**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.

**Imp:**

* Functions are first class object.
* We can assign function to the variable.
* We can pass function as parameter to another function.
* We can write function inside another function.
* We can return function from another function.

**Global and local variable:**

* The variable declared outside the function is called as global variable.
* And the variable declared inside the function is called as the local variable.
* Global variables can be used inside the function but the local variables cannot be used outside the function
* We can create global variable inside function using global keyword.
* Local variable dominates global variable inside the function.
* Y=globals()[‘x’] can be used to make the global variable x to dominate the local variable x.

**Types of function:**

* **Recursive function:**

A function calling itself repeatedly.

* **Decorator function:**

Decorator function is a function that is used to decorate or modify the result of another function.

Def décor() #this is used to decorate the function result.

**Syntax:**

Def décor(fun): #this is used to decorate the function result.

Fun is the function name of the function whose result we are modifying.

Def inner():

Res=fun() #result is stored in another variable to modify.

Res=Res+100 #now 100 is added into the result

Return res #return modified result

Return inner() #return inner function

Def function1():

Return 10

Print(décor(function1))#decor function is used to décor the original function.

* **Generator Function:**

Generator function is used to return sequence of numbers.

It uses the yield to return the sequence.