In [1]:
```
def greet():# here we just define the function
    print('good eveining')# print the statement
```

In [2]:
```
def greet():# here we just define the function
    print('good eveining')# print the statement
greet()
```

```
good eveining
```

In [5]:
```
# to print 3 time
def greet():# here we just define the function
    print('good eveining')# print the statement
greet()
def greet():# here we just define the function
    print('good eveining')# print the statement
greet()
def greet():# here we just define the function
    print('good eveining')# print the statement
greet()
```

```
good eveining
good eveining
good eveining
```

In [7]:
```
def greet():# here we just define the function
    print('good eveining')# print the statement
greet()
print()
greet()
print()
greet()
print()
greet()
```

```
good eveining

good eveining

good eveining

good eveining
```

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

add(5,6)
```

```
11
```

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

add(5,6)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[9], line 5
      2     c=x+y
      3     print(c)
----> 5 add(5,6)

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

In [10]:
```
def add(x,y,z):
    c=x+y
    print(c)

add(5,6)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 5
      2     c=x+y
      3     print(c)
----> 5 add(5,6)

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

In [11]:
```
def add(x,y):
    c=x+y
    print(c)

add(5,6,7)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 5
      2     c=x+y
      3     print(c)
----> 5 add(5,6,7)

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

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

add(5,6)
```

```
good eveining

11
```

In [14]:
```
# standard way to write
def greet():
    print('good eveining')
```

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

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

```
good eveining

11
```

In [16]:
```
def greet():
    print('good eveining')
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 eveining

11
-1
```

In [17]:
```
def add(x,y):
    c=x+y
    return c
add(5,6)
```

Out[17]: 11

In [19]:
```
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[19]: 10

In [20]:
```
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 [23]:
```
def add_sub(x,y):
    c=x+y
    d=x-y
    return c,d
print(add_sub(20,10))
```

```
(30, 10)
```

In [24]:
```
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 [30]:
```
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 [51]:
```
def add_sun(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'>
```

function has main two concept

without arg

with arg

- this is define in 2 part
- 1 formal arg
-2. Actual arg
- this is devide in 4 part
-possitional arg
-keyword
-default
-variable

In [31]:
```
def update(x):
    x=8
    return x
update(10)
```

Out[31]: 8

In [32]:
```
def update(x):
    x=8
    return x
a=10

print(update(a))
print(a)
```

```
8
10
```

In [52]:
```
def add(x,y): # x& y is called formal argument
    c=x+y
    return c

add(4,5)# 4 and 5 is called Actual arguments
```

Out[52]: 9

positional argument

In [54]:
```
def add(x,y): # x& y is called formal argument
    c=x+y
    return c
```

```
add(4,5)# 4 and 5 is called Actual arguments
```

Out[54]: 9

In [53]:
```
#positional arg
def add(x,y): # x& y is called formal argument
    c=x+y
    return c

add(4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[53], line 6
      3     c=x+y
      4     return c
----> 6 add(4)

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

In [38]:
```
def add(x): # x& y is called formal argument
    c=x+y
    return c

add(4,5)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[38], line 5
      2     c=x+y
      3     return c
----> 5 add(4,5)

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

In [39]:
```
def person(name,age):
    print(name)
    print(age)

person('nit',22)
```

```
nit
22
```

In [40]:
```
def person(name,age):
    print(name)
    print(age)

person(22,'nit')
```

```
22
nit
```

In [55]:
```
def person(name,age):
    print(name)
    print(age+1)
```

```
person(22,'nit')
```

```
22
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[55], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(22,     )

Cell In[55], 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 arggument

In [56]:
```
def person(name,age):
    print(name)
    print(age+1)

person(age=22,name='nit')
```

```
nit
23
```

In [57]:
```
def person(name,age,salary):
    print(name)
    print(age+1)

person(age=22,name='nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[57], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(age=22,name=     )

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

In [58]:
```
#default Argument
```

In [59]:
```
def person(name, age, age2):
    print(name)
    print(age)
    print(age2)

person(age = 20,  name = 'nit', age2 = 21)

#this is called keyword arguments
```

```
nit
20
21
```

In [61]:
```
def person(name,age):
    print(name)
    print(age)

person('nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[61], line 5
      2     print(name)
      3     print(age)
----> 5 person(     )

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

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

person('nit')
```