

```

def greet():    #function name greet
    print('hello')
    print('gd mrng')

def greet():    #function name greet
    print('hello')
    print('gd mrng')
greet()

hello
gd mrng

def greet():
    print('hello')
    print('gd mrng')
greet()
def greet():
    print('hello')
    print('gd mrng')
greet()

hello
gd mrng
hello
gd mrng

def greet():
    print('hello')
    print('gd mrng')
greet()
print()
def greet():
    print('hello')
    print('gd mrng')
greet()

hello
gd mrng

hello
gd mrng

#function without argument
def greet():    # declare function without argument
    print('hello')
    print('gd mrng')
greet()
print('*****')
greet()    #function calling without argument
print('*****')
greet()

```

```

hello
gd mrng
*****
hello
gd mrng
*****
hello
gd mrng

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

hello
good morning team

##functions with argument
def add(x,y):      #x,x are formal argument
    c=x+y
    print(c)
add(5,6)           #5,6 are actual argument

11

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

```

```

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

```

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

```

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

11

def add(x,y):
    c=x+y

```

```
    return(c)
add(5,6,7)
```

```
-----
-----
TypeError                                Traceback (most recent call
last)
Cell In[3], line 4
      2     c=x+y
      3     return(c)
----> 4 add(5,6,7)
```

TypeError: add() takes 2 positional arguments but 3 were given

```
def greet():
    print('hello')
    print('gd mrng')
greet()
```

```
hello
gd mrng
```

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

```
11
```

```
def add(x,y,z):
    c=x+y+z+m
    return c
add(5,6,7)
```

```
-----
-----
NameError                                Traceback (most recent call
last)
Cell In[5], line 4
      2     c=x+y+z+m
      3     return c
----> 4 add(5,6,7)
```

```
Cell In[5], line 2, in add(x, y, z)
      1 def add(x,y,z):
----> 2     c=x+y+z+m
      3     return c
```

NameError: name 'm' is not defined

```
def add(x,y,z,n):  
    c=x+y+z+m  
    return c  
add(5,6,7,8)
```

-----

NameError Traceback (most recent call last)

```
Cell In[6], line 4  
      2     c=x+y+z+m  
      3     return c  
----> 4 add(5,6,7,8)
```

```
Cell In[6], line 2, in add(x, y, z, n)  
      1 def add(x,y,z,n):  
----> 2     c=x+y+z+m  
      3     return c
```

NameError: name 'm' is not defined

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

26

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

```
hello  
good morning team
```

11

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

```
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 team
11
-1

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'>

def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d
result, result1=add_sub(4,5)
print(result)
print(result1)
print(type(result))

9
-1
<class 'int'>

def add_sub_mul(x,y):
    c=x+y
    d=x-y
    e=x*y
    return c,d,e
add, sub, mul=add_sub_mul(4,5)

add
sub
mul

20

def add_sub_mul(x,y):
    c=x+y
    d=x-y
    e=x*y

```

```

    return c,d,e
add, sub, mul=add_sub_mul(4,5)

print(add)
print(sub)
print(mul)

9
-1
20

```

## update

```

def update():
    x=8
    print(x)
update()

```

8

```

def update():
    x=8
    print(x)
update(8)

```

```

-----
-----
TypeError                                Traceback (most recent call
last)
Cell In[18], line 4
      2     x=8
      3     print(x)
----> 4 update(8)

```

TypeError: update() takes 0 positional arguments but 1 was given

```

def update(x):
    x=8
    return x
update(100)

```

8

```

def update(x):
    x=8
    return x
a=15
update(a)
print(a)

```

15

