Python: Assignment-1

```
By Ajay Kareti
  -> Reduce()
 > filter()
 -> ZIPL)
 (1) enumerate ():-
 The enumerate() function takes a collection
 (exs a tuple) and returns it as an enumerate object
-> The enumerate() function adds a counter as the
 key of the enumerate object.
 Syntax:- enumerate ( iterable, Start)
     îterable - îterable object
      Start - Starting value ( Default 0)
Ex: x = ('Goa', 'Paris', 'Island', 'Patna')
     y: enumerate (x)
     print ( list (y))
Octput: [ (0, 'Goa'), (1, 'Paris'), (2, 'Island'),
         (3, 'Patna')]
```

```
In [1]: x=('Goa', 'Paris', 'Island', 'Patna')
        y=enumerate(x)
        print(list(y))
```

[(0, 'Goa'), (1, 'Paris'), (2, 'Island'), (3, 'Patna')]

```
The reduce (fun, seq) function is used to appropriate
 a porticular function passed in it's argument to all the list elements mentioned in the organization
 passed along. This function defined in functiontage
 Syntax: reduce (function, sequence [ inttal])
6: from functools import reduce
     def do-sum (x1, x2):
     return x1+x2
     print (reduce ( do-sum, [1,2,3,4]))
Output:- 10
(((1+2)+3)+4)=>10
```

```
In [4]: from functools import reduce
        def do_sum(x1,x2):
           return x1+x2
        print(reduce(do_sum,[1,2,3,4]))
```

Python: Assignment-1

```
(3) map ():-

The may function executes a specified further
for each them to an themble. The them is sent
for each them to as a parameter.

The function as a parameter.

Syntax:- map (function, "fenables)

Ex:- ## Return double of n value

def addition(n):

Teturn n+n

# we double all numbers using map ()

numbers = (10, 20, 30, 40)

Tesult = map (addition, numbers)

print (list (result))

O[p:- [20, 40, 60, 80]
```

```
In [13]: # Return double of n
def addition(n):
    return n + n

# We double all numbers using map()
numbers = (10, 20, 30, 40)
result = map(addition, numbers)
print(list(result))

[20, 40, 60, 80]
```

```
## Filter():-

> filter():-

> filter() function is used for performing data filtering

| procedure on Sequences like lists, tuples and strings.

Syntax:- filter (function, iterable)

Ex:- ## python filter() function

def filterdata(x):

If x<6:

return x

# calling function

result = filter (filterdata, (1,2,6))

## Displaying result

print (list-(result))

Output:- [1,2]
```

```
In [18]: # Python filter() function example
    def filterdata(x):
        if x<6:
            return x
# Calling function
    result = filter(filterdata,(1,2,6))
# Displaying result
    print(list(result))</pre>
[1, 2]
```

Python: Assignment-1

```
The zipes function returns a zip object, which is
 ( 2°p1):-
 an iterator of tuples where the first item in
each passed iterator is paired together, and then
 the second item in each passed iterator are paired
    If the passed iterables have different lengths,
 together etc.
  the "terable with the least- items decided the
  length of the new sterator.
  Ex= a = ("John", " Kumar", " Prathap")
        b = (" Michel", " Ray", " Priya", " Kajo!")
         x = zip (a, b)
       print (tuple (x))
  (('John', 'Michel'), ('hurrar', 'Roy'), ('Prattap', Priya
 Output:-
In [10]: a=("John", "Kumar", "Prathap")
         b=("Michel","Roy","Priya","Kajol")
         X=zip(a,b)
         print(tuple(X))
         (('John', 'Michel'), ('Kumar', 'Roy'), ('Prathap', 'Priya'))
 @ id ():-
 -, The ide function returns a unique id for the
 specified Object.
All objects in python has it's own unique id.
Syntax: id (Object)
        x = ( ' car', 'Bike', 1 Plane')
        y = id(x)
       print (y)
Delput: 22646607066944
every time changes. This is the memory address.
In [11]: X=('Car','Bike','Plane')
         y=id(X)
         print(y)
         2822999750272
```

Python: <u>Assignment-1</u>