

In [ ]: *#Agenda of the day :*

1. Files **in** python
2. List Comprehension
- Problem solving on above topics

In [ ]:

*# What is file?*

- Files are named locations on our disk to store related information **in** a Permanent

Key Points:

1. We must store our files **in** a non-volatile memory (Hard disk) - ROM (Read On

*#Why use files in Programming ?*

1. **for** security of our data
2. easy to access files **in** organized way
3. **for** reusable

In [ ]:

*#Example:*

1. Small amount of data - (files) - Very Common Traditional System  
**for** storing 250 + students of python batch2 **in** our system  
We use excel sheets to store all your data
2. Huge amount of data: (Databases)-servers  
Ex: employee data of any software
3. Very Very huge amount of data: (Big Data)  
Ex: Distributed System Server

In [ ]:

*#In Python File Handling follow some order:*

What **is** file handling **or** file data processing ?

- Its a process of performing the some operations of files to perform required **or**  
To get the specific data **from** the files by manipulating, editing..etc

In [ ]:

*#Order:*

1. Open a file
2. read a file **or** write to file (Operations)
3. close the file (we must close the files after our work)

reasons to close the files?

1. If its **not** closed it may be accessed by someone to hack (may mis use your d
2. we must freeze our data everytime after our operations
3. Effective resource management
4. Good Programming practice.

In [ ]:

*#Opening a file in python:*

**for** open a file **in** python we have open()

syntax:

open(filename,mode)

-Here file name **if** name of your created file

-mode **is** what type of operation will be performed on that file.

```
In [1]: #ex:
        open("demo.txt","x")
```

```
Out[1]: <_io.TextIOWrapper name='demo.txt' mode='x' encoding='cp1252'>
```

```
In [ ]:
```

```
In [ ]: #File handling:
        1. File modes (x,r,w,a,t,b)
        2. File operations (open,read,write,close)
        3. File methods (write(),writelines(),read(),readline(),readlines())
```

```
In [ ]: #File Modes:
        Mode                                Operation

        "x"          -          Opens for exclusive file creation
        "r"          -          opens file for reading
        "w"          -          opens a file for writing (create a file new file if it not exists
                                data in it)
                                or
                                erase previous data in that file if it exists with data.

        "a"          -          open a file for appending the data end of the file with erasing
                                the previous data.
        "t"          -          text mode
        "b"          -          binary mode
```

```
In [ ]: #File Methods:
        write operation:

        write() - its write some size of data into file.
        writelines()- to write list of lines into a file
        read operation:

        read() - to read entire data in a file at once. (start character to end character.)
        read(10) - its reads first 5 characters of file data.
        readline() - its reads a one line at a time
        readlines() - its returns the data of your file in a list of lines.
```

```
In [4]: #create and write to a file:
        f=open("python1.txt","x")
```

```
In [ ]: #Os module: (Operating System Module)
        #syntax:
        modulename.methodname()
```

```
In [6]: #to know the current working directory
        import os
        os.getcwd()
```

```
Out[6]: 'C:\\Users\\Lenovo\\Desktop\\Python Workshop'
```

```
In [7]: #To change the one directory to another directory
        os.chdir("E:\\external_backup")
```

```
In [8]: os.getcwd()
```

Out[8]: 'E:\\external\_backup'

In [9]: `open("newfile.txt", "x")`

Out[9]: <\_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1252'>

In [10]: `os.chdir("C:\\Users\\Lenovo\\Desktop\\Python Workshop")`

In [11]: `os.getcwd()`

Out[11]: 'C:\\Users\\Lenovo\\Desktop\\Python Workshop'

In [12]: *#To know the list of file in current working directory:*  
`os.listdir()`

Out[12]: ['.ipynb\_checkpoints',  
'Day-14 File Handling in Python.ipynb',  
'Day-14 Notebook.txt',  
'demo.txt',  
'python.txt',  
'python1.txt']

In [19]: `os.rename("python1.txt", "new1.txt")`

-----  
**PermissionError** Traceback (most recent call last)  
<ipython-input-19-6be40100c529> in <module>  
----> 1 `os.rename("python1.txt", "new1.txt")`

**PermissionError**: [WinError 32] The process cannot access the file because it is being used by another process: 'python1.txt' -> 'new1.txt'

In [18]: `os.getcwd()`

Out[18]: 'C:\\Users\\Lenovo\\Desktop\\Python Workshop'

In [20]: `os.listdir()`

Out[20]: ['.ipynb\_checkpoints',  
'Day-14 File Handling in Python.ipynb',  
'Day-14 Notebook.txt',  
'demo.txt',  
'python.txt',  
'python1.txt']

In [21]: `os.mkdir("day14")`

In [22]: `os.rmdir("day14")`

In [23]: *#Changing of directory*  
`os.getcwd()`

Out[23]: 'C:\\Users\\Lenovo\\Desktop\\Python Workshop'

In [24]: *#to change into another directory:*  
`os.chdir("F:\\Windows10Upgrade")`

```
In [25]: #to check directory is changed or not  
os.getcwd()
```

```
Out[25]: 'F:\\\\Windows10Upgrade'
```

```
In [27]:
```

```
In [28]: os.chdir("C:\\\\Users\\\\Lenovo\\Desktop\\Python Workshop")
```

```
In [38]: #Writing data into a file:  
f=open("cse.txt","w")  
f.write("Today is Tuesday\n") #Its write only one line  
f.write(" hello Everyone")  
f.writelines(["\n1.Python is fun\n","2.Data Science\n","3.Machine Learning\n","4.Django\n"])  
f.close()
```

```
In [60]: #Reading data from a file:  
f= open("cse.txt","r")  
#f.read() #reads the entire data at once  
#print(f.read(6))  
#print(f.read(6)) #reads the data with some size  
#print(f.read(16))  
#print(f.readline()) #reads one line at a time  
#print(f.readline())  
print(f.readlines()) #returns data in the form of list of lines
```

```
['Today is Tuesday\n', ' hello Everyone\n', '1.Python is fun\n', '2.Data Science\n', '3.  
Machine Learning\n', '4.Django\n', '5.AI']
```

```
In [65]: #Binary file  
f=open("binaryfile","wb")  
data = [123]  
f.write(bytearray(data))  
f.close()
```

```
In [66]: f= open("binaryfile","rb")  
f.read()
```

```
Out[66]: b'{'
```

```
In [ ]:
```