**kwargs

```
def func(**kwargs):
    print(kwargs)

func(x=100, y=200)
{'x': 100, 'y': 200}
```

We can also use it in conjunction with ***args**:

```
def func(*args, **kwargs):
    print(args)
    print(kwargs)

func(1, 2, a=100, b=200)
(1, 2)
{'a': 100, 'b': 200}
```

Note: You cannot do the following:

```
def func(*, **kwargs):
    print(kwargs)
  File "<ipython-input-9-330c63b7f22e>", line 1
    def func(*, **kwargs):
               ^
SyntaxError: named arguments must follow bare *
```

There is no need to even do this, since ****kwargs** essentially indicates no more positional arguments.

```
def func(a, b, **kwargs):
    print(a)
    print(b)
    print(kwargs)

func(1, 2, x=100, y=200)
1
2
{'x': 100, 'y': 200}
```

Also, you cannot specify parameters **after ****kwargs** has been used:

```
def func(a, b, **kwargs, c):
    pass
  File "<ipython-input-12-ffdc3153243b>", line 1
    def func(a, b, **kwargs, c):
                             ^
SyntaxError: invalid syntax
```

If you want to specify both specific keyword-only arguments and ****kwargs** you will need to first get to a point where you can define a keyword-only argument (i.e. exhaust the positional arguments, using either ***args** or just ***)

```
def func(*, d, **kwargs):
```

```
    print(d)
    print(kwargs)

func(d=1, x=100, y=200)
1
{'x': 100, 'y': 200}
```