25/05/25 Map Reduce & Filter.

Maps: a collection to another collection object based on certain functionality.

→ map(function, iterable object)

→ Example: firstname = ["Adele, Bob, Carrie, John"]

Map the list to obtain the names in uppercase

→ list(map(lambda x: x.upper(), firstname))

→ Filter → Similar function, but it requires the function to look for a condition and then returns only these elements from the collection that satisfies the condition.

Data = [d1, d2, ..., dn]

filter(f, data)

→ Reduce : An operation that breaks down the entire process into pair-wise operations and uses the result from each operation, with the successive element.

Function = f(x,y)

reduce (f, data):

Step1:

Example: Traditional way:

```
import math
def area(r):
    return math.pi*(r**2)
radii = [1, 2, 3, 4, 5]
areas = []
for r in radii:
    a = area(r)
    areas.append(a)
```

I/P: areas

O/P: [3.14151926....,
12.56637....,
28.2743....,
50.2654....,
78.53981....]

i/p: map(area, radii)

To get output: list(map(area, radii))

→ Convert Celsius scale to Farheneit.

F = 9/5*C + 32

Ans: temps = [("Mumbai", 32), ("Vizag", 44), ("Nandyal", 23)]

i/p → Cel_to_F = lambda data: (data[0], (9/5)* data[1] +32)

i/p → map(Cel_to_F, temps)

i/p → list(map(Cel_to_F, temps))

o/p → [('Mumbai', 77.0), ('Vizag', 111.2), ('Nandyal', 73.4)]

* Filter functions: filters out the data

filter(function, data)

i/p: import statistics

data = [1, 2, 3, 4, 5, 9, 12, 4, 2, 9]

i/p: avg = statistics.mean(data)

print(avg)

→ filter(lambda x: x > avg, data)

→ list(filter(lambda x: x > avg, data))

o/p: [9, 12, 9]

→ using filter function, you can get rid of NULL values.

Example

name = ["KK", "preethi", 0]

i/p: filter(None, name)

i/p: list(filter(None, name))

~~o/p: ['Satya', 'N~~

o/p: ["KK", "preethi"]

* Reduce Functions:

i/p: from functools import reduce.

data = [1, 2, 3, 4, 5]

i/p: multiplier = lambda x,y: x*y

reduce(multiplier, data)

o/p: 120

File handling

Basic functions and methods:

1. open()
2. read()
3. close()

Reading a file: file = open('test.txt', 'r')
(filename, Mode)

Three Modes of file roles → reading, writing, executing.

```
i/p: for line in file:
        print(line)
```

we can read the file by:

```
i/p: file = Open('test.txt', 'r')
     print(file.read())
```

i/p: Print(file.read(5))
→ returns 5 characters
o/p: 'Hello'

Writing a file:

```
i/p: file = open('test.txt', 'w')
     file.write("This is a write operation")
     file.close()
```

```
i/p:
print(file.readline())
→ returns first line of file
o/p: Hello world.
```

Append Operation: (→ append o/p)

```
i/p: with open('test.txt', 'a') as file:
        file.write("This is a write operation")
```

```
i/p: file = open('test.txt', 'r')
     for line in file:
         print(line)
```

In operation (new line operation)

```
file = open('test.txt', 'w')
file.write('This is write operation')
file.write("\n")
file.write("Hello! Thanks")
file.close()
```