Keyword Arguments

Functions can also accept keyword arguments! They can actually accept both regular arguments and keyword arguments. What this means is that you can specify which keywords are which and pass them in. You saw this behavior in a previous example.

```
def keyword_function(a=1, b=2):
    return a+b

result = keyword_function(b=4, a=5)
print(result) # 9
```

You could have also called this function without specifying the keywords. This function also demonstrates the concept of default arguments. How? Well, try calling the function without any arguments at all!

```
def keyword_function(a=1, b=2):
    return a+b

result = keyword_function()
print(result) # 3
```

The function returned the number 3! Why? The reason is that **a** and **b** have default values of 1 and 2 respectively. Now let's create a function that has both a regular argument and a couple keyword arguments:

```
def mixed_function(a, b=2, c=3):
    return a+b+c

result = mixed_function(1, b=4, c=5)
print(result) # 10
print(result) # 10

result = mixed_function(1)

print(result) # 6

result = mixed_function(b=4, c=5)
# TypeError: mixed_function() missing 1 required positional argument: 'a'
```

There are 3 example cases in the above code. Let’s go over each of them.

In our first example, we call the mixed function with 3 values, naming two of them. This works and gives us the expected result, which was 1+4+5=10.

In the second example shows what happens if we only call the function by passing in just one value…the one that didn’t have a default. This also works by taking the “1” and adding it to the two default values of “2” and “3” to get a result of “6”! Isn’t that cool?

In our third example, we try calling our function using just the keyword arguments. This will give us a confusing error. The Traceback says that our function accepts at least one argument, but that two were given. What’s going on here? The fact is that the first argument is required because it’s not set to anything, so if you only call the function with the keyword arguments, that causes an error.