**PYTHON FUNCTIONALS :**

1. **MAP & LAMDA FUNCTION :**

The map() function applies a function to every member of an iterable and returns the result. It takes two parameters: first, the function that is to be applied and secondly, the iterables.

>> print (list(map(len, ['Tina', 'Raj', 'Tom'])))

[4, 3, 3]

Lambda is a single expression anonymous function often used as an inline function. In simple words, it is a function that has only one line in its body. It proves very handy in functional and GUI programming.

>> sum = lambda a, b, c: a + b + c

>> sum(1, 2, 3)

6

1. **REDUCE :**

The **reduce(fun,seq)** function is used to **apply a particular function passed in its argument to all of the list elements** mentioned in the sequence passed along.This function is defined in “**functools**” module

**import functools**

**# initializing list**

**lis = [ 1 , 3, 5, 6, 2, ]**

**# using reduce to compute sum of list**

**print ("The sum of the list elements is : ",end="")**

**print (functools.reduce(lambda a,b : a+b,lis))**

**# using reduce to compute maximum element from list**

**print ("The maximum element of the list is : ",end="")**

**print (functools.reduce(lambda a,b : a if a > b else b,lis))**

Output:

The sum of the list elements is : 17

The maximum element of the list is : 6

import functools

# importing operator for operator functions

import operator

# initializing list

lis = [ 1 , 3, 5, 6, 2, ]

# using reduce to compute sum of list

# using operator functions

print ("The sum of the list elements is : ",end="")

print (functools.reduce(operator.add,lis))

# using reduce to compute product

# using operator functions

print ("The product of list elements is : ",end="")

print (functools.reduce(operator.mul,lis))

# using reduce to concatenate string

print ("The concatenated product is : ",end="")

print (functools.reduce(operator.add,["geeks","for","geeks"]))

Output

The sum of the list elements is : 17

The product of list elements is : 180

The concatenated product is : geeksforgeeks

1. **FILTER :**

The function filter(function, list) offers an elegant way to filter out all the elements of a list, for which the function *function* returns True.   
The function filter(f,l) needs a function f as its first argument. f returns a Boolean value, i.e. either True or False. This function will be applied to every element of the list *l*. Only if f returns True will the element of the list be included in the result list.

>>> fib = [0,1,1,2,3,5,8,13,21,34,55]

>>> result = filter(lambda x: x % 2, fib)

>>> print result

[1, 1, 3, 5, 13, 21, 55]

>>> result = filter(lambda x: x % 2 == 0, fib)

>>> print result

[0, 2, 8, 34]

>>>

alphabets = ['a', 'b', 'd', 'e', 'i', 'j', 'o']

# function that filters vowels

def filterVowels(alphabet):

vowels = ['a', 'e', 'i', 'o', 'u']

if(alphabet in vowels):

return True

else:

return False

filteredVowels = filter(filterVowels, alphabets)

print('The filtered vowels are:')

for vowel in filteredVowels:

print(vowel)

O/P :

The filtered vowels are:

a

e

i

o