

Honors Class07 Activity: Creating Python Functions

Team Members: _____

Learning Objectives

- Understanding how to create and use Python functions
- Understanding the difference between parameters and arguments
- Understanding the difference between local and global variables
- Understanding how to use return statements

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_one_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?

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 a number as an argument? Why?

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