- insert {} into a string to indicate where values should go
- use .format([values]) after the string itself or the variable that has the string to insert values
- use {{[commands]}} to format
  - .fillchar[[align][width],[decimal places][type]
  - Fillchar – what char to put in open places
  - Align (right >, left < or center ^)
  - Width – how many total spots to use
  - Decimal places – how many digits after the decimal (better than the round function)
  - Type – use d for integers, f for floats, and % for percentages
  - More fields are available if you look
  - All fields are optional if you don’t need them

```python
w = “World”
my_string = “Hello {}” .format(w)
print(my_string)
```
This will print: Hello World

```python
my_string = “a = {}, b = {}, c = {}”
my_string = my_string.format(1, 2, 3)
print(my_string)
```
will print: a = 1, b = 2, c = 3

```python
my_string = “a = {1}, b = {0}, c = {2}”
my_string = my_string.format(3, 4, 5)
print(my_string)
```
will print: a = 4, b = 3, c = 5

```python
my_string = “a = {:.3f}, b = {0:0>6}, c = {:.1%}”
my_string = my_string.format(1.2, 0.3)
print(my_string)
```
will print: a = 1.200, b = 000002, c = 30.0%

Print with commas example:
```python
my_string = “{:,}” .format(123456789)
print(my_string)
```
will print: 123,456,789