

Get started

Open in app

<{Blake+Codez}>/>

35 Followers

About

Follow

Formatting Numbers in Python Cheat Sheet

 <{Blake+Codez}>/> Feb 22 · 3 min read

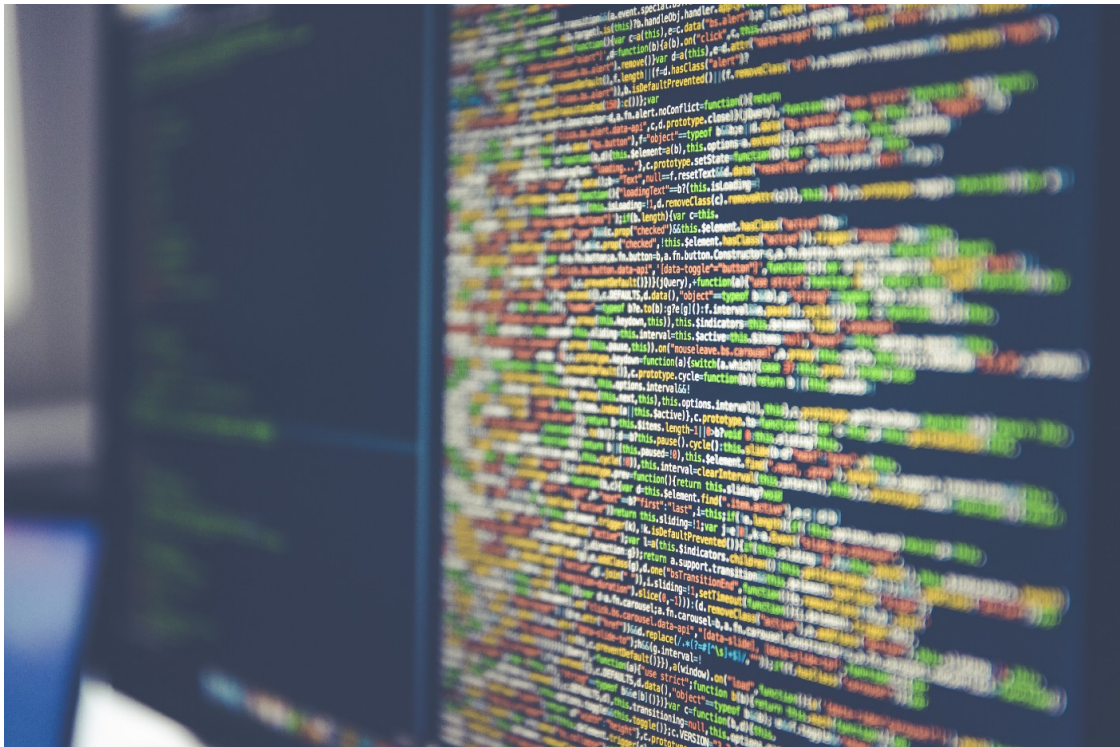


Photo by Markus Spiske on Unsplash

Here is a simple cheat sheet when it comes to formatting outputs with the print function — print(), specifically numbers and data.

Rounding Decimals: (floating point numbers)

To round a number simply use the format() function to do so. The first argument will be the number you are wanting to format, and the second argument will be how you'd like

[Get started](#)[Open in app](#)

```
format(12.6253, '.2f');
```

```
# Output:
```

```
#
```

```
# 12.63
```

use `format()` and `'.2f'`

You can pass in a comma (,) to also add a comma to your number format like so:

```
format(12000.6253, ',.2f');
```

```
# Output:
```

```
#
```

```
# 12,000.63
```

Format in Scientific Notation

If you need to format your output into a scientific notation format, use the `format()` function in conjunction with `'e'`.

```
# Example 1  
format(12345.6789, 'e');
```

```
# Output:
```

```
# 1.2345678e+04
```

```
# Example 2
```

```
format(12345.6489, '.2e');
```

```
# Output:
```

```
# 1.23e+04
```

Get started

Open in app



Insert Comma Separator — Integer or Floating Point

Adding a comma is useful for handling large number especially in currency amounts. To use this with a **floating point** number simply pass 'f' for floating point as the number type. To use this with a **integer** or whole number, pass 'd' for digit as the number type.

```
# Example 1
format(123456789, ',d');
#
# Output:
#      123,456,789

# Example 2
format(12345.6489, ',.2f');
#
# Output:
#      12,345.65
```

Formatting a Percentage

To format a decimal (floating point) value as a percentage you simply pass the % character into the format() function as an argument.

```
# Example 1
print(format(.5, '%'));
#
# Output:
#      50.000000%

# Example 2
print(format(.75, '.0%'));
#
# Output:
#      75%
```

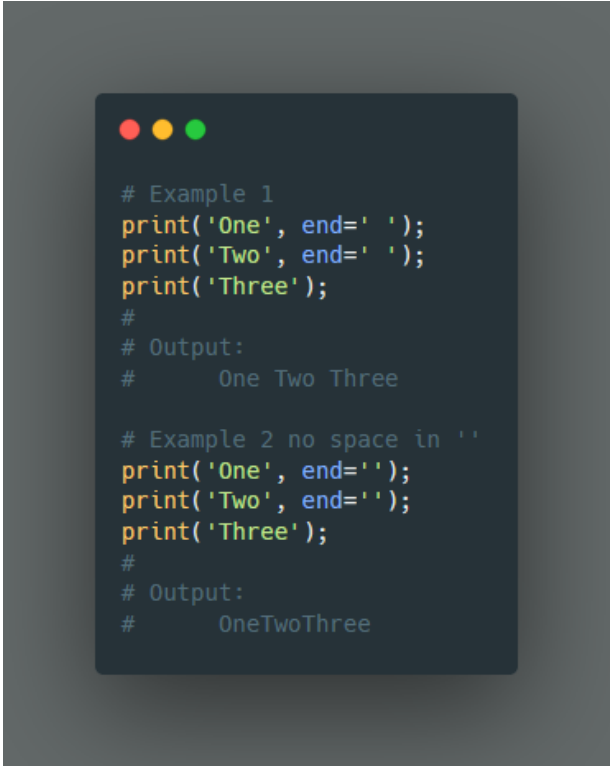
Get started

Open in app



Suppress Print's Newline

When using the `print()` function, python creates a new line each time you use it. To suppress this feature simply at the `end=` to the last argument of the `print()` function.



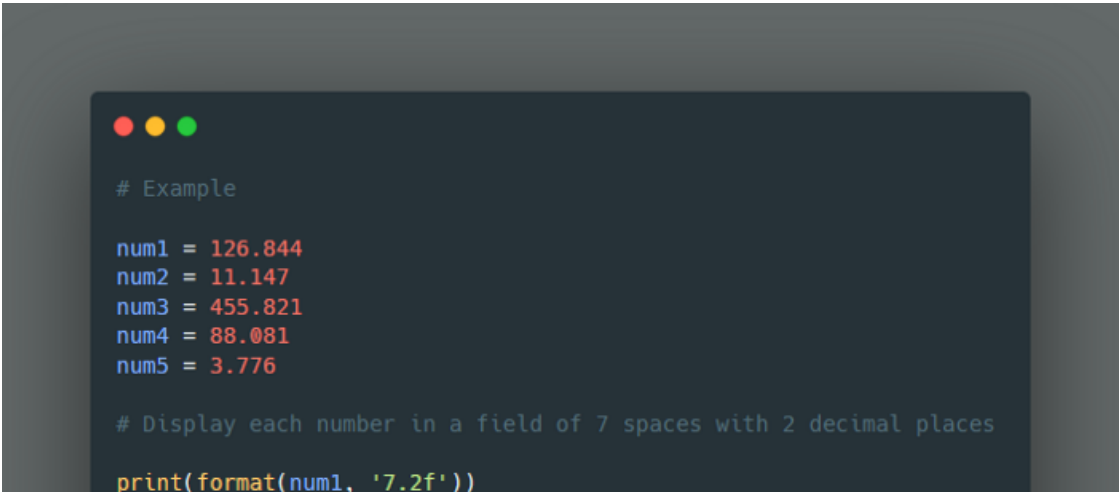
```
# Example 1
print('One', end=' ');
print('Two', end=' ');
print('Three');
#
# Output:
#      One Two Three

# Example 2 no space in ''
print('One', end='');
print('Two', end='');
print('Three');
#
# Output:
#      OneTwoThree
```

If you don't put a space in between the 'quotes', you won't get a space.

Minimum Field Width

You can specify a minimum field width inside a `format()` function which will make a field `x` spaces wide; `x` being the number of spaces. This is useful for aligning columns to show decimal numbers.



```
# Example

num1 = 126.844
num2 = 11.147
num3 = 455.821
num4 = 88.081
num5 = 3.776

# Display each number in a field of 7 spaces with 2 decimal places

print(format(num1, '7.2f'))
```

[Get started](#)[Open in app](#)

```
print(format(nums, '7.2f'))
```

```
# Output:
```

```
#      126.84
```

```
#      11.15
```

```
#     455.82
```

```
#      88.08
```

```
#       3.78
```

[Python](#)[Programming](#)[Coding](#)[Technology](#)[Data Science](#)[About](#) [Write](#) [Help](#) [Legal](#)[Get the Medium app](#)