

Python String Formatting Reference Sheet

`{:[fill]align[sign][#]<width>.<precision>[bdeEgffGnosxX%]}`

Where `[align]` is one of `<` (left align, default for most things), `>` (right align, default for numbers) or `^` (centered). Any character placed in front of `[align]` is considered a `[fill]` and is used to pad the output up to the required size. `[sign]` is either `+` (show sign for both positive or negative numbers), `-` (show sign for negative numbers only, default) or `=` (show sign for negative numbers and a space for positive numbers). `#` is valid only for integers and is used to prepend information about the number's base.

The final character is the presentation type. This defaults to `s` (string), other options are described here:

Fixed Point [fF%]

`<width>` and `<precision>` give the total width of the number and the number of decimal places, respectively. There is no difference between `f` and `F`. Using `%` multiplies the number by 100 and puts a `%` sign on the end.

Value	Format	Output
<code>math.pi</code>	<code>{:7.2f}</code>	3.14
<code>math.pi</code>	<code>{:9.7f}</code>	3.1415927
<code>math.pi</code>	<code>{:<7.2f}</code>	3.14
<code>math.pi</code>	<code>{:0>7.2f}</code>	0003.14
<code>math.pi</code>	<code>{:+7.2f}</code>	+3.14
<code>math.pi</code>	<code>{:_^11.2%}</code>	314.16%

Integers [dnboxX]

`<width>` gives the total width of the number. The presentation types are `d` (decimal), `b` (binary), `o` (octal) and `x` (hexadecimal). `#` prepends the presentation type `'0b'` (binary), `'0o'` (octal) or `'0x'` (hexadecimal). `X` is the same as `x` but it uses uppercase letters. Using a comma in place of a period formats with comma separators. `n` is the same as `d` but it will insert locale dependent separators instead of commas.

Value	Format	Output
255	<code>{:0>10d}</code>	0000000255
123	<code>{:#b}</code>	0b1111011
1000000	<code>{:8,d}</code>	1,000,000

Scientific Notation [eE]

`<width>` and `<precision>` give the total width of the number and the number of decimal places, respectively.

Value	Format	Output
1000000	<code>{: >+13e}</code>	+1.000000e+06
1000000	<code>{: >+13.2e}</code>	+1.00e+06
1e10	<code>{:13.8e}</code>	1.00000000e+10

General Notation [gG]

General notation tries to figure out from the number the best representation. `g` and `G` are the same except that `G` uses a capital letter E in scientific notation. Note that `g` overrides the precision argument.

Value	Format	Output
100	<code>{:g}</code>	100
1000000	<code>{:g}</code>	1e6
1e10	<code>{:13.8G}</code>	1e+10