Functions

In Python, a function is a group of related statements that perform a specific task

Functions help break our program into smaller and modular chunks.

As our program grows larger and larger, functions make it more organized and managable.

Furthermore, it avoids repetition

Creating clean repeatable code is a key part of becoming an effective programmer

Functions allow us to create blocks of code that can be easily executed many times, without needing to constantly rewrite the entire block of code.

What is a function?

Fomarlly, a function is a useful device that groups together a set of statements so they can be run more than once.

They also let us specify parameters that can serve as inputs to the functions.

On a more fundamental level, functions allow us to not have to repeatedly write the same code again and again.

If you remember back toi the lessons on strings and lists, remember that we used a function len() to get the length of a string.

Since checking the length of a sequence is a common task you would want to write a function that can do this repeatedly at command

FUnctions will be one of the most basic levels of reusing code in Python, and it will also allow us to start thinking of program design

Why even use Functions?

Put simply, you should use functions when you plan on using a block of code multiple times. The function will allow you to call the same block of code without having to write it multiple times. This in turn will allow you to create more complex Python scripts.

In [1]:
```
Function Topics

- def keyword
- simple example of a function
- calling a function with ()
- accepting parameters
- print versus return
- adding in logic inside a function
- multiple returns inside a function
- adding in loops inside a function
- tuple unpacking
- interactions between the functions
```

```
  File "C:\Users\Keegz\AppData\Local\Temp/ipykernel_12424/4151068234.py", line 1
    Function Topics
             ^
alid syntax
```

In [ ]: