Parameters and Arguments

-var which is defined inside fun headig is called formal parameters,used for storing values comming from fun call

-var which is defined inside fun body is called local parameters ,used for storing temp result

-var/values which is defined inside fun call is called argumnents ,all the values of argumets are passing to parameters .

THIS MECHANISH IS CALLED ARGUMENTS PASSING.

PYTHON HAS 5 TYPES OF PATAMETERS OR AGRUMENTS 1.positional arguments 2.default arguments 3.keyword arguments 4.variabe length arguments 5.keyword variable length arguments

1.POSITIONAL ARGUMENTS

THE NUMBER OF ARGUMENTS OF FUN CALL MUST BE EQUAL TO NUMBER OF PARAMETERS IN FUN HEADING.

In [31]:

```
#PROGRAM FOR POSITONAL ARGUMENTS
def studinfo(rollno,name,sub):
    print('{} \t{} \t{}'.format(rollno,name,sub))
studinfo(1,'alina','python')
studinfo(2,'naresh','python')
studinfo(3,'suresh','python')
```

```
1       alina   python
2       naresh  python
3       suresh  python
```

2.DEFAULT ARGUMENTS

WHEN THERE IS A COMMON VALUE FOR A FAMILY OF SIMILAT FUN CALLS ,THEN SUCH TYPE OF COMMON VALUE MUST BE TAKEN AS DEFAULT ATGUMENTS.

In [62]:

```
#PROGRAM FOR DEFAULT ARGUMENTS
def studinfo(rollno,name,marks,course='data science'):
    print('{} \t{} \t{} \t{}'.format(rollno,name,marks,course))
studinfo(1,'alina',90)
studinfo(2,'naresh',88)
studinfo(3,'suresh',77)
```

```
1       alina   90      data science
2       naresh  88      data science
3       suresh  77      data science
```

3.KEYWORD ARGUMENTS

IN SOME OF CIRCUMSTANCE,WE KNOW THE FUN NAME AND FORMAL PARAM NAMES AND WO DO NOT KNOW THE ORDER OF FORMAL PARAM NAMES ,TO OVERCOME THIS PROBLEM WE USE KEYWORD ARGUMENTS ALL THE FORMAL PARAM NAMES USED AS ARGUMENTS IN FUN CALL AS KEYS.

In [72]:

```
#program for keyword argumets
def values(a,b,c,d):
    print(a, b, c, d)
#main
values(a=20,c=40,d=20,b=50)
values(c=43,d=54,a=80,b=543)
values(a=87,c=503,d=20,b=54)
values(c=48,d=56,a=60,b=5)
```

```
20 50 40 20
80 543 43 54
87 54 503 20
60 5 48 56
```

In [76]:

```
#program for keyword argumets
def values(a,b,c,d):
    print('a={}\t b={}\t c={}\t d={}'.format(a,b,c,d))
#main
values(a=20,c=40,d=20,b=50)
values(c=43,d=54,a=80,b=543)
values(a=87,c=503,d=20,b=54)
values(c=48,d=56,a=60,b=5)
```

```
a=20     b=50     c=40     d=20
a=80     b=543    c=43     d=54
a=87     b=54     c=503    d=20
a=60     b=5      c=48     d=56
```

4.VARIABLE LENGTH ARGUMENTS

WHEN WE HAVE FAMILT OF MULTIPLEFUN CALLSEITH VARIABLE NUMBER OF VALUES THEN ,WE MUST DEFINE MULTIPLE FUN DEFINITIONS, THIS PROCESS LEADS TO MORE DEV TIME. VAR LENGTH ARG CONCEPTS,WE MUST DEFINE SINGLE FUN DEFINITON AND TAKES A FORMAL PARAM PROCEDED WITH A SUMBOL CALLED ASTRISK(* PARAM),ALSO CALLED KEY, WHOSE PURPOSE IS TO STORE ANY NUMBER VALUES COMMING FROM FUN CALLS AND WHOSE TYPE IS <class,'Tuple'>.

In [95]:

```
#program for without var length arg
def disp(a,b,c,d):
    print(a,b,c,d)
disp(10,20,30,40)

def disp(a,b,c):
    print(a,b,c)
disp(10,20,30)

def disp(a,b):
    print(a,b)
disp(10,20)

def disp(a):
    print(a)
disp(10)
```

```
10 20 30 40
10 20 30
10 20
10
```

In [101…

```
#program for var length argumets
def disp(*alina): #here alina is var length  arg whose type is 'tuple'.
    print(alina,type(alina),len(alina))
#main
disp(10,20,30,40,50)
disp(10,20,30,40)
disp(10,20)
disp(10)
```

```
(10, 20, 30, 40, 50) <class 'tuple'> 5
(10, 20, 30, 40) <class 'tuple'> 4
(10, 20) <class 'tuple'> 2
(10,) <class 'tuple'> 1
```

In [113…

```
#program for var length argumets
def disp(sno,name,*alina): #here alina is var length  arg whose type is 'tuple'.
    print( sno , name, alina,type(alina),len(alina))
#main
disp(1,'alina',10,20,30,40,50,)
disp(2,'naresh',10,20,30,40)
disp(3,'suresh',10,20)
disp(4,'rossum',10)
```

```
1 alina (10, 20, 30, 40, 50, 'jaya') <class 'tuple'> 6
2 naresh (10, 20, 30, 40) <class 'tuple'> 4
3 suresh (10, 20) <class 'tuple'> 2
4 rossum (10,) <class 'tuple'> 1
```

In [123…

```
#program for var length arg with loop
def disp(sno,name,*vals):
    print('serial number: {}'.format(sno))
    print('Name :{}'.format(name))
    print('values :{}'.format(vals))
    s=0
    for val in vals :
        print('{} '.format(val))
        s=s+val
        print('sum ={} '.format(s))
#main
disp(1,'alina',10,20,30,40,50)
disp(2,'rossum',10,20,30,40)
disp(3,'travis',10,20 ,30)
disp(4,'gosling',10,20)
```

```
serial number: 1
Name :alina
values :(10, 20, 30, 40, 50)
10
sum =10
20
sum =30
30
sum =60
40
sum =100
50
sum =150
serial number: 2
Name :rossum
values :(10, 20, 30, 40)
10
sum =10
20
sum =30
30
sum =60
40
sum =100
serial number: 3
Name :travis
values :(10, 20, 30)
10
sum =10
20
sum =30
30
sum =60
serial number: 4
Name :gosling
values :(10, 20)
10
sum =10
20
sum =30
```

In [155…

```
#program for var length arg with loop
def disp(sno,name,*vals,ctry='INDIA'):
    print('='*50)
    print('DETAIL')
    print('='*50)
    print('serial number: {} '.format(sno))
    print('Name :{}'.format(name))
    print('Country :{}'.format(ctry))
    print('values :{}'.format(vals))
    print('='*50)
    s=0
    for val in vals :
        print('{} '.format(val))
        s=s+val
        print('sum ={} '.format(s))

#main
disp(1,'alina',10,20,30,40,50 )
disp(2,'rossum',10,20,30,40)
disp(3,'travis',10,20 ,30,ctry='USA')
disp(4,'gosling',10,20)
```

```
==================================================
DETAIL
==================================================
serial number: 1
Name :alina
Country :INDIA
values :(10, 20, 30, 40, 50)
==================================================
10
sum =10
20
sum =30
30
sum =60
40
sum =100
50
sum =150
==================================================
DETAIL
==================================================
serial number: 2
Name :rossum
Country :INDIA
values :(10, 20, 30, 40)
==================================================
10
sum =10
20
sum =30
30
sum =60
40
sum =100
==================================================
DETAIL
==================================================
serial number: 3
Name :travis
Country :USA
values :(10, 20, 30)
==================================================
10
sum =10
20
sum =30
30
sum =60
==================================================
DETAIL
==================================================
serial number: 4
Name :gosling
Country :INDIA
values :(10, 20)
==================================================
10
sum =10
20
sum =30
```

5.KEYWORD VARIABLE LENGTH ARGUMENTS

WHEN WE HAVE FAMILT OF MULTIPLEFUN CALLSEITH VARIABLE NUMBER OF VALUES THEN ,WE MUST DEFINE MULTIPLE FUN DEFINITIONS, THIS PROCESS LEADS TO MORE DEV TIME. VAR LENGTH ARG CONCEPTS,WE MUST DEFINE SINGLE FUN DEFINITON AND TAKES A FORMAL PARAM PROCEDED WITH A SUMBOL CALLED DOUBLE ASTRISK(** PARAM), WHICH ASLO CALLED KEY,VALUES PAIR WHOSE PURPOSE IS TO STORE ANY NUMBER VALUES COMMING FROM FUN CALLS AND WHOSE TYPE IS <class,'dict'>.

In [176…

```
def disp(**alina):
    print(alina,type(alina))
#main
disp(eno=100,ename='rossum',dsg='SE')
disp(a=10,b=20,c=30,d=40)
disp(sn=10,name='alina',)
```

```
{'eno': 100, 'ename': 'rossum', 'dsg': 'SE'} <class 'dict'>
{'a': 10, 'b': 20, 'c': 30, 'd': 40} <class 'dict'>
{'sn': 10, 'name': 'alina'} <class 'dict'>
```

In [192…

```
def disp(**alina):
    print(alina,type(alina))
    for k,v in alina.items():
        print('{} {}'.format(k,v))
```

```python
#main
disp(eno=' :100',ename=' :rossum',dsg=' :SE')
disp(a=10,b=20,c=30,d=40)
disp(sn=' :10',name=' :alina',course=' :python')
```

```
{'eno': ' :100', 'ename': ' :rossum', 'dsg': ' :SE'} <class 'dict'>
eno  :100
ename  :rossum
dsg  :SE
{'a': 10, 'b': 20, 'c': 30, 'd': 40} <class 'dict'>
a 10
b 20
c 30
d 40
{'sn': ' :10', 'name': ' :alina', 'course': ' :python'} <class 'dict'>
sn  :10
name  :alina
course  :python
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js