**Functions**

**Topics Covered -**

* Functions
* Return keyword

### **Functions**

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

Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organised and manageable.

Furthermore, it avoids repetition and makes the code reusable.

**Syntax -**

def function\_name(parameters):

statement(s)

* Above shown is a function definition that consists of the following components.
* First, Keyword def marks the start of the function header.
* Second, a function name to uniquely identify the function. Function naming follows the same rules of writing identifiers in Python.
* Parameters (arguments) through which we pass values to a function. They are optional.
* A colon (:) to mark the end of the function header.
* One or more valid python statements that make up the function body. Statements must have the same indentation level (usually four spaces).
* An optional return statement to return a value from the function.

Example -

def greet(name):

"""

This function greets to

the person passed in as

a parameter

"""

print("Hello, " + name + ". Good morning!")

### **How to call a function in python?**

Once we have defined a function, we can call it from another function, program or even the Python prompt. To call a function, we simply type the function name with appropriate parameters.

**Syntax -**

greet('Paul')

Note: Try running the above code in the Python program with the function definition to see the output.

def greet(name):

print("Hello, " + name + ". Good morning!")

greet('Paul')

### **Return statement**

The return statement is used to exit a function and go back to the place from where it was called.

**Syntax -**

return [expression\_list]

This statement can contain an expression that gets evaluated, and the value is returned. If there is no expression in the statement or the return statement itself is not present inside a function, then the function will return the None object.

Example -

def absolute\_value(num):

if num >= 0:

return num

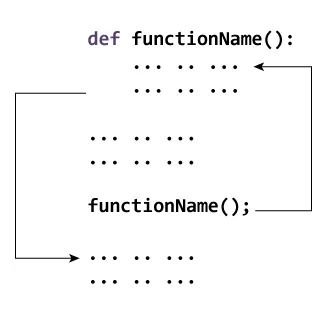
else:

return -num

print(absolute\_value(2))

print(absolute\_value(-4))

### **How Function works in Python?**



### **Types of Functions**

We can divide functions into the following two types:

* Built-in functions - Functions that are built into Python.
* User-defined functions - Functions defined by the users themselves.