Map, Reduce, Filter Functions Eng (1)

February 8, 2023

```python
[1]: l = [1,2,3,4,45,5]
```

```python
[4]: def test(l):
         l1 = []
         for i in l :
             l1.append(i**2)
         return l1
```

```python
[5]: test(l)
```

```
[5]: [1, 4, 9, 16, 2025, 25]
```

```python
[8]: def sq(x):
         return x**2
```

```python
[9]: list(map(sq , l))
```

```
[9]: [1, 4, 9, 16, 2025, 25]
```

```python
[10]: list(map(lambda x : x**2 , l ))
```

```
[10]: [1, 4, 9, 16, 2025, 25]
```

```python
[11]: list(map(lambda x : x+10 , l ))
```

```
[11]: [11, 12, 13, 14, 55, 15]
```

```python
[12]: list(map(lambda x : str(x) , l ))
```

```
[12]: ['1', '2', '3', '4', '45', '5']
```

```python
[13]: l1 = [1,2,3,4,5]
      l2 = [6,7,8,9,10]
```

```python
[14]: list(map(lambda x,y :x+y , l1,l2))
```

```
[14]: [7, 9, 11, 13, 15]
```

```
[16]: list(map(f , l1,l2))
```

```
[16]: [7, 9, 11, 13, 15]
```

```
[15]: f = lambda x,y :x+y
```

```
[17]: s = "pwskills"
```

```
[18]: list(map(lambda x : x.upper()   , s))
```

```
[18]: ['P', 'W', 'S', 'K', 'I', 'L', 'L', 'S']
```

```
[19]: from functools import  reduce
```

```
[20]: l = [1,2,3,4,5,4]
```

```
[21]: reduce(lambda x , y : x+y , l)
```

```
[21]: 19
```

```
[22]: def add(x,y):
          return x+y
```

```
[24]: reduce (add , l)
```

```
[24]: 19
```

```
[25]: reduce(lambda x , y , z : x+y+z , l)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[25], line 1
----> 1 reduce(lambda x , y , z : x+y+z , l)

TypeError: <lambda>() missing 1 required positional argument: 'z'
```

```
[26]: l
```

```
[26]: [1, 2, 3, 4, 5, 4]
```

```
[27]: reduce(lambda x , y : x+y , [])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
```

```
Cell In[27], line 1
----> 1 reduce(lambda x , y : x+y , [])

TypeError: reduce() of empty iterable with no initial value
```

[28]: `reduce(lambda x , y : x+y , [1])`

[28]: 1

[29]: `reduce(lambda x , y : x*y , [1])`

[29]: 1

[30]: `l`

[30]: [1, 2, 3, 4, 5, 4]

[31]: `reduce(lambda x,y : x if x> y else y , l)`

[31]: 5

[32]: `l`

[32]: [1, 2, 3, 4, 5, 4]

[34]: `list(filter(lambda x : x % 2 ==0 , l))`

[34]: [2, 4, 4]

[35]: `list(filter(lambda x : x % 2 !=0 , l))`

[35]: [1, 3, 5]

[36]: `l1 = [-2,4,5,6,-3,-6,-7]`

[38]: `list(filter(lambda x : x < 0 , l1))`

[38]: [-2, -3, -6, -7]

[39]: `list(filter(lambda x : x >= 0 , l1))`

[39]: [4, 5, 6]

[40]: `l2 = ["sudh" , "pwskills" , "kumar" , "bengalore" , "krish"]`

[41]: `list(filter(lambda x : len(x) < 6 , l2))`

```
[41]: ['sudh', 'kumar', 'krish']
```

```
[ ]:
```