

1 Printf

The % symbol is followed by either

- % *symbol*. It prints a %.
- *Flags*. One or several flags #0 - + ' '.

%2\$0-

- *Decimal number*: Specifies a minimum **field width** (minimum number of characters to be printed). If the converted value has fewer characters than the field width, it will be padded with spaces on the left (or right, if the left-adjustment flag has been given) to fill out the field width.

%2\$0 - 5

- *Point and decimal number*: Specifies a **precision**. This means the minimum number of digits to appear for diouxX conversions, the number of digits to appear after the decimal-point for a, A, e, E, f, and F conversions, the maximum number of significant digits for g and G conversions, or the maximum number of characters to be printed from a string for s conversions.

%2\$0 - 5.8

- *A length modifier h, hh, l, ll, L*: Specifies the size of the argument and depends on the particular conversion. See man.

%2\$0 - 5.8ll

- *Conversion csp, diouxX, f*: The conversion to be applied

%2\$0 - 5.8llX

1.1 Conversions

- *c*: Convert *int* to unsigned char.
- *s*: Characters from the array are written up to (but not including) a terminating NUL character; if a precision is specified, no more than the number specified are written. If a precision is given, no null character need be present; if the precision is not specified, or is greater than the size of the array, the array must contain a terminating NUL character.
- *p*: Address in Hexa. A pointer stores an address as a number. The type of this number depends on the operating system. For 32 bits it is usually an unsigned int (4 bytes) while in 64 bits is a long unsigned int (8 bytes).

- *diouxX* : The int (or appropriate variant) argument is converted to signed decimal (d and i), unsigned octal (o), unsigned decimal (u), or unsigned hexadecimal (x and X) notation. The letters “abcdef” are used for x conversions; the letters “ABCDEF” are used for X conversions. The precision, if any, gives the minimum number of digits that must appear; if the converted value requires fewer digits, it is padded on the left with zeros.
- *f*: The double argument is rounded and converted to decimal notation in the style [-]ddd.ddd, where the number of digits after the decimal-point character is equal to the precision specification. If the precision is missing, it is taken as 6; if the precision is explicitly zero, no decimal-point character appears. If a decimal point appears, at least one digit appears before it.

1.2 Lengh modifiers

-

1.3 Flags

- #: Change precision.
-

1.4 To check

- Positional and non positional arguments
- Integer limits
- Flags with some conversions have undefined behaviour (for example + and u)
- Flag - overrides 0 but gives a warning and thus and error if -Werror is used to compile
- Maximum field width