* Why should we use functions at all?

**Program development made easy :** Work can be divided among project members thus implementation can be completed in parallel.

**Program testing becomes easy :** Easy to locate and isolate a faulty function for further investigation

**Code sharing becomes possible :** A function may be used later by many other programs this means that a c programmer can use function written by others, instead of starting over from scratch.

**Code re-usability increases :** A function can be used to keep away from rewriting the same block of codes which we are going use two or more locations in a program. This is especially useful if the code involved is long or complicated.

**Increases program readability :** It makes possible top down modular programming. In this style of programming, the high level logic of the overall problem is solved first while the details of each lower level functions is addressed later. The length of the source program can be reduced by using functions at appropriate places.

**Function facilitates procedural abstraction :** Once a function is written, it serves as a black box. All that a programmer would have to know to invoke a function would be to know its name, and the parameters that it expects.

**Functions facilitate the factoring of code :** Every C program consists of one main( ) function typically invoking other functions, each having a well-defined functionality.

* How to define/declare a function?

Use the keyword “**def**” followed by the function name.

* How to call/use a function?

We have to call the function by using the function’s name with a list of values used to assign to the function’s parameters.

* What is return, why and how do we use it?

 The **return** statement causes your function to exit and hand back a value to its caller. The point of functions in general is to take in inputs and return something. The**return** statement is used when a function is ready to return a value to its caller.

* Do we have to use return in **every** function?

No, we don’t have to use return in every function.

Because in Python, if you don’t use return, the program will return a value to None. In other languages, there is a type of functions that is not executed for its resulting value, but is executed because it does something useful. It’s called “**void functions**” in languages like Java, C#, C and C++ , other languages like Pascal call it a procedure.

* What are function arguments/parameters, why and how we use it?

**A parameter** is the variable which is part of the method’s signature (method declaration).

**An argument** is an expression used when calling the method.

We put **parameters** in () in the ends of **def function\_name** and use the parameters below that.

* How to use function from a different file other than our currently working file?

We have to import the **sys** library and then use “**sys.path.append”** to include the pathname of file where the function is located.