

3.10: Default Parameters Walkthrough

Default parameters provide the programmer with the opportunity to define default values for one or more parameters of a function. This means that these parameters do not need to be specified when the function is run. In the case where arguments are not passed to these parameters, the default value will be used.

Let’s adapt our multiply 4 numbers function so that it has 2 required parameters and 2 defaults. We will start with the required and leave them as they are. For the default parameters (num\_3 and num\_4) we need to define the default value. To achieve this, we use the following format:

parameter\_name = <<default\_value>>

The below code shows how this is implemented in practice. Update your function to match this code:

def multiply\_four\_numbers(num\_1, num\_2, num\_3=1, num\_4=1):

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

You will notice that the default parameters are at the end of the parameter list. This is a requirement – you must have parameters which need a value at the beginning and defaults at the end.

multiplied\_numbers = multiply\_four\_numbers(5, 6)

print(multiplied\_numbers)

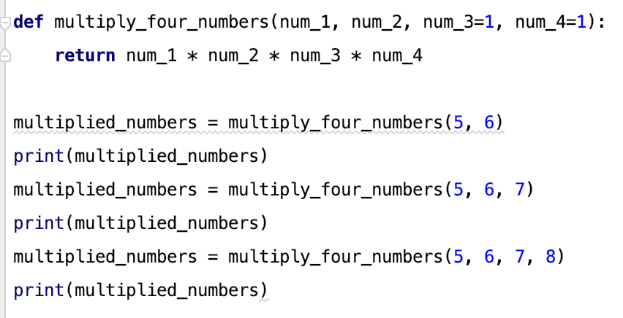
multiplied\_numbers = multiply\_four\_numbers(5, 6, 7)

print(multiplied\_numbers)

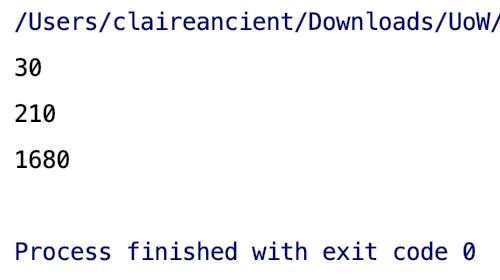
multiplied\_numbers = multiply\_four\_numbers(5, 6, 7, 8)

print(multiplied\_numbers)

Therefore, your Python program should be as follows:



This produces the following output:



As you can see from this output, we have been able to use the function with default parameters three ways:

1. With two arguments (the following values will be used: num\_1 = 5; num\_2 = 6; num\_3 = 1; and num\_4 = 1)
2. With three arguments (the following values will be used: num\_1 = 5; num\_2 = 6; num\_3 = 7; and num\_4 = 1)
3. With four arguments (the following values will be used: num\_1 = 5; num\_2 = 6; num\_3 = 7; and num\_4 = 8)