

Java's *printf()* is a method that is used for formatting data output and is a combination of *String.format()* and *out.print()*. The types of formatting include:

- flags
- width
- precision
- conversions

General Equation: %[*flag*][*width*][.*precision*][*conversion*]

Flags: plus(+) or minus(-) sign, zero-padding, comma delimiter, and left justify.

Width: minimum amount of spaces data takes up.

Precision: amount of digits after decimal (only applies to floating points).

Conversion: decimal integer(d), floating point(f), char(c), String(s), boolean(b), and hashCode(h).

Decimal Integer – *byte, short, int, long*

Floating Point – *float, double*

[\[Example\]](#) [\[Example\]](#)

[\[Resource\]](#) [\[Resource\]](#)

[\[Resource\]](#) [\[Resource\]](#)

General

%c	character
%C	converts to uppercase character (if not already)
%d	decimal integer (base 10)
%e	scientific notation
%E	scientific notation with a capital 'E'
%f	floating-point number
%i	integer (base 10)
%b	converts to boolean
%B	converts to uppercase boolean

%o	octal number (base 8)
%s	a string of characters
%S	converts to a string of uppercase characters (if not already)
%u	unsigned decimal integer
%h	converts to hashcode
%H	converts to uppercase hashcode
%x	number in hexadecimal (base 16)
%%	prints a percent sign
\%	prints a percent sign

Decimal Integer

%0Xd	zero-fill for <i>X</i> digits
%Xd	right justify for <i>X</i> digits
%-Xd	left justify for <i>X</i> digits
%+d	adds plus sign(+) to positive integers, minus sign for negative integers(-)
% d	prints minus sign(-) if integer is negative, prints a space otherwise
%,d	uses comma delimiter between every 3 digits (ex: 1,000)

Floating Point

%.Yf	prints <i>Y</i> positions after decimal
%Xf	takes up <i>X</i> spaces
%0X.Yf	zero-fills

<code>%-X.Yf</code>	left justifies
<hr/>	
String	
<hr/>	
<code>%Xs</code>	formats string for a minimum of <i>X</i> spaces
<hr/>	
<code>%-Xs</code>	left justify
<hr/>	
Special Characters	
<hr/>	
<code>\a</code>	audible alert
<hr/>	
<code>\b</code>	backspace
<hr/>	
<code>\f</code>	form feed
<hr/>	
<code>\n</code>	newline or linefeed
<hr/>	
<code>\r</code>	carriage return
<hr/>	
<code>\t</code>	tab
<hr/>	
<code>\v</code>	vertical tab
<hr/>	
<code>\\</code>	backslash
<hr/>	