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

In [6]:
```
def greet():
    print('Hello')
    print('Good Morning')
greet()
```

```
Hello
Good Morning
```

In [8]:
```
def greet():
    print('Hello')
    print('Good Morning')
greet()

def greet():
    print('Hello')
    print('Good Morning')
greet()
```

```
Hello
Good Morning
Hello
Good Morning
```

In [9]:
```
def greet():
    print('Hello')
    print('Good Morning')
greet()

def greet():
    print('Hello')
    print('Good Morning')
greet()
```

```
Hello
Good Morning
Hello
Good Morning
```

In [10]:
```
def greet():
    print('Hello')
    print('Good Morning')
greet()

print()

def greet():
    print('Hello')
    print('Good Morning')
greet()
```

```
Hello
Good Morning

Hello
Good Morning
```

In [12]:
```
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 [14]:
```
def greet():                # declare function without arguement
    print('Hello')
    print('Good Morning')
greet()
print('**************')
greet()                     #function calling witout arguement
print('**************')
greet()
```

```
Hello
Good Morning
**************
Hello
Good Morning
**************
Hello
Good Morning
```

In [16]:
```
# function without arguement

def greet():
    print('Hello')
    print('Good Morning')
greet()
```

```
Hello
Good Morning
```

In [21]:
```
# Function with arguement
def add(x,y):
    c = x+ y
```

```
    print(c)
add(2,3)
```

```
5
```

In [22]:
```
# Function with arguement
def add(x,y):
    c = x+ y
    return(c)
add(2,3)
```

Out[22]:
```
5
```

In [23]:
```
def add(x,y):
    c = x+ y
    return c
add(5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[23], line 4
      2     c = x+ y
      3     return c
----> 4 add(5)

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

In [24]:
```
def add(x,y):
    c = x+ y
    return c
add(5,3,4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[24], line 4
      2     c = x+ y
      3     return c
----> 4 add(5,3,4)

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

In [25]:
```
def add(x,y,z):
    c = x+ y
    return c
add(5,3,4)
```

Out[25]:
```
8
```

In [26]:
```
def add(x,y,z):
    c = x+ y+z+m
    return c
add(5,3,4)
```

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

Cell In[26], 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
```

In [28]:
```
def add(x,y,z,n):
    c = x+ y+z+m
    return c
add(5,3,4,5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[28], line 4
      2     c = x+ y+z+m
      3     return c
----> 4 add(5,3,4,5)

Cell In[28], 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
```

In [30]:
```
# function with arguement

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

Out[30]: 26

In [32]:
```
def greet():
    print('hello')
    print('good morning')
greet()

def add(x,y):
    c = x+y
    return c
add(4,5)
```

```
hello
good morning
```

Out[32]: 9

In [34]:
```
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 [35]:
```
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 [37]:
```
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'>
```

In [40]:
```
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
```

Out[40]: 20

In [42]:
```
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

In [43]:
```
def update():
    x = 8
    print(x)
update()
```

```
8
```

In [44]:
```
def update():
    x = 8
    print(x)
update(8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 4
      2     x = 8
      3     print(x)
----> 4 update(8)

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

In [45]:
```
def update(x):
    x = 8
    return x

update(100)
```

Out[45]: 8

In [48]:
```
def update(x):
    x = 8
    return x

a = 15
update(a)
print(a)
```

```
15
```

In [49]:
```
def fun():
    print("Welcome to India")
```

In [50]:
```
def fun():
    print("Welcome to India")
```

```
fun()
```

```
Welcome to India
```

In [55]:
```
def evenodd(x: int)  :
    if(x%2 == 0):
        return "Even"
    else:
        return "Odd"

print(evenodd(13))
print(evenodd(4))
```

```
Odd
Even
```

In [56]:
```
def evenodd(x)  :
    if(x%2 == 0):
        return "Even"
    else:
        return "Odd"

print(evenodd(18))
print(evenodd(3))
```

```
Even
Odd
```