What Is A Function?

A Function is a sequence of statements/instructions that performs a particular task. A function is like a black box that can take certain input(s) as its **parameters** and can output a value after performing a few operations on the parameters. A function is created so that one can use a block of code as many times as needed just by using the name of the function.

Why Do We Need Functions?

- **Reusability:** Once a function is defined, it can be used over and over again. You can call the function as many times as it is needed. Suppose you are required to find out the area of a circle for 10 different radii. Now, you can either write the formula πr^2 10 times or you can simply create a function that takes the value of the radius as an input and returns the area corresponding to that radius. This way you would not have to write the same code (formula) 10 times. You can simply invoke the function every time.
- **Neat code:** A code containing functions is concise and easy to read.
- **Modularisation:** Functions help in modularizing code. Modularization means dividing the code into smaller modules, each performing a specific task.
- **Easy Debugging:** It is easy to find and correct the error in a function as compared to raw code.

Defining Functions In Python

A function, once defined, can be invoked as many times as needed by using its name, without having to rewrite its code.

A function in Python is defined as per the following syntax:

```
def <function-name>(<parameters>):
      """ Function's docstring """
      <Expressions/Statements/Instructions>
```

- Function blocks begin with the keyword `def` followed by the **function name** and **parentheses** (( )).
- The input **parameters** or **arguments** should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function is optional - the documentation string of the function or **docstring**. The **docstring** describes the functionality of a function.
- The code block within every function starts with a **colon** (:) and is **indented**. All statements within the same code block are at the same indentation level.
- The `return` statement exits a function, optionally passing back an expression/value to the function caller.

Let us define a function to add two numbers.

```
def add(a,b):
      return a+b
```

The above function returns the sum of two numbers **a** and **b**.

The `return` Statement

A `return` statement is used to end the execution of the function call and it “returns” the result (value of the expression following the `return` keyword) to the caller. The statements after the return statements are not executed. If the `return` statement is without any expression, then the special value `None` is returned.

In the example given above, the sum `a+b` is returned.

Note: In Python, you need not specify the return type i.e. the data type of returned value.