

Passing Arguments to a Function

Now we're ready to learn about how to create a function that can accept arguments and also learn how to pass said arguments to the function. Let's create a simple function that can add two numbers together:

```
def add(a, b):  
    return a + b  
  
print(add(1, 2)) # 3
```



All functions return something. If you don't tell it to return something, then it will return `None`. In this case, we tell it to return **a + b**. As you can see, we can call the function by passing in two values. If you don't pass enough or you pass too many arguments, then you'll get an error:

```
def add(a, b):  
    return a + b  
  
add(1)  
  
# TypeError: add() takes exactly 2 arguments (1 given)
```



You can also call the function by passing the name of the arguments:

```
def add(a, b):  
    return a + b  
  
result = add(a=2, b=3)  
print(result) # 5  
  
total = add(b=4, a=5)  
print(total) # 9
```





You'll notice that it doesn't matter what order you pass them to the function as long as they are named correctly. In the second example, you can see that we assign the result of the function to a variable named **total**. This is the usual way of calling a function as you'll want to do something with the result. You are probably wondering what would happen if we passed in arguments with the wrong names attached. Would it work? Let's find out:

```
def add(a, b):  
    return a + b  
add(c=5, d=2)
```



```
# TypeError: add() got an unexpected keyword argument 'c'
```



Whoops! We received an error. This means that we passed in a keyword argument that the function didn't recognize. Coincidentally, keyword arguments are our next topic!