1) **map():**

It applies a **given function** to each item of an iterable **return a list** of the results

Syntax:

***map(fun, iter)***

Example:

def calculatesquare(n):

return n \* n

l = [1,2,3,4]

res = map(calculatesquare,n)

print(res)

list1 = list(res)  
 print(list)

Output:

<map object at 0x7f6b8e5999e8>

[1, 4, 9, 16]

**2)filter():**

Its only filters the values in iterable which returns True after passing through function .Hence func here returns a Boolean value.

* Returns filter object

Syntax:

***filter(fun,iter)***

Example :

scores = [66, 90, 68, 59, 76, 60, 88, 74, 81, 65]

def is\_A\_student(score):

return score > 75

over\_75 = list(filter(is\_A\_student, scores))

print(over\_75)

Output:

[90, 76, 88, 81]

**3) reduce():**

applies a function of **two arguments** **cumulatively** to the elements of an iterable, optionally starting with an initial argument

Syntax:

***reduce(func, iterable[, initial])***

Example:

from functools import reduce

numbers = [3, 4, 6, 9, 34, 12]

def custom\_sum(first, second):

return first + second

result = reduce(custom\_sum, numbers, 10)

print(result)  
  
Output:

78

Explanation:

If initial is not provided then it takes 1st two elements of iterable as initial argument of function ,otherwise it takes 1st argument as initial and 2nd argument as first element as iterable. It return an integer which acts as first argument for the function when called again and 2nd argument will be next element of iterable.