

# Functions in Python

## Functions are divided into 2 parts

1. inbuilt functions ----> Ex: print(), max(), min(), id(), len(), mean(), describe()

2. user defined functions ---> we create our own functions using def keyword

## Syntax to write function code:

define the function

write the function logic

call the function

```
In [1]: def greet():  
        print('hello')  
        print('good morning')
```

```
In [2]: def greet():  
        print('hello')  
        print('good morning')  
        greet()
```

hello  
good morning

```
In [3]: def greet():  
        print('hello')  
        print('good morning')  
        greet()  
        print()  
        def greet():  
            print('hello')  
            print('good morning')  
        greet()
```

hello  
good morning

hello  
good morning

```
In [4]: def greet():  
        print('hello')  
        print('good morning')  
        greet()  
        print()  
        def greet():
```

```

    print('hello')
    print('good morning')
greet()
print()
def greet():
    print('hello')
    print('good morning')
greet()

```

hello  
good morning

hello  
good morning

hello  
good morning

```

In [5]: def greet(): # declare function
        print('hello')
        print('good morning')
        greet() # function calling without argument
        print()
        greet()

```

hello  
good morning

hello  
good morning

## function without argument

```

In [6]: def greet(): # declare function
        print('hello')
        print('good morning')
        greet() # function calling without argument

```

hello  
good morning

## function with argument

```

In [7]: def add(x,y): # here x and y called formal argument
        c=x+y
        print(c)
        add(5,6) # here 5 and 6 are called actual argument

```

11

```

In [8]: def add(x,y):
        c=x+y
        return c
        add(5,6)

```

Out[8]: 11

```
In [9]: def add(x,y):  
        c=x+y  
        return c  
        add(5)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[9], line 4  
      2     c=x+y  
      3     return c  
----> 4 add(5)  
  
TypeError: add() missing 1 required positional argument: 'y'
```

```
In [10]: def add(x,y):  
         c=x+y  
         return c  
         add(5,6,7)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[10], line 4  
      2     c=x+y  
      3     return c  
----> 4 add(5,6,7)  
  
TypeError: add() takes 2 positional arguments but 3 were given
```

```
In [12]: def add(x,y,z):  
         c=x+y  
         return c  
         add(5,6,7)
```

Out[12]: 11

```
In [13]: def add(x,y,z):  
         c=x+y+z  
         return c  
         add(5,6,7)
```

Out[13]: 18

```
In [14]: def add(x,y,z,n):  
         c=x+y+z+n  
         return c  
         add(5,6,7,8)
```

Out[14]: 26

```
In [15]: def greet():  
         print('hello')  
         print('good morning')  
         greet()
```

```
def add(x,y):  
    c=x+y  
    return c  
add(5,6)
```

```
hello  
good morning
```

Out[15]: 11

```
In [16]: def greet():  
          print('hello')  
          print('good morning')  
          def add(x,y):  
              c=x+y  
              return c  
          greet()  
          add(5,6)
```

```
hello  
good morning
```

Out[16]: 11

```
In [19]: def greet():  
          print('hello')  
          print('good morning')  
          def add(x,y):  
              c=x+y  
              return c  
          def sub(x,y):  
              d=x-y  
              return d  
          greet()  
          add(5,6)  
          sub(5,6)
```

```
hello  
good morning
```

Out[19]: -1

```
In [20]: def greet():  
          print('hello')  
          print('good morning')  
          def add(x,y):  
              c=x+y  
              return c  
          def sub(x,y):  
              d=x-y  
              return d  
          greet()  
          print(add(5,6))  
          print(sub(5,6))
```

```
hello
good morning
11
-1
```

```
In [21]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d
          add_sub(4,5)
```

```
Out[21]: (9, -1)
```

```
In [22]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d
          result=add_sub(4,5)
          print(result)
          print(type(result))
```

```
(9, -1)
<class 'tuple'>
```

```
In [23]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d
          result1, result2=add_sub(4,5)
          print(result1)
          print(result2)
          print(type(result1))
```

```
9
-1
<class 'int'>
```

```
In [25]: def add_sub_mul(x,y):
          c=x+y
          d=x-y
          e=x*y
          return c,d, e
          result1, result2, result3=add_sub_mul(4,5)
          print('Addition of Two Numbers:',result1)
          print('Subtraction of Two Numbers:',result2)
          print('Multiplication of Two Numbers:',result3)
          print(type(result1))
```

```
Addition of Two Numbers: 9
Subtraction of Two Numbers: -1
Multiplication of Two Numbers: 20
<class 'int'>
```

## update

```
In [26]: def update():  
         x=8  
         print(x)  
         update()
```

8

```
In [27]: def update(x):  
         x=8  
         return x  
         update(10)
```

Out[27]: 8

```
In [29]: def update(x):  
         x=8  
         print(x)  
         a=10  
         update(a)  
         print(a)
```

8

10