Name: Veeransh Shah

Regno.:221070063

# Experiment 3

## Aim:

Write a program to analyse list using map, reduce, filter methods

## 1. filter

### Theory:

The filter() method is a built-in python function that filters the given set of iterable with the help of a function that tests each element in the sequence to be True or False. It is useful when you have to iterate over a set of elements and differentiate elements on the basis of specific criteria.

Returning an iterator makes filter() more memory efficient than an equivalent for loop.

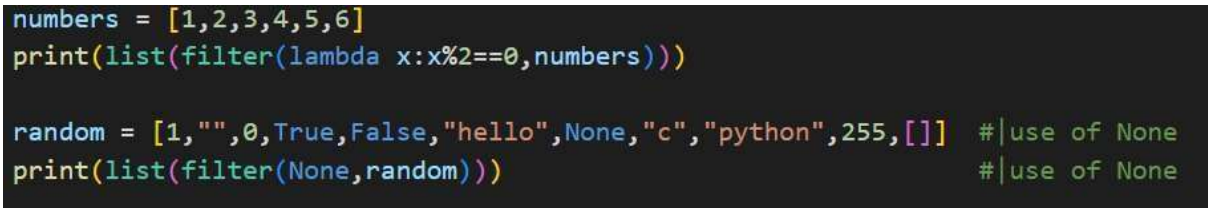
The syntax of filter() function:

filter(Function,sequence)

The filter method takes two arguments:

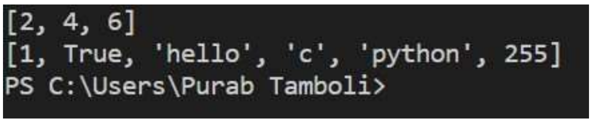
1. Function: It is a User-defined set of rules to be done when a particular function is called.
2. 2. Sequence: It is a set of lists or tuples that need to be filtered.

### Code:



```
numbers = [1,2,3,4,5,6]
print(list(filter(lambda x:x%2==0,numbers)))

random = [1,"",0,True,False,"hello",None,"c","python",255,[]]  #|use of None
print(list(filter(None,random)))                                #|use of None
```

### Output:



```
[2, 4, 6]
[1, True, 'hello', 'c', 'python', 255]
PS C:\Users\Purab Tamboli>
```

## 2. Map

### Theory:

The map() function applies a given function to each element of an iterable (list, tuple etc.) and returns an iterator containing the results.

Its syntax is: map(function,

iterable, ...)

The map() function takes two arguments:

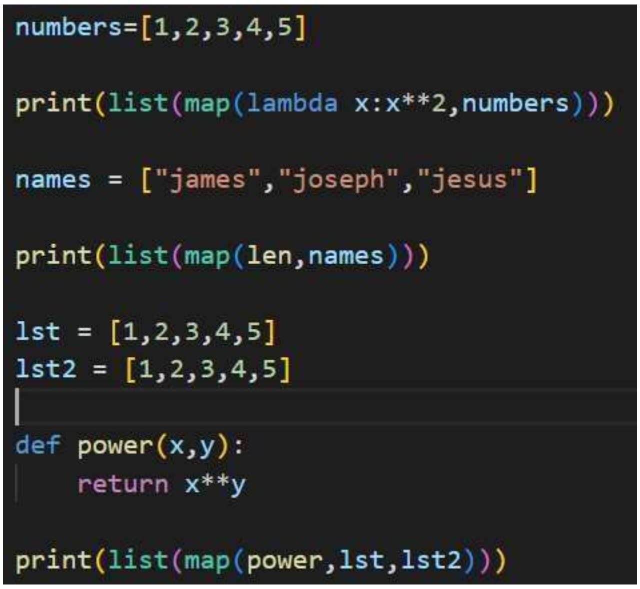
1. function - a function
2. iterable - an iterable like sets, lists, tuples etc

You can pass more than one iterable to the map() function.

The map() function returns an object of map class. The returned value can be passed to functions like

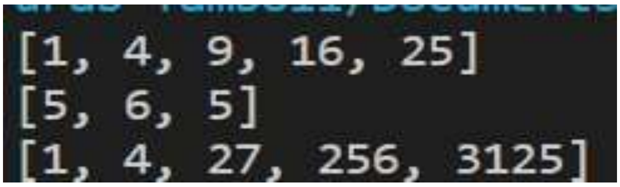
1. list() - to convert to list
2. set() - to convert to a set, and so on.

### Code:



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

print(list(map(lambda x:x**2,numbers)))

names = ["james","joseph","jesus"]

print(list(map(len,names)))

lst = [1,2,3,4,5]
lst2 = [1,2,3,4,5]

def power(x,y):
    return x**y

print(list(map(power,lst,lst2)))
```

### Output:



```
[1, 4, 9, 16, 25]
[5, 6, 5]
[1, 4, 27, 256, 3125]
```

## 3. reduce

### Theory:

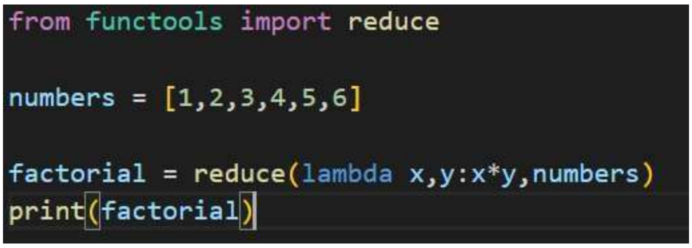
Unlike the map() and filter() functions, the reduce() isn’t a built-in function in Python. In fact, the reduce() function belongs to the functools module.

It doesn't return multiple values; it just returns a single value. Syntax:

functools.reduce(function, iterable)

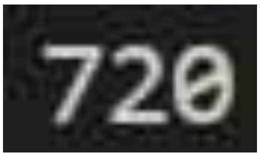
1. The first argument in reduce() is a function. This function will be applied to all the elements in an iterable in a cumulative manner to compute the result.
2. The second argument is iterable. Iterables are those python objects that can be iterated/looped over, including lists, tuples, sets, dictionaries, generators, iterators, etc.

### Code:



```
from functools import reduce

numbers = [1,2,3,4,5,6]

factorial = reduce(lambda x,y:x*y,numbers)
print(factorial)
```

### Output:



```
720
```

## Conclusion:

Hence knowledge of various applications on list using filter, map and reduce functions is learnt in python.