Day7_CW_DSM

April 29, 2023

0.1 Day7_CW_PWDSM

0.2 4th Feb'23

0.2.1 Lamda Functions

```
[1]: n=3
     p=2
```

```
[2]: def test(n,p):
         return n**p
```

```
[3]: test(3,2)
```

```
[3]: 9
```

```
[4]: test(4,0.5)
```

```
[4]: 2.0
```

```
[5]: lambda n,p : n**p
```

```
[5]: <function __main__.<lambda>(n, p)>
```

```
[6]: a= lambda n,p : n**p
```

```
[7]: a(3,2)
```

```
[7]: 9
```

```
[8]: a(4,0.5)
```

```
[8]: 2.0
```

```
[9]: add = lambda x,y: x+y
```

```
[10]: add(2,3)
```

```
[10]: 5
```

```
[11]: c_to_f = lambda c: (9/5)*c+32
```

```
[12]: c_to_f(34)
```

```
[12]: 93.2
```

```
[14]: max_two = lambda x,y : x if x>y else y
```

```
[15]: max_two(1,2)
```

```
[15]: 2
```

```
[16]: min_two = lambda x,y : x if x<y else y
```

```
[17]: min_two(1,2)
```

```
[17]: 1
```

```
[18]: max_two(1,2)
```

```
[18]: 2
```

```
[19]: name="Driptarshi Ray"
```

```
[20]: len_str = lambda s: len(s)
```

```
[21]: len_str(name)
```

```
[21]: 14
```

0.2.2 Map, Reduce & Filter Functions

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

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

```
[3]: test(l)
```

```
[3]: [1, 4, 9, 16, 25]
```

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

```
[5]: sq(4)

[5]: 16

[6]: map(sq,l)

[6]: <map at 0x7f189680f0a0>

[7]: list(map(sq,l))

[7]: [1, 4, 9, 16, 25]

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

[8]: [1, 4, 9, 16, 25]

[9]: list(map(lambda x: x + 2, l))

[9]: [3, 4, 5, 6, 7]

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

[10]: ['1', '2', '3', '4', '5']

[11]: l1=[1,2,3,4,5]
      l2=[6,7,8,10,11]

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

[12]: [7, 9, 11, 14, 16]

[14]: f = lambda x,y : x-y

[15]: list(map(f,l1,l2))

[15]: [-5, -5, -5, -6, -6]

[16]: s = "qwerty"

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

[17]: ['Q', 'W', 'E', 'R', 'T', 'Y']

[18]: # Reduce
      from functools import reduce

[19]: l=[1,2,3,4,5,6,7,8,9]
```

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

```
[20]: 45
```

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

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

```
[22]: 45
```

```
[23]: reduce(lambda x,y: x-y,l)
```

```
[23]: -43
```

```
[24]: def subtract(x,y):
          return(x-y)
```

```
[25]: reduce(subtract,l)
```

```
[25]: -43
```

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 reduce(lambda x,y,z: x+y+z,l)

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

```
[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,[2])
```

[29]:

```
2
```

[30]:

```
l
```

[30]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[31]:

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

[31]:

```
9
```

[32]:

```
reduce(lambda x,y: x if x<y else y,l)
```

[32]:

```
1
```

[33]:

```
# filter

filter(lambda x: x%2==0,l)
```

[33]:

```
<filter at 0x7f1894542740>
```

[34]:

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

[34]:

```
[2, 4, 6, 8]
```

[35]:

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

[35]:

```
[1, 3, 5, 7, 9]
```

[36]:

```
l1=[-5,-4,-3,-2,-1,1,2,3,4,5]
```

[37]:

```
filter(lambda x: x<0,l1)
```

[37]:

```
<filter at 0x7f189451c130>
```

[38]:

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

[38]:

```
[-5, -4, -3, -2, -1]
```

[39]:

```
l2=["Driptarshi","Ray","Ashirbad","Noapara"]
```

[40]:

```
list(filter(lambda x: len(x)>3, l2))
```

[40]:

```
['Driptarshi', 'Ashirbad', 'Noapara']
```

[ ]:

```

```