Honors Class07 Activity: Creating Python Functions

Team Members: ____

____

Learning Objectives

- Understanding how to create and use Python functions
- The difference between parameters and arguments
- The difference between local and global variables
- Return statements
- Boolean variables
- Keyword arguments and default parameter values

In the last few classes, you've been using functions that other people have created. Today, you'll learn how to create your own functions. This is a very important skill, because it allows you to write code that is reusable and easier to read.

Once again, there is a starter Jupyter notebook under "class activities." Open the one for class_07.

Problem 1: Add a line to the function print_any_dad_joke() that prints "LOL" like this:

```
def print_one_dad_joke():
    print("What did one plant say to the other? Aloe! Long thyme no see.")
    print("LOL")
```

Call the function to see the result.

Now remove the indent from the second print statement, run the cell, and call the function again. **What happens? Why do you think that is?**

Adding a parameter to a function

You don't want have to redefine the function every time you change the joke, so we will use a parameter to allow us to pass in a new joke each time we call the function.

Call the function print_any_dad_joke() with a new joke: "What brand of underwear do chemists wear? Kelvin Klein."

Terminology Alert: Parameters vs. Arguments

The variables that are defined in the function definition are called parameters. The values that you pass into the function when you call it are called arguments.

- Parameter → the variable name in the function definition

- Argument → the actual value passed into the function when you call it

Problem 2: In our Function 2 example, what is the parameter? What is the argument?

Creating a function with one parameter and one return value

The Function 3 example returned a value. Create a function called `square_it` that takes one parameter (a number) and returns the square of that number. Test your function.

Problem 3: What happens if you call your function without an argument? Why do you think that is?

In Function 4, we defined a function that takes two parameters. Create a function called `add_numbers` that takes two parameters (numbers) and returns the sum of those numbers. Test your function.

Problem 4: What happens if you pass in strings to `add_numbers` instead of numbers as arguments? Why?

Problem 5: What happens if you pass in more than two arguments? Why?

Creating a function with multiple input parameters and multiple return values

Problem 6: Create af function that calculates the area and circumference of a rectangle given its length and width as parameters.

The function should return both the area and circumference. If you want an extra challenge, return the length of the diagonal as well. **Test your function for a rectangle with length 5 and width 3 and report the results.**

DIGRESSION: Boolean variables

A Boolean variable is a variable that can only take on two values: True or False. They are tested using comparison operators (==, !=, <, >, <=, >=) and logical operators (and, or, not). Here is a simple example of a boolean variable:

```
a = 5
b = 10
```

```
is_a_greater_than_b = a > b
print(is_a_greater_than_b)
```

This will be False because 5 is not greater than 10. **Try it.**

Boolean variables are often used in conditional statements (if this do that) to control the flow of a program, a topic we will cover in more detail in a future class, but you will see an example in Function 5.

End Digreession

Function keyword arguments and default parameter values

When you define a function, you can specify default values for parameters. This allows you to call the function without providing arguments for those parameters, and the default values will be used instead.

Here is another example, a function called `greet` that takes one parameter, `name`, and has a default value of “World”. The function prints a greeting message that says “Hello, [name]!”

```
def greet(name="World"):
    print("Hello, " + name + "!")
```

Problem 7: Call the `greet` function without an argument. What is the output? Now call it with your name as an argument. What is the output?

Putting it all together

Look at the final example in the notebook, Function 6, that combines many of the concepts you’ve learned today.

Problem 8: What are the keywords arguments used for in this function? What are their default values?

Extra Practice

Create a function called `fahrenheit_to_celsius` that takes one parameter (a temperature in Fahrenheit) and returns the temperature in Celsius. The formula to convert Fahrenheit to Celsius is: C = (F - 32) * 5/9

Test your function with the following Fahrenheit values: 32, 68, 100, and 212. Report the results.