In [3]:
```
# Functions using def
def task1():
    print('Doing something')
    return "Bye","for","Now"
```

In [4]:
```
print(task1())
```

```
Doing something
('Bye', 'for', 'Now')
```

In [5]:
```
# Positional parameters
def percentageage(roll,name,m1,m2,m3):
    print("Roll :",roll)
    print("Name :",name)
    print("Percentage :",100*(m1+m2+m3)/300)
```

In [6]:
```
percentageage(101,"Trupti",50,60,70)
```

```
Roll : 101
Name : Trupti
Percentage : 60.0
```

In [12]:
```
# Default parameters
def percentageage(roll,name,m1=0,m2=0,m3=0):
    print('roll:',roll)
    print('name:',name)
    print('percentage:',100*(m1+m2+m3)/300)
```

In [13]:
```
percentageage(101,"Trupti",78,45)
```

```
roll: 101
name: Trupti
percentage: 41.0
```

In [16]:
```
# Parameters as Varargs(variables no of arguments)
def keepAdditing(a,b,c=89,*arg):
    print(a,b,c)
    print(arg)
    print("No of var args :",len(arg))
```

In [17]:
```
keepAdditing(50,60)
```

```
50 60 89
()
No of var args : 0
```

In [21]:
```
keepAdditing(50,60,10,20,40,50,9,5)
```

```
50 60 10
(20, 40, 50, 9, 5)
No of var args : 5
```

In [22]:
```
# **a = key word arguments
def acceptRecords(**d):
    print(len(d))
    print(type(d))
    print(d.keys())
    print(d.items())
```

In [25]:
```
acceptRecords(a=67,k=66,j=90)
```

```
3
<class 'dict'>
dict_keys(['a', 'k', 'j'])
dict_items([('a', 67), ('k', 66), ('j', 90)])
```

In [27]:
```
acceptRecords()
```

```
0
<class 'dict'>
dict_keys([])
dict_items([])
```

In [32]:
```
t1={'val1',23}
dt=dict({t1})
print(t1)
```

```
{'val1', 23}
```

In [33]:
```
acceptRecords(**dt)
```

```
1
<class 'dict'>
dict_keys(['val1'])
dict_items([('val1', 23)])
```

In [64]:
```
def acceptRecords(*arg,**d):
    print(arg)
    print(len(d))
    print(type(d))
    print(d.keys())
    print(d.items())
```

In [65]:
```
acceptRecords(2,3,4,5,6,'Hello',a=67,k=66,j=90)
```

```
(2, 3, 4, 5, 6, 'Hello')
3
<class 'dict'>
dict_keys(['a', 'k', 'j'])
dict_items([('a', 67), ('k', 66), ('j', 90)])
```

In [42]:
```
def details(eid,ename,basic,a=5,b=6):
    print('eid:',eid)
    print('ename:',ename)
    gross=basic+a+b
    return print( 'gross Salary:',gross)
```

In [43]:
```
details(1,'Trupti',70000)
```

```
eid: 1
ename: Trupti
gross Salary: 70011
```

In [45]:
```
def details(eid,ename,sal,*a,**b):
    print('eId:',eid)
    print('eName:',ename)
    print ('Salary:',sal)
    print('Skills:',a)
    print('Training :',b)
```

In [47]:
```
details(1,'trupti',70000,'python','c','c++',python=80,c=50)
```

```
eId: 1
eName: trupti
Salary: 70000
Skills: ('python', 'c', 'c++')
Training : {'python': 80, 'c': 50}
```

In [48]:
```
def fact(n):
    fact=1
    for i in range(1,n+1):
        fact=fact*i
    return fact
```

In [51]:
```
fact(4)
```

Out[51]:
```
24
```

In [70]:
```
# Positional argument-complusory to pass all arguement
def emp(eid,ename,age):
    print('Empid:',eid,' Name:',ename,' Age:',age)
```

In [71]:
```
emp(1,'Trupti',21)
```

```
Empid: 1  Name: Trupti  Age: 21
```

In [72]:
```
# Default argument-
def emp(eid,ename,age=25):
    print('Empid:',eid,' Name:',ename,' Age:',age)
```

In [74]:
```
emp(2,'Trupti')
```

```
Empid: 2  Name: Trupti  Age: 25
```

In [76]:
```
# Keyword argument
emp(age=19,ename='nazar',eid=3)
```

```
Empid: 3  Name: nazar  Age: 19
```

In [77]:
```
# varga
def add(a,b,c=0):
    return(a+b+c)
```

In [78]:
```
add(65,67,89)
```

Out[78]:
```
221
```

In [79]:
```
add(45,34)
```

Out[79]:
```
79
```

In [80]:
```
def add(*a): # *-is variable length of arguements
    r=0
    for i in a:
        r=r+i
    return r
```

In [83]:
```
add(54,34,23,54)
```

Out[83]:
```
165
```

In [84]:
```
def add(*a):
    return (sum(a))
```

In [85]:
```
add(6,5,7,8,9,3,2,4)
```

Out[85]:
```
44
```

In [87]:
```
# Kwargs-**
def n(**a):
    print(a)
```

In [91]:
```
n(name='Trupti',percentage=70)
```

```
{'name': 'Trupti', 'percentage': 70}
```

In [93]:
```
def m(**d):
    for k in d:
        print(k,'--->',d[k])
```

In [94]:
```
m(hobby='read',favsport='football',ht=5.8)
```

```
hobby ---> read
favsport ---> football
ht ---> 5.8
```

In [95]:
```
#lambda arguements expression-
x=lambda a,b,c:a+b+c
```

In [97]:
```
x(35,45,5)
```

Out[97]:
```
85
```

In [99]:
```
sq=lambda a:a*a
sq(7)
```

Out[99]:
```
49
```

In [102…
```
# map filter,reduce function
l1=[4,5,6,7,8,9]
```

In [103…
```
# map(function,iterable)
sq=lambda a:a*a
list(map(sq,l1))
```

Out[103…
```
[16, 25, 36, 49, 64, 81]
```

In [104…
```
l2=['Trupti','srushti']
list(map(str.upper,l2))#built in function
```

Out[104…
```
['TRUPTI', 'SRUSHTI']
```

In [110…
```
# map is applying the given function
l2=list(map(lambda x:x*x,l1))
l2
```

Out[110…
```
[16, 25, 36, 49, 64, 81]
```

In [113…
```
# Filter()
l3=[56,67,34,45,65,78]
list(filter(lambda x:x%2!=0,l3))
```

Out[113…
```
[67, 45, 65]
```

In [114…
```
import functools
```

In [116…
```
t=(3,4,5,6)
functools.reduce(lambda a,b:a*b,t)
```

Out[116…
```
360
```

In [118…
```
def fact(a,b):
    print('a :',a,'b :',b)
    return a*b
functools.reduce(fact,t)
```

```
a : 3 b : 4
a : 12 b : 5
a : 60 b : 6
```

Out[118…
```
360
```

In [119…
```
a=lambda x:print('hello',s1)
s1='nagma'
r=lambda s:s%2==0
r(4)
```

Out[119…
```
True
```

In [ ]: