Python Functions

**Getting Started**

A function is a block of organized, reusable code that is only run once called. They can be used for specific tasks or reducing repetitive code. They let developers avoid having repeating blocks of code in a program and overall improve the organization of a program by allowing functions to be separated on more than one file.

**Syntax – Creating a Function**

1. Define the function – def keyword
2. Give the function a name – any appropriate name
3. Closing brackets where parameters are passed through – empty if none

def greeting(name):

print(“Hello”, name)

**Syntax – Calling a Function**

General Rules

1. Function name followed by parenthesis
2. For functions with parameters, you can pass in variables or any other data via the same method.

greeting(“Adam”)

If the function is returning a value, we can set a variable to be assigned to the data that is being returned from the function.

def square(x):

return x\*x

number\_yet\_to\_be\_squared = square(number\_yet\_to\_be\_squared)

**Parameters and Return**

Functions can take in data, known as parameters. Think of parameters as temporary variables that can be used throughout the function. Your function is a set of instructions that these parameters follow that can later be mimicked by variables in your main code when you call the function. The return statement allows the function to return data as a result. No code after the return statement is executed.

**Common Errors**

Variables out of scope – we cannot pass through variables from our main code to within our function. Think of functions as separate from our main code and only newly declared variables inside the function and parameters can be used.

Return errors – for functions with parameters, you must use a return statement even if the return is not used.