

Function is divided into two parts

1. Function definition
2. Function calling or invoking

Function definition is nothing but writing function. Defining function is not invoking.

Function definition is developing a function.

After defining function it has to be called or invoked as executable statement.

Example of function without arguments

defining function

```
def sayHello():  
    "this is my first function within python"  
    print("Hello User,Welcome to Functions")
```

```
def sayBye():  
    print("Bye User")
```

function invoking or main

```
sayHello() # function invoking or function calling  
sayHello()  
sayHello()  
sayBye()
```

Output:

```
Hello User,Welcome to Functions  
Hello User,Welcome to Functions  
Hello User,Welcome to Functions  
Bye User
```

Example:

```
def drawLine():  
    for i in range(30):  
        print("*",end="")  
    print()
```

```
#main  
drawLine()
```

```
print("PYTHON")
drawLine()
print("PROGRAMMING")
drawLine()
print("LANGUAGE")
drawLine()
```

Output:

```
*****
PYTHON
*****
PROGRAMMING
*****
LANGUAGE
*****
```

Memory for function is allocated when function is invoked or called and memory is de-allocated after execution of function. Whenever function is called or invoked, the execution control switched from calling place to called function and after execution of function return to calling place.

Function without arguments do not receive any values.

Example:

```
def printPattern():
    for i in range(0,5):
        for j in range(0,i+1):
            print("*",end="")
        print()
```

```
printPattern()
```

Output:

```
*
**
***
****
*****
```

Local variables

What is local variable?

A variable created inside function is called local variable. The scope of this variable within function and lifetime of this variable until execution of function.

Example:

```
def fun1():  
    x=100 # local variable  
    y=200 #local variable  
    print(x,y,x+y)
```

fun1()

Output:

100 200 300

Example:

```
def f1():  
    x=100  
    print(x)
```

```
def f2():  
    y=200  
    print(y)  
    f1()
```

```
def f3():  
    z=300  
    f2();  
    print(z)
```

f3()

Output:

```
===== RESTART: F:/python6pmaug/funtest3.py =====  
200
```

100
300

Example:

```
def fun1():  
    x=100  
    print(x)
```

```
def fun2():  
    fun1()  
    print(x)
```

Output:

100

Traceback (most recent call last):

File "F:/python6pmaug/funtest4.py", line 11, in <module>
 fun2()

File "F:/python6pmaug/funtest4.py", line 7, in fun2
 print(x)

NameError: name 'x' is not defined

fun2()

Global variables

Function with arguments

Function with arguments receives values at the time of invoking or calling the function (OR) if function required input to perform operations we define that function with arguments.

Python allows to define function with 4 types of arguments/parameters

1. Required positional arguments
2. Default arguments

3. Variable length arguments
4. Keyword arguments