't a command. However, you can insert the decimal form of a number in text by putting a \number command (p. '\number') in front of it or the roman numeral form by putting a \romannumeral command in front of it.

You can also use decimal constants, i.e., numbers with a fractional part, for specifying dimensions (see "dimension", p. 'dimension'). A decimal constant has a decimal point, which can be the first character of the constant. You can use a comma instead of a period to represent the decimal point. A decimal constant can be preceded by a plus or minus sign. Thus '.5in', '-3.22pt', and '+1,5\baselineskip' are valid dimensions. You can't, however, use decimal constants in any context *other* than as the "factor" part of a dimension, i.e., its multiplier.

¹ When you're defining a macro that ends in a number, you should always put a space after that number; otherwise T_EX may later combine that number with something else.