Python Fuction:

Function is a group of related statements that perform a specific task.

Functions provide better modularity for your application and a high degree of code reusing.

Functions help break our program into smallar and modular chunks. As our program grows larger and larger, function make it more organized and manageable.

It avoid repetition and makes code reusable.

Syntax:

def function_name(parameters):

```
"""
doc

"""

statement(s)
```

1. Keywords "def" marks the start of function header.
2. Parameters(arguments) through which we pass values to a function. these are optional.
3. A colon(:) to mark the end of the function header.
4. Doc string describe what the function does. This is optional.
5. "return" statement return a value form the function. This is optional.

Example:

```
def print_name(name):
    """
    The function prints the name
    """
    print("hello"+ str(name))
    print("My friend")
    print("good morning")
```

Function Call:

once we have defined a function, we can call it form anywhere.

```
print_name("Harsh")
```

Doc String:

The First string after the function header is called the docstring and is short the documentation string.

Although optional, documentation is a good programming practice, always document your code.

Doc string will be written in triple quotes so that docstring can extend up to multiple lines.

```
print(print_name.__doc__) # print docstring of the function.
```

Return Statement:

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

syntax:

```
return[expression]
```

The return statement can contain an expression which gets evaluated and the value is returned.

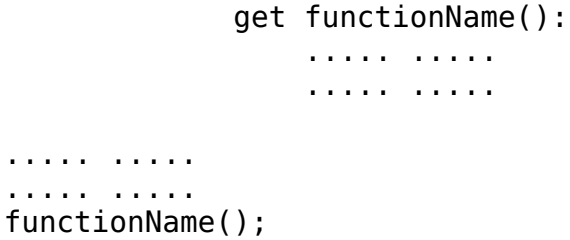
if there is no expression in the statement or the return statement itselt in not present inside a function, then the function will return None Object.

```
def get_sum(lst):
    """
    function returns the sum of all the elements in a list
    """
    #initialize sum
    sum = 0

    #iteration over list lst=[10,20,30,40,50,60]
    for num in lst:   # num =20
        sum += num    # sum  = 10 + 20 = 30
    return sum

s = get_sum([10,20,30,40,50,60])
print(s)
```

How Function work in python:



....
....

Scope and Life Time of Variable:

-> Scope of a variable is the portion of a program where the variable is recognized.

-> Variable defined inside a function is not variable from outside. Hence, They have a local scope.

-> Lifetime of a varible is the period throughout which the variable exits in the memory.

-> The Lifetime of variable inside a function is as long as the function executes.

-> variables are destroyed once we return from the function.

Example:

```
global_var = "This is global variable"

def test_life_time():
        """
        This function test the life time of a variable
        """
        local_var = "This is local variable"
        print(local_var)                # print local variable
local_var

        print(global_var)               # print global variable
global_var

# calling function
test_life_time()

# print global variable global_var
print(global_var)

# print local variable local_var
print(local_var)
```

```
This is local variable
This is global variable
This is global variable
Harsh raj Singh
```

Python Program to Print Highest Common factor(HCF) of two numbers:

```
def computeHCF(a,b):    # a = num1   b = num2
    """
    Computing HCF of two numbers
    """
    smaller =  b if a>b else a   # TRUE statement if text expression else False statement # a>b 24>16

    hcf = 1
    for i in range(1, smaller+1):        # i = 1,2,3,4---,16,17
        if(a % i == 0) and (b % i == 0):   # 24 %16 ==0 and 16%16 == 0
            hcf = i                        # hcf = 8
    return hcf

a = int(input("num1: "))
b = int(input("num2: "))

print("HCF of {0} and {1} is: {2}".format(a,b, computeHCF(a,b)))
```

```
num1: 16
num2: 8
HCF of 16 and 8 is: 8
```