

3.8: Return Statement Walkthrough

Functions are more powerful than simply printing text to the screen. We can also return values back to whatever was calling the function. The value returned is often the result of a calculation, or another process which produces a response. To be able to pass back the response, we use the return keyword.

Adapt your Python file from the previous example, so that it shows the following:

def multiply\_four\_numbers():

num\_1 = 5

num\_2 = 6

num\_3 = 7

num\_4 = 8

return num\_1\*num\_2\* num\_3\*num\_4

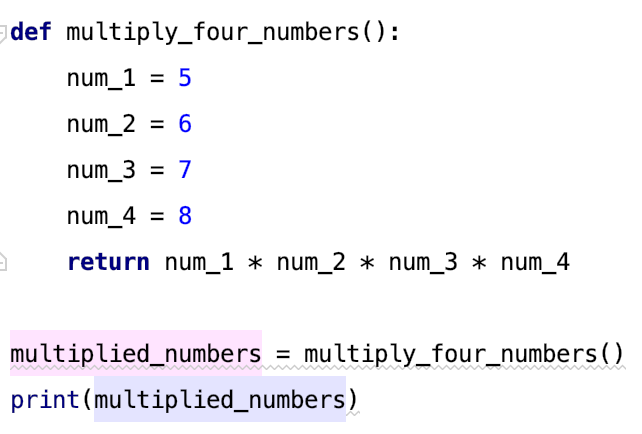
Note: I have removed the inputs in this example, technically, functions should only have a single purpose or task (in this case the task will be the multiplication). By including the input, it has a second purpose. We will see in later weeks how this is overcome.

Now that we have our function, we need to write the code required to call it and use the result. When we call the function, we need to do something with the output. In this case (and in most cases), we have created a variable which we will then assign the result to. Enter the following code beneath your function:

multiplied\_numbers = multiply\_four\_numbers()

print(multipliedNumbers)

Once completed, this section of the Python file should be:



When you run the program, you get the following output:

