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

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

```
hello
good morning
```

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

```
hello
good morning
hello
good morning
```

In [7]:
```
def greet():
    print('hello')
    print('good morning')
greet()
print()
greet()
```

```
hello
good morning

hello
good morning
```

In [5]:
```
def greet():
    print('hello')
    print('good morning')
greet()
print('******************')
greet()
```

```
hello
good morning
******************
hello
good morning
```

In [6]:
```
def greet():
    print('hello')
    print('good morning')
greet()
```

```
print('**************')
greet()
print('##################')
```

```
hello
good morning
**************
hello
good morning
##################
```

define the function without argument

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 4
      2     c=x+y
      3     print(c)
----> 4 add(5)

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

In [9]:
```
def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
9
```

In [11]:
```
def greet():
    print('hello')
    print('good morning')
greet()
def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
hello
good morning
9
```

In [13]:
```
def greet():
    print('hello')
    print('good morning')
def add(x,y):
    c=x+y
    print(c)
greet()
add(5,4)
```

```
hello
good morning
9
```

In [15]:
```
def greet():
    print('hello')
    print('good morning')
def add(x,y):
    c=x+y
    print(c)
greet()
print('-------')
add(5,4)
```

```
hello
good morning
-------
9
```

In [17]:
```
def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
9
```

In [19]:
```
def add(x,y):
    c=x+y
    return c
add(5,4)
```

Out[19]: 9

In [21]:
```
def add(x,y,z):
    c=x-y-z
    return c
add(5,2,3)
```

Out[21]: 0

In [23]:
```
def greet():
    print('hello')
    print('good morning')
def add(x,y):
    c=x+y
    print(c)
def sub(x,y):
    d=x-y
    return d
greet()
add(5,4)
sub(5,4)
```

```
hello
good morning
9
```

Out[23]: 1

In [25]:
```
def greet():
    print('hello')
    print('good morning')
def add_sub(x,y):
    c=x+y
    d=x-y
    return c
    return d

greet()
add_sub(5,4)
```

```
hello
good morning
```

Out[25]: 9

In [27]:
```
def greet():
    print('hello')
    print('good morning')
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d
greet()
add_sub(4,7)
```

```
hello
good morning
```

Out[27]: (11, -3)

In [29]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d
results=add_sub(4,5)
print(results)
print(type(results))
```

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

In [31]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d
res1,res2=add_sub(4,5)
print(res1,res2)
print(type(res1))
print(type(res2))
```

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

19th march

In [34]:
```
def update():
    x=8
    print(x)
update(8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[34], line 4
      2     x=8
      3     print(x)
----> 4 update(8)

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

In [ ]:
```
def update(x):       #update function take the value from the user
    x = 8
    print(x)
update(10)
```

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

In [ ]:
```
def add(a,b):    # a & b called formal argument
    c=a+b
    print(c)
add(5,6)   #5 and 6 we called as actual argument
```

In [ ]:
```
def add(a,b):  # a & b called formal argument
    c=a+b
    print(c)
add(2,3,7)   #5 and 6 we called as actual argument
```

In [ ]:
```
def add(a,b):    # a & b called formal argument
    c=a+b
    print(c)
add(5)   #5 and 6 we called as actual argument
```

positional argument

In [ ]:
```
def positional(name,age):
    print(name)
    print(age)
positional('nit',23)
```

In [ ]:
```
def positional(name,age):
    print(name)
    print(age)
positional('nit')
```

In [36]:
```
def  position(name,age):
     print(name)
     print(age-5)
positional(22,'nit')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[36], line 4
      2      print(name)
      3      print(age-5)
----> 4 positional(22,'nit')

NameError: name 'positional' is not defined
```

In [37]:
```
def person(name, age):
    print(name)
    print(age-5)

person(22, 'nit')
```

```
22
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[37], line 5
      2     print(name)
      3     print(age-5)
----> 5 person(22, 'nit')

Cell In[37], line 3, in person(name, age)
      1 def person(name, age):
      2     print(name)
----> 3     print(age-5)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

keyword Argument

In [40]:
```
def positional(name,age):
    print(name)
    print(age-5)
positional(name='nit',age=23)
```

```
nit
18
```

In [42]:
```
def positional(name,age,mob):
    print(name)
```

```
    print(age-5)
positional(name='nit',age=23)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[42], line 4
      2     print(name)
      3     print(age-5)
----> 4 positional(name='nit',age=23)

TypeError: positional() missing 1 required positional argument: 'mob'
```

In [44]:
```
def positional(name,age,mob):
    print(name)
    print(age-5)
    print(mob)
positional(name='nit',age=23,mob=102)
```

```
nit
18
102
```

Defult argument

In [47]:
```
def positional(name,age=21):
    print(name)
    print(age)
positional('nit')
```

```
nit
21
```

In [49]:
```
def positional(name,age=21):
    print(name)
    print(age)
positional('nit',40)
```

```
nit
40
```

28th APR

variable length

In [53]:
```
def positional(name,age):
    print(name)
    print(age)
positional('nit',40)
```

```
nit
40
```

In [55]:
```
def positional(name,age):
    print(name)
    print(age)
positional('nit',40,30,20,60)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[55], line 4
      2     print(name)
      3     print(age)
----> 4 positional('nit',40,30,20,60)

TypeError: positional() takes 2 positional arguments but 5 were given
```

In [57]:
```
def positional(name,*age):
    print(name)
    print(age)
positional('nit',40,20,50,60)
```

```
nit
(40, 20, 50, 60)
```

In [59]:
```
def person(name,*age):
    print(type(name))
    print(type(age))
person('nit',40,20,36,60,50)
```

```
<class 'str'>
<class 'tuple'>
```

In [61]:
```
def sum(a,b):
    c=a+b
    print(c)
sum(10,7)
```

```
17
```

In [63]:
```
def sum(a,b):
    c=a+b
    print(c)
sum(10,7,4,9,10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[63], line 4
      2     c=a+b
      3     print(c)
----> 4 sum(10,7,4,9,10)

TypeError: sum() takes 2 positional arguments but 5 were given
```

In [65]:
```
def sum(a,*b):
    c=a+b
    print(c)
sum(10,7,6,8,9,10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[65], line 4
      2     c=a+b
      3     print(c)
----> 4 sum(10,7,6,8,9,10)

Cell In[65], line 2, in sum(a, *b)
      1 def sum(a,*b):
----> 2     c=a+b
      3     print(c)

TypeError: unsupported operand type(s) for +: 'int' and 'tuple'
```

In [67]:
```
def sum(a,*b):
    #c=a+b
    print(type(a))
    print(type(b))
sum(10,7,6,8,9,10)
```

```
<class 'int'>
<class 'tuple'>
```

In [69]:
```
def sum(a,*b):
    c=a+b
    print(a)
    print(b)
sum(10,7,6,8,9,10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[69], line 5
      3     print(a)
      4     print(b)
----> 5 sum(10,7,6,8,9,10)

Cell In[69], line 2, in sum(a, *b)
      1 def sum(a,*b):
----> 2     c=a+b
      3     print(a)
      4     print(b)

TypeError: unsupported operand type(s) for +: 'int' and 'tuple'
```

In [71]:
```
def sum(a,*b):
    c = a
    for i in b:
        c = c + i
    print(c)
sum(1,40,28,39,40,50)
```

```
198
```

In [73]:
```
def sum(a,*b):
    c = a
    for i in b:
```

```
        c = c + i
    print(c)
sum(20,56,97)
```

```
173
```

In [75]:
```
def person():
    person('harika',21,'chinnu',18)
```

In [77]:
```
def person(name,*data):
    print(name)
    print(data)
person('harika',21,'chinnu',18)
```

```
harika
(21, 'chinnu', 18)
```

In [79]:
```
def person(name,*data):
    print('name')
    print(data)
person('harika',age=21,'chinnu',age=18,mob=9640)
```

```
  Cell In[79], line 4
    person('harika',age=21,'chinnu',age=18,mob=9640)
                                                    ^
SyntaxError: positional argument follows keyword argument
```

In [81]:
```
def person(name,**data):
    print('name')
    print(data)
person('harika',age=21,mob=9640)
```

```
name
{'age': 21, 'mob': 9640}
```

In [83]:
```
def person(name,**data):
    print('name')
    print(data)

person('harika',age=21,marks=50 ,mob =987767)
# we got error as keyword argument thats why we add another *
```

```
name
{'age': 21, 'marks': 50, 'mob': 987767}
```

Global & local variable

In [86]:
```
a = 10
print(a)
```

```
10
```

In [ ]:
```
a = 10     # global var
def something():
    b = 17    # local var
```

```
    print('in function', b)
    print('out function', a)
```

In [73]:
```
a = 10     # global var
def something():
    b = 17    # local var
    print('in function', b)
print('out function', a)
```

```
out function 10
```

In [77]:
```
a = 10     # global var
def something():
    b = 17    # local var
    print('in function', b)
something()
print('out function', a)
```

```
in function 17
out function 10
```

In [81]:
```
a = 10
def something():
    b = 15
print(' in function',a)
print('out function',a)
```

```
 in function 10
out function 10
```

In [ ]: