



# Format Conversions: printf, fprintf, sprintf



To get started, use **%hi** to display a **short**, **%i** for an **int**, **%li** for a **long**, **%G** for a **float** or **double**, **%LG** for a **long double**, **%c** for a **char** (or **%i** to display it as a number), and **%s** for a string (**char \*** or **char []**). Then refine the formatting further as desired.

To print a percent sign, use **%%**.

## 1. Prototypes ( in <stdio.h>)

```
int printf(const char *format, ...)
```

```
int fprintf(FILE *stream, const char *format, ...)
```

```
int sprintf(char *string, const char *format, ...)
```

The functions return the number of characters written, or a negative value if an error occurred.

## 2. Format string

The format string is of the form

```
% [flags] [field_width] [.precision] [length_modifier] conversion_character
```

where components in brackets **[]** are optional. The minimum is therefore a **%** and a conversion character (e.g. **%i**).

## 3. Flags

These can be in any order.

Flag	Meaning
-	The output is left justified in its field, not right justified (the default).
+	Signed numbers will always be printed with a leading sign (+ or -).
<i>space</i>	Positive numbers are preceded by a space (negative numbers by a - sign).
0	For numeric conversions, pad with leading zeros to the field width.
#	An alternative output form. For <b>o</b> , the first digit will be '0'. For <b>x</b> or <b>X</b> , " <b>0x</b> " or " <b>0X</b> " will be prefixed to a non-zero result. For <b>e</b> , <b>E</b> , <b>f</b> , <b>F</b> , <b>g</b> and <b>G</b> , the output will always have a decimal point; for <b>g</b> and <b>G</b> , trailing zeros will not be removed.

## 4. Field width

The converted argument will be printed in a field at least this wide, and wider if necessary. If the converted argument has fewer characters than the field width, it will be padded on the left (or right, if left adjustment has been requested) to make up the field width. The padding character is normally ' ' (space), but is '0' if the zero padding flag (**0**) is present.

If the field width is specified as \*, the value is computed from the next argument, which must be an **int**.

## 5. Precision

A dot '.' separates the field width from the precision.

If the precision is specified as \*, the value is computed from the next argument, which must be an **int**.

Conversion	Meaning
<b>s</b>	The maximum number of characters to be printed from the string.
<b>e, E, f</b>	The number of digits to be printed after the decimal point.
<b>g, G</b>	The number of significant digits.
<b>d, i, o, u, x, X</b>	The minimum number of digits to be printed. Leading zeros will be added to make up the field width.

## 6. Length modifier

Character	Meaning
<b>h</b>	The value is to be displayed as a <b>short</b> or <b>unsigned short</b> .
<b>l</b>	For <b>d, i, o, u, x</b> or <b>X</b> conversions: the argument is a <b>long</b> , not an <b>int</b> .
<b>L</b>	For <b>e, f, g</b> or <b>G</b> conversions: the argument is a <b>long double</b> .

## 7. Conversion character

Character	Meaning
<b>d, i</b>	Display an <b>int</b> in signed decimal notation.
<b>o</b>	Display an <b>int</b> in unsigned octal notation (without a leading <b>0</b> ).

<b>u</b>	Display an <b>int</b> in unsigned decimal notation.
<b>x, X</b>	Display an <b>int</b> in unsigned hexadecimal notation (without a leading <b>0x</b> or <b>0X</b> ). <b>x</b> gives lower case output, <b>X</b> upper case.
<b>c</b>	Display a single <b>char</b> (after conversion to <b>unsigned int</b> ).
<b>e, E</b>	Display a <b>double</b> or <b>float</b> (after conversion to <b>double</b> ) in scientific notation. <b>e</b> gives lower case output, <b>E</b> upper case.
<b>f</b>	Display a <b>double</b> or <b>float</b> (after conversion to <b>double</b> ) in decimal notation.
<b>g, G</b>	<b>g</b> is either <b>e</b> or <b>f</b> , chosen automatically depending on the size of the value and the precision specified. <b>G</b> is similar, but is either <b>E</b> or <b>F</b> .
<b>n</b>	Nothing is displayed. The corresponding argument must be a pointer to an <b>int</b> variable. The number of characters converted so far is assigned to this variable.
<b>s</b>	Display a string. The argument is a pointer to <b>char</b> . Characters are displayed until a ' <b>\0</b> ' is encountered, or until the number of characters indicated by the precision have been displayed. (The terminating ' <b>\0</b> ' is not output.)
<b>p</b>	Display a pointer (to any type). The representation is implementation dependent.
<b>%</b>	Display the <b>%</b> character.

[David C. Hamill](#)

[D.Hamill@surrey.ac.uk](mailto:D.Hamill@surrey.ac.uk)

Last updated 11/28/2004 05:24:46