

Name: SIDDHI PAREKH

Reg No.: 221071047

Batch: C

SY CE

Experiment: 3

AIM: To implement filter, map and reduce function in lists in Python

1. Filter functions in list:

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. Sequence: It is a set of lists or tuples that need to be filtered.

CODE:

- a. Checking for even numbers in the list:

```
lst1=[1,2,3,4,5]
```

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

```
def check_num(num):
```

```
    if(num%2 == 0):
```

```
        return True
```

```
    else:
```

```
        return False
```

```
even_num = filter(check_num, lst)
```

```
new_lst7 = list(even_num)
```

```
print(f"Even numbers in the list {lst} are: {new_lst7}")
```

```
33
34 def check_num(num):
35     if(num%2 == 0):
36         return True
37     else:
38         return False
39
40 even_num = filter(check_num, lst)
41 new_lst7 = list(even_num)
42 print(f"Even numbers in the list {lst} are: {new_lst7}")
43
```

Even numbers in the list [1, 2, 3, 4, 5] are: [2, 4]

...Program finished with exit code 0
Press ENTER to exit console.

b. To check for vowel in the list:

```
lst4 = ['a','e','i','o','f','h','k']
```

```
def check_vowel(v):
```

```
    if(v=='a' or v=='e' or v=='i' or v=='o' or v=='u'):
```

```
        return True
```

```
    else:
```

```
        return False
```

```
vowel=filter(check_vowel, lst4)
```

```
new_lst8 = list(vowel)
```

```
print(f"Vowels in our list {lst4} are: {new_lst8}")
```

```

43
44 lst4 = ['a','e','i','o','f','h','k']
45
46 def check_vowel(v):
47     if(v=='a' or v=='e' or v=='i' or v=='o' or v=='u'):
48         return True
49     else:
50         return False
51 vowel=filter(check_vowel, lst4)
52 new_lst8 = list(vowel)
53 print(f"Vowels in our list {lst4} are: {new_lst8}")
54

```

input

```

Vowels in our list ['a', 'e', 'i', 'o', 'f', 'h', 'k'] are: ['a', 'e', 'i', 'o']

...Program finished with exit code 0
Press ENTER to exit console.

```

c. To use filter function along with lambda function:

```
lst4 = ['a','e','i','o','f','h','k']
```

```
vowel=filter(lambda v:v=='a' or v=='e' or v=='i' or v=='o' or v=='u', lst4)
```

```
new_lst9 = list(vowel)
```

```
print(f"Vowels in our list {lst4} are: {new_lst9}")
```

```
43
44 lst4 = ['a','e','i','o','f','h','k']
45
46 # def check_vowel(v):
47 #     if(v=='a' or v=='e' or v=='i' or v=='o' or v=='u'):
48 #         return True
49 #     else:
50 #         return False
51 # vowel=filter(check_vowel, lst4)
52 # new_lst8 = list(vowel)
53 # print(f"Vowels in our list {lst4} are: {new_lst8}")
54
55 vowel=filter(lambda v:v=='a' or v=='e' or v=='i' or v=='o' or v=='u', lst4)
56 new_lst9 = list(vowel)
57 print(f"Vowels in our list {lst4} are: {new_lst9}")
58
```

input

Vowels in our list ['a', 'e', 'i', 'o', 'f', 'h', 'k'] are: ['a', 'e', 'i', 'o']

d. Using the none function to filter out the true and false values in the list:

```
lst5 = [1,0,'c',True,False,"siddhi"]
```

```
check_none = filter(None, lst5)
```

```
new_lst10 = list(check_none)
```

```
print(f"After using none function for the list {lst5}, output is: {new_lst10}")
```

```
58
59 lst5 = [1,0,'c',True,False,"siddhi"]
60 check_none = filter(None, lst5)
61 new_lst10 = list(check_none)
62 print(f"After using none function for the list {lst5}, output is: {new_lst10}")
63
64
```

input

After using none function for the list [1, 0, 'c', True, False, 'siddhi'], output is: [1, 'c', True, 'siddhi']

...Program finished with exit code 0
Press ENTER to exit console.

2. Map functions in list:

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:

- function - a function
- 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

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

CODE:

- a. To find the value of elements raised to elements of another list (involving 2 parameters).

```
lst1=[1,2,3,4,5]
```

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

```
def power(x,y):
```

```

    return(x**y)

new_lst2=list(map(power, lst, lst1))

#new_lst2 = list(map(pow, lst, lst1))

print(new_lst2)

```

```

1  lst=[1,2,3,4,5]
2  lst1=[1,2,3,4,5]
3  #using 1 parameter
4  # def square(n):
5  #     return (n**2)
6
7  # new_lst1=list(map(square , lst))
8  # print(new_lst1)
9
10 #using more parameters
11
12 def power(x,y):
13     return(x**y)
14 new_lst2=list(map(power, lst, lst1))    #new_lst2 = list(map(pow, lst, lst1))
15 print(new_lst2)
16

```

input

[1, 4, 27, 256, 3125]

...Program finished with exit code 0
Press ENTER to exit console.

b. To find the square of elements of the list (involving 1 parameter):

```

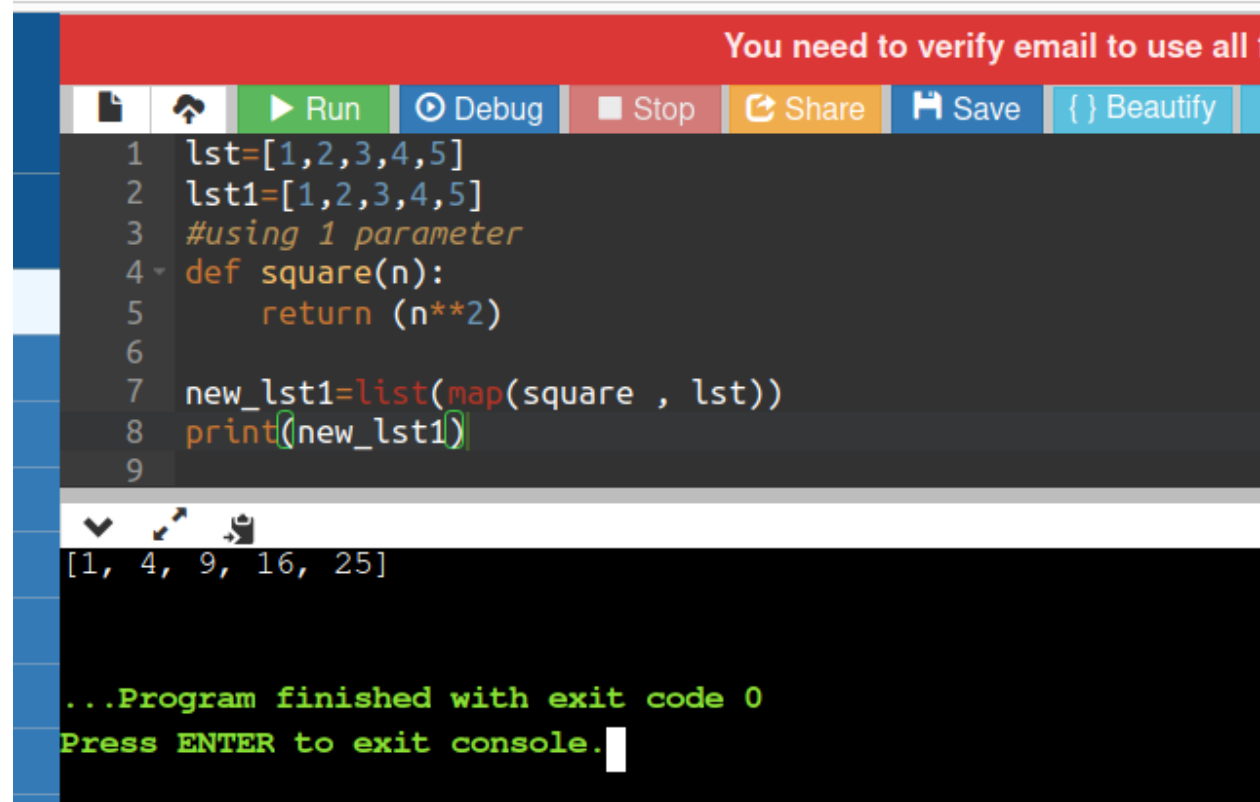
def square(n):

    return (n**2)

new_lst1=list(map(square , lst))

print(new_lst1)

```



The screenshot shows an online Python IDE interface. At the top, there is a red banner that says "You need to verify email to use all". Below the banner is a toolbar with buttons for "Run", "Debug", "Stop", "Share", "Save", and "Beautify". The main editor area contains the following Python code:

```
1 lst=[1,2,3,4,5]
2 lst1=[1,2,3,4,5]
3 #using 1 parameter
4 def square(n):
5     return (n**2)
6
7 new_lst1=list(map(square , lst))
8 print(new_lst1)
9
```

Below the code editor, the output is displayed in a black console window. It shows the list `[1, 4, 9, 16, 25]` and a message: "...Program finished with exit code 0 Press ENTER to exit console."

c. To find the length of each string in a list:

```
lst3=["druhi", "gauri", "richa", "siddhi"]
```

```
new_lst4=list(map(len, lst3))
```

```
print(new_lst4)
```



```
16
17 lst3=["druhi", "gauri", "richa", "siddhi"]
18 new_lst4=list(map(len, lst3))
19 print(new_lst4)
20
```

< [5, 5, 5, 6]

...Program finished with exit code 0
Press ENTER to exit console.

3. Using Reduce functions in list:

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)

- 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.
- 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:

- a. To find the product of all the elements inside the list:

```
from functools import reduce
```

```
def multiply(a,b):
```

```
    x=a*b
```

```
    return x
```

```
lst2=[1,2,3]
```

```
new_lst5=reduce(multiply,lst2)
```

```
print(f"Multiplication of digits of list is: {new_lst5}")
```

```
20
21 from functools import reduce
22
23 def multiply(a,b):
24     x=a*b
25     return x
26
27 lst2=[1,2,3]
28 new_lst5=reduce(multiply,lst2)
29 print(f"Multiplication of digits of list is: {new_lst5}")
30
31 # new_lst6=reduce(lambda x,y: x*y, lst2)
```

```
✓ ↗ 📄
Multiplication of digits of list is: 6
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

b. Using reduce function along with lambda:

```
from functools import reduce
```

```
lst2=[1,2,3]
```

```
new_lst6=reduce(lambda x,y: x*y, lst2)
```

```
print(f"Multiplication of digits of list is: {new_lst6}")
```

```
20
21 from functools import reduce
22
23 # def multiply(a,b):
24 #     x=a*b
25 #     return x
26
27 lst2=[1,2,3]
28 # new_lst5=reduce(multiply,lst2)
29 # print(f"Multiplication of digits of list is: {new_lst5}")
30
31 new_lst6=reduce(lambda x,y: x*y, lst2)
32 print(f"Multiplication of digits of list is: {new_lst6}")
33
```

```
✓ ↗ 🐞
Multiplication of digits of list is: 6
```

```
<
...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION:

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

