Python : map( ) | filter ( ) | reduce( )

map() function.

Python map() function is used to apply a function on all the elements of specified iterable and return map object. Python map object is an iterator, so we can iterate over its elements. We can also convert map object to sequence objects such as list, tuple etc. using their factory functions.

#map(func,*iterables) -> map object

```
lst=[1,2,3,4,5]

def sq(x):
   s=x*x
   return s

m=map(sq,lst)
print("Type is : ",type(m))
lst_sq=list(m)
print("Actual " ,lst)
print("Result is : ",lst_sq)
```

Eg : 2

#map(func,*iterables) -> map object [iterable]

```
'''
lst=[1,2,3,4,5]
m=map(lambda x:x*x,lst)
lst2=list(m)
print("Result is : ",lst2) '''


print("Result is : ",list(map(lambda x:x*x,[1,2,3,4,5])))
```

Eg: 3

```
lst_rad=[1.1, 2.2, 3.3, 4]

def area_of_circle(rad):
   area=3.14*rad*rad
   return area

m=map(area_of_circle,lst_rad)
lst_area=list(m)
print(lst_area)
```

Eg: 4

```
lst_rad=[1.1, 2.2, 3.3, 4]

m=map(lambda rad:3.14*rad*rad,lst_rad)
lst_area=list(m)
print(lst_area)
                or
print(list(map(lambda rad:3.14*rad*rad,lst_rad)))
```

Eg 5 :

```
print(list(map(lambda x:x*x*x,range(1,11))))
```

Reduce Function.

The reduce() function is defined in the functools module.
Like the map and filter functions, the reduce() function receives two arguments, a function and an iterable. ... This function is used in the reduce() function along with a range of numbers between 1 and 4, which are 1,2 and 3.

Syn: reduce(function,sequence) - > int

Eg: 1

```
import functools

#functools.reduce(function,sequence) -> val
lst=list(range(1,6))
print(lst)

s=functools.reduce(lambda x,y:x+y,lst)
print("Sum is : ",s)
```

Eg: 2

```
lst=[1,2,3]
f=functools.reduce(lambda x,y:x*y,lst)
print("Fact is : ",f)
```

filter() in python.

The filter() method filters the given sequence with the help of a function that tests each element in the sequence to be true or not.

syntax: filter(function or None, sequence) -> filter

Parameters:
function: function that tests if each element of a sequence true or not.

Eg 1:

```
lst=["anu","sai","roja","srija","kooja","sri"]
f=filter(lambda x: len(x)==3,lst)
print("Type is : ",type(f))
lst_3=list(f)
print(lst_3)
```

Eg: 2

```
lst=["anu","sai","roja","srija","kooja","sri"]
print("Result : ",list(filter(lambda x:len(x)==3,lst)))
```

Eg: 3:

```
lst=["anu","sai","roja","Srija","kooja","sri","Sudha"]
lst_s=list(filter(lambda x: x.startswith('s'),lst))
print(lst_s)

lst_ss=list(filter(lambda x: x.startswith('s') or x.startswith('S'), lst))
print(lst_ss)

lst_s1=list(filter(lambda x: x.upper().startswith('S'),lst))
print(lst_s1)
```

Eg: 4:

```
lst=["ramesh","","","chinni","Roja","","srija",""]
print("Data From Datasrc : ",lst)

lst_name=list(filter(None,lst))
print("After Filter : ",lst_name)

lst=["ramesh","",0,"chinni","Roja",[],"srija",None,(),False]
lst_r=list(filter(None,lst))
print("Result is : ",lst_r)
```