Map Filter Reduce

July 13, 2022

1 Map, Filter and Reduce Functions in Python

1.1 Map Function :

[1]:
```
import math

def calculatearea(radius):
    # calculate area of a circle
    area = math.pi * radius * radius
    return area

radii = [4, 5.2, 9, 10, 3,8, 6, 7.5]

areas = [ ]

for radius in radii:
    area = calculatearea(radius)
    areas.append(area)
print(areas)
```

```
[50.26548245743669, 84.94866535306801, 254.46900494077323, 314.1592653589793,
28.274333882308138, 201.06192982974676, 113.09733552923255, 176.71458676442586]
```

[2]:
```
areas = map(calculatearea, radii)
print(list(areas))
```

```
[50.26548245743669, 84.94866535306801, 254.46900494077323, 314.1592653589793,
28.274333882308138, 201.06192982974676, 113.09733552923255, 176.71458676442586]
```

1.2 Map With Lambda

[3]:
```
import math

radii=(4,5.2,9,10,3,8,6,7.5)
print(list(map(lambda x:math.pi*x**2,radii)))
```

```
[50.26548245743669, 84.94866535306801, 254.46900494077323, 314.1592653589793,
28.274333882308138, 201.06192982974676, 113.09733552923255, 176.71458676442586]
```

1.3 Map Function With Multiple Aruguments

[1]:
```
# Multiply two sequences using map and lambda

list_numbers = [1, 5, 8, 9]
tuple_numbers = (11, 20, 54, 23)

map_iterator = map(lambda x, y: x * y, list_numbers, tuple_numbers)
print(list(map_iterator))
```

```
[11, 100, 432, 207]
```

1.3.1 map( ) can listify the list of strings individually

[2]:
```
# List of strings

l = ['sat', 'bat', 'cat', 'mat']

print(list(map( list, l )))
```

```
[['s', 'a', 't'], ['b', 'a', 't'], ['c', 'a', 't'], ['m', 'a', 't']]
```

[5]:
```
a=list('sat')
a
```

[5]:
```
['s', 'a', 't']
```

2 Filter Function:

[1]:
```
scores=[45,70,94.2,75,51,49,35.1]
newscores=[]
# holds scores above average

def isaboveaverage (scores):
    for score in scores:
        if score>50:
            newscores.append(score)␣
 ↪# if score is above average, add it to newscores
isaboveaverage(scores)
print(newscores)
```

```
[70, 94.2, 75, 51]
```

[2]:
```
def student(scores):
    return scores>50
print(list(filter(student,scores)))
```

```
[70, 94.2, 75, 51]
```

2.1 Using lambda within filter( )

[13]:
```
y=filter(lambda x:(x>50),scores)
print(set(y))
```

```
{51, 75, 70, 94.2}
```

2.1.1 Filtering out missing data

[14]:
```
students = ["Jadon", "Solace", "" "Treasure", "", "", "Onyx", "Booboo"]
students
```

[14]:
```
['Jadon', 'Solace', 'Treasure', '', '', 'Onyx', 'Booboo']
```

[15]:
```
newStudents = filter(None, students)
print(list(newStudents))
```

```
['Jadon', 'Solace', 'Treasure', 'Onyx', 'Booboo']
```

3 Reduce function

[16]:
```
from functools import reduce

# Returns the sum of two elements

def sumTwo(a,b):
    return a+b

result=reduce(sumTwo,[23,21,45,98]) # 23 + 21 = 44 , 44+45 = 89 , 89+98 = 187
result
```

[16]:
```
187
```

3.1 Using a lambda function within reduce

[17]:
```
from functools import reduce

# Returns the sum of all the elements using `reduce`

result=reduce((lambda a,b:a+b),[23,21,45,98]) # 23 + 21 = 44 , 44+45 = 89 ,␣
↪89+98 = 187
print(result)
```

```
187
```

3.1.1 Using map( ),filter( ) and reduce( ) along with each other

Using filter( ) within map( )

[9]:
```
c=map(lambda x:x+x,filter(lambda x:(x>=3),(1,2,3,4))) # (3,4)
list(c)
```

[9]:
```
[6, 8]
```

Using map( ) within filter( )

[10]:
```
c = filter(lambda x:(x>=3),map(lambda x:x+x,(1,2,3,4))) # [2,4,6,8] #lambda x:␣
↪(x>=3)
print(list(c))
```

```
[4, 6, 8]
```

Using map( ) and filter( ) within reduce( )

[12]:
```
from functools import reduce
d = reduce(lambda x,y:x+y,map(lambda x:x+x,filter(lambda x:(x>=3),(1,2,3,4))))
d
```

[12]:
```
14
```

3.2 To-Do : Excercises :

Write a Python program to add three given lists using Python map and lambda.

Given a list of strings and a string str, print all anagrams of str using python filter

Write a Python program to find palindromes in a given list of strings using Lambda.

Orginal list of strings: [‘php’, ‘w3r’, ‘Python’, ‘abcd’, ‘Java’, ‘aaa’] ; List of palindromes: [‘php’, ‘aaa’]

```
 $$ $$
 <li>Write a Python program to find the values of length six in a given list using Lambda.
['Nitheesh','Naresh','Mahesh','sai'] ==>['Naresh','Mahesh']</li>
```

Use each of map, filter, and reduce to fix the broken code

[ ]:
```
from functools import reduce


# Use map to print the square of each numbers rounded
# to three decimal places
my_floats = [4.35, 6.09, 3.25, 9.77, 2.16, 8.88, 4.59]
```

```python
# Use filter to print only the names that are less than
# or equal to seven letters
my_names = ["olumide", "akinremi", "josiah", "temidayo", "omoseun"]


# Use reduce to print the product of these numbers
my_numbers = [4, 6, 9, 23, 5]


# Fix all three respectively.
map_result = list(map(lambda x: x, my_floats))
filter_result = list(filter(lambda name: name, my_names, my_names))
reduce_result = reduce(lambda num1, num2: num1 * num2, my_numbers, 0)


print(map_result)
print(filter_result)
print(reduce_result)
```
