**Function:**

Set of Statements that performs a task/ action. (Calculation processing)

* Built-in function.
* User defined function.

**User - defined – function:**

**Syntax:**

Def fun(parameters):

Task

Return(data)

Input ()

Input ()

Print (fun (input data))

Return Statement: useful to return result out of a function.

Return x

Return 0

Etc.

* **Parameters:** When functions are created some values are passed to it to perform action on those values for receiving those values we add variables with the function so we can use those variables to write the program in function these variables or values are called as parameters or formal arguments.

Ex:

Def funcName(Parameters Or formal arguments.)

* **Actual arguments:** Actual arguments are those values in a program that are passed to the parameters or the formal arguments. These actual arguments exist outside the function.

There are four types of actual arguments.

**Types of actual arguments:**

1. **Positional arguments:**

The arguments that are passed in a specific order.

Ex:

def attach(s1, s2): # function definition

attach('New', 'York') # positional arguments

1. **Keyword arguments:**

they are passed using parameter names.

Ex:

def grocery(item, price): # function definition

grocery(item='Sugar', price=50.75) # key word arguments

1. **Default arguments:**

declared in the parameter of the function.

1. **Variable length argument:**

they can store 0 or more values.

\*x this type of symbol is used for variable length argument.

variable length argument should be last in all arguments in the function.

def total(num, \*x):  
 tot=num+sum(x)  
 print("total=",tot)  
total(100,10)  
total(100,30,40,60,70)

function are first class object.

Object is a memory block. It contain specific type of data (ex: int, float, etc.).

X=10 (10 is the object of int class)

Function can be used exactly like object

X=10

X=func()

Func(10)

Func(func)

b

B is a inner function.

* Recursive function or recursion: is a function that calls itself.
* Function decorator: function decorator that modifies the result of another function.

1. A decorator function takes another function as its parameter.
2. Write another function inside the decorator function and call the fun passed to decorator.
3. Return inner function name.