# Untitled1

April 7, 2023

### 0.1 Map ,Reduce and Filter function

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

```
[52]: def test1(a):
          n=[]
          for x in a:
              if type(x)==int or type(x)==float:
                  n.append(x**2)
          return n
```

```
[53]: test1(l)
```

```
[53]: [1, 4, 9, 16, 25]
```

```
[30]: square_root= lambda x : x**2
```

```
[31]: square_root(10)
```

```
[31]: 100
```

```
[32]: ## for doing the square for the list
```

```
[37]: def test5(x):
          return x**2
```

```
[43]: list(map(test5,l))
```

```
[43]: [1, 4, 9, 16, 25]
```

```
[55]: ## map helps to pass one by one element
```

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

```
[60]: [1, 4, 9, 16, 25]
```

```
[65]: ## conversion
```

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

[62]: [11, 12, 13, 14, 15]

[64]: list(map(lambda x :str(x), l))

[64]: ['1', '2', '3', '4', '5']

[91]: ## map does mapping of the elements

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

[86]: ## do addition operation? parllely

[90]: list(map(lambda x,y : x+y ,l1,l2 ))

[90]: [7, 9, 11, 13, 15]

[ ]:

[92]: s="pwskills"

[95]: list(map(lambda x : str(x).upper() , s))

[95]: ['P', 'W', 'S', 'K', 'I', 'L', 'L', 'S']

[ ]:
```

## 0.2 Reduce

```
 [97]: from functools import reduce

[110]: l=[1,2,3,4,5,4]

[104]: ## sum of all the numbers in the list

[108]: ## reduce the entire itterable in the list or compress

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

[111]: 19

[114]: def test21(x,y):
           return x+y

[116]: reduce(test21,l)
```

```
[116]: 19
```

```
[117]: ## always ue the 2 arguements
```

```
[ ]:
```

```
[118]: l
```

```
[118]: [1, 2, 3, 4, 5, 4]
```

```
[121]: ## max value of the above  list?
```

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

```
[126]: 5
```

```
[ ]:
```

## 0.3 filter

```
[127]: l
```

```
[127]: [1, 2, 3, 4, 5, 4]
```

```
[128]: ## filter me only the even num and odd num
```

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

```
[133]: [2, 4, 4]
```

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

```
[134]: [1, 3, 5]
```

```
[142]: ## get me only negative value and positive value
```

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

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

```
[140]: [-2, -3, -6, -7]
```

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

```
[141]: [4, 5, 6]
```

```
[ ]:
```

```
[143]: l2=["sudh","pwskills","punith", "Banglore","krish"]
```

```
[145]: ## extract only the name whose len is less  than 6?
```

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

```
[150]: ['sudh', 'krish']
```

```
[ ]:
```

```
[ ]:
```