

WHY FUNCTIONS?

Repetitions (Birthday song)

```
print("Happy birthday to you!")
print("Happy birthday to you!")
print("Happy birthday to Dear name")
print("Happy birthday to you!")
print("Happy birthday to you!")
print("Happy birthday to you!")
print("Happy birthday to you!")
print("Happy birthday to Dear name")
print("Happy birthday to you!")
print("Happy birthday to you!")
```

function with argument (custom birthday wishes)

```
def wishBirthday(name): #passing argument/parameter
    " " "This function is used to wish birthday to someone" " "
    happy()
    happy()
    print(" Happy birthday to you dear", name, '!')
    happy()
    wishBirthday("Sourabh")
    wishBirthday("Jack")
```

Function required argument

```
1 def shoume(str): #required argument
2   print(str)
3   return
4   shoume("Sourabh")
5   shoume()

Sourabh
-----
TypeError: Traceback (most recent call last)
```

default arguments in PYTHON

```
1 def shoume(name,contact,age="18"): #default argument
2   print(name,contact,age)
3   shoume("Ram","343434333")
4   shoume("Shyam","343434333","33")
5   shoume("Willi","343434333",age="33")

Ram 343434333 18
Shyam 343434333 33
Willi 343434333 33
```

map function in PYTHON

```
2 def add(x,y): #mapping
3   return x+y
4 lst=[10,20,30,40,50]
5 lst2=[10,20,30,40,50]
6 print(lst)
7 print(*map(add,lst,lst2))

10, 20, 30, 40, 50
0 40 60 80 100
```

Lets define function

```
def happy():
    print("Happy birthday to you!")
    :#definition of function
    happy() #calling of function
    happy()
    print("Happy birthday to you dear name")
    happy()
    happy()
    happy()
    happy()
    print("Happy birthday to you dear name")
    happy()
    happy()
```

help/ documentation of function

using . operator wishBirthday.__doc__

using help function help(wishBirthday)

 help(len)

keyword arguments in PYTHON

```
1 def shoume(str,str2):
2   print(str)
3   print(str2)
4   return
5   #shoume("Sourabh")
6   shoume(str="Ram",str2="Shyam") #keyword argument

Ram
Shyam
```

LAMBDA Function (no name)

```
1 #lambda function
2
3 def sum(a,b):
4   return a+b
5   sum(10,20)

30

1 t=lambda a,b:a+b
2 print(t(10,30))

40
```

Multiple functions

```
def wishSourabh():
    happy()
    happy()
    print(" Happy birthday to dear Sourabh")
    happy()
def wishJack():
    happy()
    happy()
    print(" Happy birthday to dear Jack")
    happy()
    wishSourabh()
    wishJack()
```

Function can return multiple parameters in PYTHON

```
1 def getSumAndDiff(a,b): #multiple return values
2   return a+b,a-b
3   print(getSumAndDiff(20,10))

(30, 10)
```

Variable Arguments in PYTHON

```
1 def sum(*nos): #variable length argument
2   sum=0
3   for n in nos:
4     sum+=n
5   return sum
6   sum(10,20,30,40)
7   sum(10,20,30,40,50,60,70,80,90,100,110,120)

454
```

Global in functions

```
1 s1=0 #global variable
2 def sum(a,b,c):
3   s1=a+b+c
4   return s1
5   print(sum(10,20,30)) #60
6   print(s1) #60
7   #output will remain 0 inner s1 has function level scope

1 s1=0 #global variable
2 def sum(a,b,c):
3   global s1
4   s1=a+b+c
5   return s1
6   print(sum(10,20,30)) #60
7   print(s1)

60
60
```



By sahusourabh

Published 20th July, 2022.

Last updated 20th July, 2022.

Page 1 of 1.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>