Functional Programming

- It is used to create small functions
- We can call it as single line functions
- Anonymous functions

```
1.lambda()
      syntax :-  lambda arguments:condition
2.map()
      syntax :- map(function,sequence)
3.filter()
       syntax :- filter(function,sequence)
```

```
# addition of 2 numbers using functions
def add(a,b):
    print(a+b)
add(3,6) # function calling
```

```
9
```

```
# addition of 2 numbers using lambda
k = lambda a,b:a+b
k(8,5)
```

```
13
```

```
# product of 3 numbers using lambda
j = lambda a,b,c:a*b*c
j(2,5,7)
```

```
70
```

```
# square of a number using lambda
square=lambda x:x**2
square(25)
```

```
625
```

```
# Lambda with conditional expression
ev_odd=lambda x: "Even" if x%2==0 else "odd"
ev_odd(8)
```

```
'Even'
```

2.map()

```
* syntax :- map(function,sequence)
```

```python
# map()
def square(a):
    return a*a
m = map(square,[2,4,5,7,8,9])
print(list(m))
```

```
[4, 16, 25, 49, 64, 81]
```

```python
# i/p: ['RAJU','RANI','VAMSI','RAVI']
# O/P: ["raju","rani","vamsi","ravi"]
c = ['RAJU','RANI','VAMSI','RAVI']
res = list(map(str.lower,c))
print(res)
```

```
['raju', 'rani', 'vamsi', 'ravi']
```

```python
# calculate length of each word using map
words = ["apple","banana","cherry"]
length = list(map(len,words))
print(length)
```

```
[5, 6, 6]
```

```python
# adding the two lists element wise
list1 = [1,2,3]
list2 = [4,5,6]
sums = list(map(lambda x,y:x+y,list1,list2))
print(sums)
```

```
[5, 7, 9]
```

Task - 01

convert list of strings to integers using map

i/p: ['1','2','3','4','5']

o/p: [1,2,3,4,5]

filter()

- Which is used to select items from an iterable based on condition

```
3.filter()
       syntax :- filter(function,sequence)

n = [1,2,3,4,5,6,7,8,9]
even = list(filter(lambda x:x%2==0,n))
print(even)
```

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

Task - 02

filter the numbers which is divisible by 5 using filter

```
# filter positive numbers
n1 = [3,4,5,-8,-6,-3,7,8,5]
positive = list(filter(lambda x:x>0,n1))
print(positive)
```

```
[3, 4, 5, 7, 8, 5]
```

```
# filter palindromes in a list
words = ["level","mom","dad","week","lambda","map","python"]
palindrom = list(filter(lambda word:word==word[::-1],words))
print(palindrom)
```

```
['level', 'mom', 'dad']
```

Task - 03

filter words starting with a specific letter

i/p: ["apple","banana","mango","ant","ascii","grape"]

o/p: ["apple","ant","ascii"]

```
n= ["apple","banana","mango","ant","ascii","grape"]
m=filter(lambda x:x.startswith("a"),n)
print(list(m))
```

```
['apple', 'ant', 'ascii']
```

```
n= ["apple","banana","mango","ant","ascii","grape"]
m=filter(lambda x:x.endswith("a"),n)
print(list(m))
```

```
['banana']
```

Files and file handling

files in python

```
   - By using files we can store data permanently
   format :- .mp3,.mp4,pdf,.ipynb,excel,docu,.jpg,png.....etc
```

How to create a text file 1.by using the open() function we can create the text file 2.here open() function takes 2 arguments syntax:- file variable_name = open('filename',file mode')

modes

1.read() :- 'r'--> Read or view the data in a file 2.write() :- 'w'-- we can edit,add,remove,store the data into a text file 3.append() :- 'a'---> we can add the data without erasing the previous data

How to close the file

- by using the close function we can close the file syntax:- file variable_name.close()

```
# how to create empty text file
f1 = open('data1.txt','w')
print("file created successfully")
f1.close()
```

```
file created successfully
```

```
# to store the data into a file using write method
f2 = open('data1.txt','w')
f2.write('Hello welcome to python programming internship')
f2.close()
print('success')
```

```
success
```

```
# file handling methods
    1.read()
    2.write()
    3.append()
    4.readline()
    5.readlines()
    6.seek()
    7.tell()
    8.split()
```

```
# to store the data into a file using write method
f3 = open('data1.txt','w')
f3.write('python workshop')
f3.close()
print('success')
```

```
success
```

```
# append()
f4 = open('data1.txt','a')
f4.write('\nfiles concept in python')
f4.close()
print("success")
```

```
success
```

```
# to print the entire data in a file
g = open('data1.txt','r')
data = g.read()
print(data)
```

```
python workshop
files concept in python
```

```
# to read n no of characters in a file
f5 = open('data1.txt','r')
print(f5.read(6))
print(f5.read(3))
```

```
python
 wo
```

```
# with () - we don't need to close the close
with open('data1.txt') as f1:
    print(f1.read())
```

```
python workshop
files concept in python
```

```
# readline() -- it prints only 1st line of data in your file
with open('data1.txt') as f2:
    print(f2.readline())
```

```
python workshop
```

```
# readlines() : it prints entire data in your file
with open('data1.txt') as f2:
    print(f2.readlines())
```

```
['python workshop\n', 'files concept in python']
```

```
# total no of characters in a file
with open('data1.txt') as f:
    print(len(f.read()))
```

```
39
```

```
# total no of lines in a file
with open('data1.txt') as f1:
    print(len(f1.readlines()))
```

```
2
```

```
# seek()   :- It is used to change the cursor position
with open('data1.txt') as f:
    print(f.seek(5))
    print(f.read(5))
```

```
5
n wor
```

```
# tell() - It is used to know the cursor position of the file object
with open('data1.txt') as f2:
    print(f2.seek(3))
```

```
    print(f2.read(7))
    print(f2.tell())
```

```
3
hon wor
10
```

```
# split()
with open('data1.txt') as f:
    print(f.read().split())
```

```
['python', 'workshop', 'files', 'concept', 'in', 'python']
```

TASK

Count the total no of words in your file

count the total no of spaces in your file

```
# file existed or not
import os
print(os.path.exists('data1.txt'))
```

```
True
```

```
# file existed or not
import os
print(os.path.exists('data2.txt'))
```

```
False
```

```
# how to remove the file
import os
filename = input('enter filename')
if os.path.exists(filename):
    print('yes')
    os.remove(filename)
    print('file removed successfully')
else:
    print('file does not exist')
```