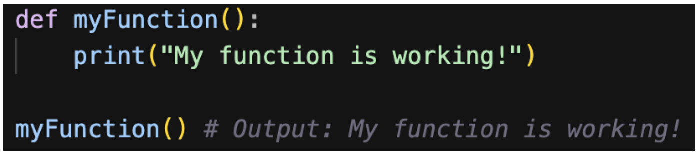
# Intro to Python Functions

Python functions allow you to abstract your code by allowing you to execute repetitive procedures quickly without having to rewrite much code. You will declare your function and call it later. With the usage of parameters and returning, you can customize your procedures to work for specific values, and be able to use outputs from those functions later on.

## #1 – Declaring Functions

The **def** keyword allows you to give a function with a name (no spaces, can't start with number, or be a keyword). A function name always has a pair of parentheses and a colon. The code block MUST be indented.



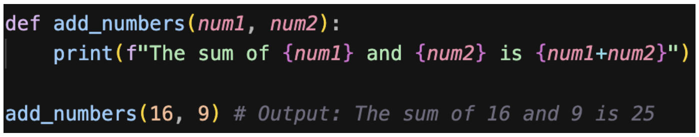
```python
def myFunction():
    print("My function is working!")

myFunction() # Output: My function is working!
```

Again, this will run if and only if we call **myFunction()**.

Try It: Write a function **addition()** that prints the sum of 2 and 3.

## #2 – Parameters

***Parameters*** replace hard coded values in your program so your procedure can handle the usage of various values in your program. When calling the function, you will supply the necessary ***arguments*** that will take the place of the parameters when the function is called.



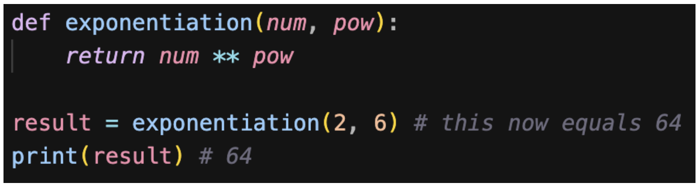
```python
def add_numbers(num1, num2):
    print(f"The sum of {num1} and {num2} is {num1+num2}")

add_numbers(16, 9) # Output: The sum of 16 and 9 is 25
```

What do you notice about what the program does with the supplied values? How does this make the program more usable?

Try It: Prompt the user to enter two numbers. Then write a function that prints the product of those numbers. *Bonus: handle invalid inputs.*

## #3 – Returning

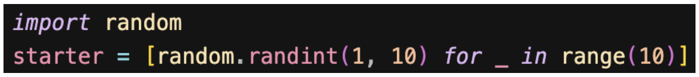
Ask yourself, if we can print out our desired outputs, why might we return values? Look at the code snippet below.



```python
def exponentiation(num, pow):
    return num ** pow

result = exponentiation(2, 6) # this now equals 64
print(result) # 64
```

The difference here is now we can assign the function's return value to a variable. Unlike before, we could only print the value, but we wouldn't be able to access it later on.

Spiral Review: Given the following starter list, create a function that returns a list with only multiples of 3.



```python
import random
starter = [random.randint(1, 10) for _ in range(10)]
```