

# Functions

```
In [1]: def greet():    # We just define the function
        print('Good Evening')    # print the statement
```

```
In [2]: def greet():    # We just define the function
        print('Good Evening')    # print the statement
        greet()           # calling function
```

Good Evening

```
In [3]: # to print 3 times
def greet():    # We just define the function
    print('Good Evening')    # print the statement
    greet()           # calling function

def greet():    # We just define the function
    print('Good Evening')    # print the statement
    greet()           # calling function

def greet():    # We just define the function
    print('Good Evening')    # print the statement
    greet()           # calling function
```

Good Evening  
Good Evening  
Good Evening

```
In [4]: def greet():    # We just define the function
        print('Good Evening')    # print the statement
        greet()           # calling function
        print()
        greet()
        print()
        greet()
        print()
        greet()
```

Good Evening  
  
Good Evening  
  
Good Evening  
  
Good Evening

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

11

```
In [6]: def add(x):  
        x=x+y  
        print(x)  
  
        add(5,6)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[6], line 5  
      2     x=x+y  
      3     print(x)  
----> 5 add(5,6)  
  
TypeError: add() takes 1 positional argument but 2 were given
```

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

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[7], line 4  
      2     c=x+y  
      3     print(c)  
----> 4 add(5,6)  
  
TypeError: add() missing 1 required positional argument: 'z'
```

```
In [8]: def add(x,y):  
        c=x+y  
        print(c)  
        add(5,6,7)
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[8], line 4  
      2     c=x+y  
      3     print(c)  
----> 4 add(5,6,7)  
  
TypeError: add() takes 2 positional arguments but 3 were given
```

```
In [9]: def greet():  
        print("Good Evening")  
        greet()  
        print()  
        def add(x,y):  
            c=x+y  
            print(c)  
  
        add(5,6)
```

Good Evening

```
In [11]: # Standard way to write

def greet():
    print("Good Evening")
def add(x,y):
    c=x+y
    print(c)

greet()
print()
add(5,6)
```

Good Evening

11

```
In [12]: def greet():
    print("Good Evening")
def add(x,y):
    c=x+y
    print(c)
def sub(x,y):
    c=x-y
    print(c)
greet()
print()
add(5,6)
sub(5,6)
```

Good Evening

11

-1

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

Out[13]: 11

```
In [14]: def add(x,y):
    c=x+y
    return c
def sub(x,y):
    d=x-y
    return d
add(20,10)
sub(20,10)
```

Out[14]: 10

```
In [15]: def add(x,y):
    c=x+y
    return c
def sub(x,y):
```

```

    d=x-y
    return d
print(add(20,10))
print(sub(20,10))

```

30

10

```

In [17]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d
          print(add_sub(20,10))
          result=add_sub(20,10)
          print(type(result))

```

(30, 10)

<class 'tuple'>

```

In [19]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d

          result1, result2=add_sub(20,10)

          print(type(result1))
          print(type(result2))
          print(result1)
          print(result2)

```

<class 'int'>

<class 'int'>

30

10

```

In [20]: def add_sub(x,y):
          c=x+y
          d=x-y
          return c,d

          result = add_sub(10,20)
          result1=add_sub(10,20)
          print(result)
          print(result1)
          print(type(result))
          print(type(result1))

```

(30, -10)

(30, -10)

<class 'tuple'>

<class 'tuple'>

## Functions has main two concepts: 1. Without arg 2. With arg

- This is define in 2 parts
- 1 Formal arg
- 2. Actual arg
- This is divide in 4 parts
- Possitional arg
- Keyword
- default
- Variable

```
In [21]: def update(x):
          x=8
          return x
          update(10)
```

Out[21]: 8

```
In [22]: def update(x):
          x=8
          return x
          a=10
          print(update(a))
          print(a)
```

8  
10

```
In [23]: def add(x,y):      # x,y are formal arguments
          c=x+y
          return c
          add(4,5)      # 4,5 are Actual arguments
```

Out[23]: 9

## Positional Arguments

```
In [24]: def add(x,y):      # x,y are Formal arguments
          c=x+y
          return c
          add(4,5)      # 4,5 are Actual arguments
```

Out[24]: 9

```
In [25]: # Positional arguments

def add(x,y):      # x,y are Formal arguments
    c=x+y
    return c
add(4)
```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[25], line 6
      4     c=x+y
      5     return c
----> 6 add(4)

TypeError: add() missing 1 required positional argument: 'y'

```

```

In [26]: def add(x):    # x,y are Formal arguments
          c=x+y
          return c
          add(4,5)

```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[26], line 4
      2     c=x+y
      3     return c
----> 4 add(4,5)

TypeError: add() takes 1 positional argument but 2 were given

```

```

In [28]: def person(name,age):
          print(name)
          print(age)

          person('vihari',3)

```

```

vihari
3

```

```

In [29]: def person(name,age):
          print(name)
          print(age)

          person(3,'vihari')

```

```

3
vihari

```

```

In [30]: def person(name,age):
          print(name)
          print(age+1)

          person(3,'vihari')

```

```

3

```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[30], line 5
      2     print(name)
      3     print(age+1)
----> 5     person(3, 'vihari')

Cell In[30], line 3, in person(name, age)
      1 def person(name, age):
      2     print(name)
----> 3     print(age+1)

TypeError: can only concatenate str (not "int") to str

```

## Keyword Argument

```

In [31]: def person(name, age):
          print(name)
          print(age+1)

          person(age=3, name='vihari')

```

vihari  
4

```

In [32]: def person(name, age, salary):
          print(name)
          print(age+1)

          person(age=3, name='vihari')

```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[32], line 5
      2     print(name)
      3     print(age+1)
----> 5     person(age=3, name='vihari')

TypeError: person() missing 1 required positional argument: 'salary'

```

## Default Argument

```

In [33]: def person(name, age, age2):
          print(name)
          print(age)
          print(age2)
          person(age=20, name='nit', age2=21)

          # this is called as Keyword arguments

```

```
nit
20
21
```

```
In [34]: def person(name,age):
          print(name)
          print(age)

          person('nit')
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[34], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

TypeError: person() missing 1 required positional argument: 'age'
```

```
In [35]: def person(name,age=18):
          print(name)
          print(age)

          person('nit')
```

```
nit
18
```