In [4]:
```
def wish(name):
    print("Good morning",name)

wish("vishal")
```

```
Good morning vishal
```

In [11]:
```
l = [1,2,3,5]
b = [6,7,8]
l.append(b)
# Here, append is not a function but a method used with object 'l'.
```

In [16]:
```
def wish():
    print("hello")
print(wish())
```

```
hello
None
```

In [19]:
```
def calc(a,b): # here (a,b) are positional arguments
    sum = a+b
    mul = a*b
    sub = a-b
    div = a/b
    return sum,sub,mul,div
l = calc(100,50)
for i in l:
    print(i)
```

```
150
50
5000
2.0
```

In [ ]:
```
#Write a program to print sum of given numbers using Variable length argument:
```

In [20]:
```
def sum(*n):    # Here n is a tuple
    result = 0
    for x in n:
        result = result + x
    print("the sum :",result)
```

In [21]:
```
sum(1,2)
```

```
the sum : 3
```

In [23]:
```
sum(1,2,3,4,5,6,7,8,9,10)
```

```
the sum : 55
```

In [45]:
```
def display(**kwargs):  # **kwargs is a dictionary by default
    print("record information:")
    for k,v in kwargs.items():
        print(k,".....",v)
```

In [47]:
```
display(name = "vishal",marks = 100,age = 32,location = "hyderabad")
```

```
record information:
name ..... vishal
marks ..... 100
age ..... 32
location ..... hyderabad
```

In [54]:
```
a = 50        # this is global variable
def add(a,b):
    b = 30
    return a+b

add(a,b)
```

Out[54]:
```
80
```

In [55]:
```
a = 50
def add(a,b):
    a = 30    # this is local variable
    b = 30
    return a+b

add(a,b)
```

Out[55]: 60

In [56]:
```
def factorial(n):
    if n == 0:
        result = 1
    else:
        result = n * factorial(n-1)
    return result
```

In [57]:
```
factorial(4)
```

Out[57]: 24

In [ ]:
```
#normal function
```

In [1]:
```
def squareit(n):
    return n*n
```

In [2]:
```
squareit(6)
```

Out[2]: 36

In [ ]:
```
# Lambda Functions
```

In [3]:
```
lambda b : b * b  # b is the input function
```

Out[3]:
```
<function __main__.<lambda>(b)>
```

In [4]:
```
s = lambda b : b * b # lambda is an inbuilt keyword
```

In [5]:
```
s(9)
```

Out[5]:
```
81
```

In [6]:
```
e = lambda a,b : a+b
e(3,4)
```

Out[6]:
```
7
```

In [7]:
```
f = lambda d,f : d if d>f else f
f(100,200)
```

Out[7]:
```
200
```

In [ ]:
```
# function inside another function
# filter , map and reduce
```

In [26]:
```
def iseven(n):
    if n%2 == 0:
        return True
    else:
        return False
```

In [27]:
```
iseven(7)
```

Out[27]:
```
False
```

In [28]:
```
list1 = [1,2,3,4,5,6,7,8,9,10,11]
```

In [29]:
```
l1 = list(filter(iseven,list1))
```

In [30]:
```
l1
```

Out[30]:
```
[2, 4, 6, 8, 10]
```

In [ ]:
```
# above example with lambda function.
```

In [37]:
```
list2 = [1,2,3,4,5,6,7,8,9,10,11]
l2 = list(filter(lambda x : x % 2 == 0 , list2))
```

In [38]:
```
# list of even numbers from the above list
l2
```

Out[38]:
```
[2, 4, 6, 8, 10]
```

In [40]:
```
# to print the odd numbers
list2 = [1,2,3,4,5,6,7,8,9,10,11]
l3 = list(filter(lambda x : x % 2 != 0 , list2))
```

In [41]:
```
l3
```

Out[41]:
```
[1, 3, 5, 7, 9, 11]
```

In [ ]:
```
# map function
```

In [42]:
```
def double(x):
    return 2*x
```

In [45]:
```
list5 = [1,2,3,4,5]
l5 = list(map(double,list5))
```

In [46]:
```
l5
```

Out[46]:
```
[2, 4, 6, 8, 10]
```

In [ ]:
```
# above example with lambda function
```

In [47]:
```
list5 = [1,2,3,4,5]
l6 = list(map(lambda x : 2 * x , list5))
```

In [48]:
```
l6
```

Out[48]:
```
[2, 4, 6, 8, 10]
```

In [ ]:
```
# reduce
```

In [56]:
```
from functools import reduce
l=[10,20,30,40,50]
result=reduce(lambda x,y:x+y,l)
print(result)
```

```
150
```

In [57]:
```
result=reduce(lambda x,y:x*y,l)
print(result)
```

```
12000000
```

In [58]:
```
from functools import *
result=reduce(lambda x,y:x+y,range(1,101))
print(result)
```

```
5050
```

In [59]:
```
print(345)
```

```
345
```

In [ ]:
```
# enumerate functions
```

In [64]:
```
list1 = ["crow","bull","hen","cow","peacock","pig"]
index = 1
for item in list1:
    if index%2 != 0:
        print(f"This is a {item} and it is a bird.")
    index += 1
```

```
This is a crow and it is a bird.
This is a hen and it is a bird.
This is a peacock and it is a bird.
```

In [ ]:
```
# In the above example , we had to create index and item seperately.
# Enumerate function makes this easy.
```

In [65]:
```
for index,item in enumerate(list1):
    if index %2 == 0:
        print(f"This is a {item} and it is a bird.")
```

```
This is a crow and it is a bird.
This is a hen and it is a bird.
This is a peacock and it is a bird.
```