**FUNCITONS IN PYTHON**

**FUNCTION :** A group of statements which performs a particular task.

* Inbuilt Functions: Egs: input() / print()
* User Defined Functions:

**Advantages:**

* Reusability – Reuse a code at n no of times.
* Modularity – Huge application, entire code in one function, instead make a modularity by creating multiple functions.
* Maintenance – Easy Maintenance.

**Example 1:**

**Code:**

def **average**(a,b):

print(*"Average of 2 numbers :"*, (a+b)/2)

valA = int(input(*"Enter the value of A : "*))

valB = int(input(*"Enter the value of B : "*))

average(valA,valB)

**Output:**

Enter the value of A : 25

Enter the value of B : 35

Average of 2 numbers : 30.0

**Example 2 (Using RETURN)**

**Code:**

def **average**(a,b):

return ((a+b)/2)

valA = int(input(*"Enter the value of A : "*))

valB = int(input(*"Enter the value of B : "*))

result = average(valA,valB)

print(*"The Average of A, B is : "*, result)

**Output:**

Enter the value of A : 12

Enter the value of B : 13

The Average of A, B is : 12.5

**Example 3: Multiple Returns**

**Code:**

def **calculation**(a,b):

calAdd=a+b

calSub=a-b

calMul=a\*b

calDiv=a/b

return calAdd,calSub,calMul,calDiv # value will be returned as Tuple

valA = int(input(*"Enter the value of A : "*))

valB = int(input(*"Enter the value of B : "*))

result = calculation(valA, valB)

print(result)

**Output:**

Enter the value of A : 25

Enter the value of B : 12

(37, 13, 300, 2.0833333333333335)

**Example 4: LOCAL AND GLOBAL VARIABLE**

**Code:**

x=125 #GLOBAL VARIABLE DECLARED OUTSIDE

def **display**():

y =150 # LOCAL VARIABLE DECLARED INSIDE

print(*"Value of Y :"*,y)

print(*"The Value of X, "*,x)

display()

**Output:**

The Value of X, 125

Value of Y : 150

**Example 5: LOCAL AND GLOBAL VARIABLE (SAME NAME)**

**Code:**

x=125 #GLOBAL VARIABLE DECLARED OUTSIDE

def **display**():

x =150 # LOCAL VARIABLE DECLARED INSIDE

print(*"Value of X :"*,x)

print(*"The Value of X, "*,x)

display()

**Output:**

The Value of X, 125

Value of X : 150

**Example 6: CALLING GLOBAL VARIABLE AND LOCAL VARIABLE WITH SAME NAME, WITHIN A FUNCITON**

**Code:**

#GLOBAL VARIABLE DECLARED OUTSIDE

x=125

def **display**():

# LOCAL VARIABLE DECLARED INSIDE

x =150

print(*"Value of X :"*,x)

# RETRIEVE THE GLOBAL VARIABLE INSIDE A FUNCTION

print(globals()[*'x'*])

print(*"The Value of X, "*,x)

display()

**Output:**

The Value of X, 125

Value of X : 150

125

**Example 7: ASSIGN A FUNCTION TO A VARIABLE**

**Code:**

#GLOBAL VARIABLE DECLARED OUTSIDE

x=125

def **display**():

# LOCAL VARIABLE DECLARED INSIDE

x =150

print(*"Value of X :"*,x)

# RETRIEVE THE GLOBAL VARIABLE INSIDE A FUNCTION

print(globals()[*'x'*])

print(*"The Value of X, "*,x)

v1 = display

v1()

print(*"=========="*)

v1()

print(*"=========="*)

**Output:**

The Value of X, 125

Value of X : 150

125

==========

Value of X : 150

125

==========

**Example 8: FUNCTION INSIDE ANOTHER FUNCTION**

**Code:**

def **display**(name):

def **message**():

return *"Jazlyn and "*

result=message()+name

return result

print(display(*"Jerrick"*))

**Output:**

Jazlyn and Jerrick

**Example 9: FUNCTION PASSED AS PARAMETER TO ANOTHER FUNCTION**

**Code:**

def **display**(anotherfunction):

return *"Jerrick and "* + anotherfunction

def **name**():

return *"Jazlyn"*

print(display(name()))

**Output:**

Jerrick and Jazlyn

**Example 10: RETURN FROM WITHIN ANOTHER FUNCTION**

**Code:**

def **display**():

def **message**():

return *"Augusta"*

return message

fun=display()

print(fun())

**Output:**

Augusta

**Example 11: PASSING LIST INTO A FUNCTION**

**Code:**

def **listFunction**(lst):

for i in lst:

print(i)

dataList=[101,*"Kumar"*,*"Ranjan"*,*"05-09-1976"*,48,*"Dublin"*]

listFunction(dataList)

**Output:**

101

Kumar

Ranjan

05-09-1976

48

Dublin

**RECURSION :** Recursion is the process of a function calling itself. Eg: Finding the factorial of a given number.

**Example 12: FACTORIAL USING RECURSION**

**Code:**

def **factorial**(n):

if n == 0:

result = 1

else:

result=n\*factorial(n-1)

return result

valA = int(input(*"Enter the value for factorial : "*))

facValue = factorial(valA)

print(*"The factorial of is : "*, facValue)

**Output:**

Enter the value for factorial : 6

The factorial of is : 720

Enter the value for factorial : 0

The factorial of is : 1

**Example 13: KEY WORD ARGUMENTS**

**Code:**

def **average**(a,b):

return ((a+b)/2)

#USING ARGUMENT

def **argAverage**(a,b):

print(a)

print(b)

return ((a+b)/2)

a = int(input(*"Enter the value of A : "*))

b = int(input(*"Enter the value of B : "*))

result = average(a,b)

print(*"The Average of A, B is : "*, result)

#USING ARGUMENT

result1 = argAverage(b=25,a=10)

print(*"The Average of A, B is : "*, result1)

**Output:**

Enter the value of A : 35

Enter the value of B : 45

The Average of A, B is : 40.0

10

25

The Average of A, B is : 17.5

**Example 14: DEFAULT ARGUMENTS**

**Code:**

def **average**(a,b):

return ((a+b)/2)

#USING ARGUMENT

def **argAverage**(a,b):

print(a)

print(b)

return ((a+b)/2)

def **defAverage**(a=175,b=135):

return ((a+b)/2)

a = int(input(*"Enter the value of A : "*))

b = int(input(*"Enter the value of B : "*))

result = average(a,b)

print(*"The Average of A, B is : "*, result)

#USING ARGUMENT

result1 = argAverage(b=25,a=10)

print(*"The Key Word Argugment passed average of A, B is : "*, result1)

#USING DEFAULT ARGUMENT

result2 = defAverage()# NO ARGUMENT PASSED

print(*"The default argument value average of A, B is : "*, result2)

result3 = defAverage(a=125)# ARGUMENT PASSED

print(*"The default argument value and parameter value -- average of A, B is : "*, result3)

**Output:**

Enter the value of A : 10

Enter the value of B : 20

The Average of A, B is : 15.0

10

25

The Key Word Argugment passed average of A, B is : 17.5

The default argument value average of A, B is : 155.0

The default argument value and parameter value -- average of A, B is : 130.0

**Example 15: BMI EXAMPLE USING FUNCTIONS**

**Code:**

# BMI CALCULATION USING FUNCTIONS

def **calculateBMI**(height,weight):

heightinMeters=height\*0.4536

bmi = weight/(heightinMeters+heightinMeters)

return bmi

print(*"BMI USING FUNCTIONS OF PERSON 1 : "*,calculateBMI(6.1, 85))

print(*"BMI USING FUNCTIONS OF PERSON 2 : "*,calculateBMI(5.1, 75))

print(*"BMI USING FUNCTIONS OF PERSON 3 : "*,calculateBMI(5.8, 70))

print(*"BMI USING FUNCTIONS OF PERSON 4 : "*,calculateBMI(5.5, 65))

print(*"BMI USING FUNCTIONS OF PERSON 5 : "*,calculateBMI(5.1, 55))

print(*"BMI USING FUNCTIONS OF PERSON 6 : "*,calculateBMI(7.1, 105))

print(*"=================================="*)

**Output:**

==================================

BMI USING FUNCTIONS OF PERSON 1 : 15.35981727238558

BMI USING FUNCTIONS OF PERSON 2 : 16.210187778815232

BMI USING FUNCTIONS OF PERSON 3 : 13.303533418475947

BMI USING FUNCTIONS OF PERSON 4 : 13.027096360429693

BMI USING FUNCTIONS OF PERSON 5 : 11.887471037797836

BMI USING FUNCTIONS OF PERSON 6 : 16.30151278038602

==================================

**Example 16: USING ARGS AND KWARGS (WHICH IS OPTIONAL) IN FUNCTIONS**

**Code:**

def myArgsFunction(x, \*args, \*\*kwargs):  
 print(x)  
 print(args)  
 print(kwargs)  
  
print("======Only x value is provided========")  
myArgsFunction(10)  
print("======Only x value and args value (which is optional) is provided========")  
myArgsFunction(125, 32,45,75)  
print("======Only x value and args value and keyword argynebts (which is optional) is provided========")  
myArgsFunction(1025, 75,45,125, id=101, fname="Kumar", lname="Ranjan", dob="05-09-1976", age=48, place="Dublin")

**Output:**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\optionalArgsDemo.py

======Only x value is provided========

10

()

{}

======Only x value and args value (which is optional) is provided========

125

(32, 45, 75)

{}

======Only x value and args value and keyword argynebts (which is optional) is provided========

1025

(75, 45, 125)

{'id': 101, 'fname': 'Kumar', 'lname': 'Ranjan', 'dob': '05-09-1976', 'age': 48, 'place': 'Dublin'}

Process finished with exit code 0

**Example 16 a: USING ARGS AND KWARGS**

**Code:**

def myArgsFunction(x, \*args, \*\*kwargs):  
 print(x)  
 for each\_args in args:  
 print(each\_args)  
   
 for key, value in kwargs.items():  
 print(key, value)  
  
print("======Only x value is provided========")  
myArgsFunction(10)  
print("======Only x value and args value (which is optional) is provided========")  
myArgsFunction(125, 32,45,75)  
print("======Only x value and args value and keyword argynebts (which is optional) is provided========")  
myArgsFunction(1025, 75,45,125, id=101, fname="Kumar", lname="Ranjan", dob="05-09-1976", age=48, place="Dublin")

**Output:**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\optionalArgsDemo.py

======Only x value is provided========

10

======Only x value and args value (which is optional) is provided========

125

32

45

75

======Only x value and args value and keyword argynebts (which is optional) is provided========

1025

75

45

125

id 101

fname Kumar

lname Ranjan

dob 05-09-1976

age 48

place Dublin

Process finished with exit code 0

**Example 16 b: USING ARGS AND KWARGS**

**Code:**

ArgsFunction(x, \*args, \*\*kwargs):  
 print(x)  
 for each\_args in args:  
 print(each\_args)  
  
 for key, value in kwargs.items():  
 print(key, value)  
  
def myArgsFunction1(x, \*pos\_param, \*\*key\_param):  
 print(x)  
 for each\_args in pos\_param:  
 print(each\_args)  
  
 for key, value in key\_param.items():  
 print(key, value)  
 print("Calling funciton - myArgsFunction2===========")  
 myArgsFunction2(\*pos\_param, \*\*key\_param)  
 print("Modifing the myArgsFunction2 by adding new value 150 =============")  
 modified\_pos\_param = pos\_param+(150,) # adding values as tuple  
 myArgsFunction2(\*modified\_pos\_param, \*\*key\_param)  
def myArgsFunction2(\*args,\*\*kwargs):  
 print(args)  
 print(kwargs)  
  
  
# \*args AND \*kwargs  
  
print("======Only x value is provided========")  
myArgsFunction(10)  
print("======Only x value and args value (which is optional) is provided========")  
myArgsFunction(125, 32,45,75)  
print("======Only x value and args value and keyword keyparam (which is optional) is provided========")  
myArgsFunction(1025, 75,45,125, id=101, fname="Kumar", lname="Ranjan", dob="05-09-1976", age=48, place="Dublin")  
  
print("=========POS\_PARAM AND KEY\_PARAM ARGUMENTS OUTPUT============")  
myArgsFunction1(1025, 75,45,125, id=101, fname="Kumar", lname="Ranjan", dob="05-09-1976", age=48, place="Dublin")  
  
print("=========POS\_PARAM AND KEY\_PARAM ARGUMENTS OUTPUT============")  
myArgsFunction1(1025, 75,45,125, id=101, fname="Kumar", lname="Ranjan", dob="05-09-1976", age=48, place="Dublin")

**Output:**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\optionalArgsDemo.py

======Only x value is provided========

10

======Only x value and args value (which is optional) is provided========

125

32

45

75

======Only x value and args value and keyword keyparam (which is optional) is provided========

1025

75

45

125

id 101

fname Kumar

lname Ranjan

dob 05-09-1976

age 48

place Dublin

=========POS\_PARAM AND KEY\_PARAM ARGUMENTS OUTPUT============

1025

75

45

125

id 101

fname Kumar

lname Ranjan

dob 05-09-1976

age 48

place Dublin

Calling funciton - myArgsFunction2===========

(75, 45, 125)

{'id': 101, 'fname': 'Kumar', 'lname': 'Ranjan', 'dob': '05-09-1976', 'age': 48, 'place': 'Dublin'}

Modifing the myArgsFunction2 by adding new value 150 =============

(75, 45, 125, 150)

{'id': 101, 'fname': 'Kumar', 'lname': 'Ranjan', 'dob': '05-09-1976', 'age': 48, 'place': 'Dublin'}

=========POS\_PARAM AND KEY\_PARAM ARGUMENTS OUTPUT============

1025

75

45

125

id 101

fname Kumar

lname Ranjan

dob 05-09-1976

age 48

place Dublin

Calling funciton - myArgsFunction2===========

(75, 45, 125)

{'id': 101, 'fname': 'Kumar', 'lname': 'Ranjan', 'dob': '05-09-1976', 'age': 48, 'place': 'Dublin'}

Modifing the myArgsFunction2 by adding new value 150 =============

(75, 45, 125, 150)

{'id': 101, 'fname': 'Kumar', 'lname': 'Ranjan', 'dob': '05-09-1976', 'age': 48, 'place': 'Dublin'}

**QUIZ**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**