**FILES AND DIRECTORIES:**

* Data storage is Non Volatile
* Default it works on .txt format files.
* For .pdf to be used, we should import PyPdf2.
* For .csv to be used, we should import csv.
* For .xslx to be used, we should import xslxwrite.
* To write or to read a file we will access it through a file object. But not direct by the code.
* There are different types of operational modes. They are:
  + r - read
  + w - write
  + a - append
  + x - exclusive creation
  + b - binary
  + t – update
* Files will not be in ASCII format, but in other format,
  + linux:-UTF-8
  + windows:-cp1252
* file object creation:
  + the basic operations are open()
    - operate()
    - close()
  + the first step in file object creation is open() function
    - open() uses two arguments
      * 1. Path of the file
      * 2.mode
  + example: fp=open(“path”,”mode”)
  + it throws an error if the file does not exist
  + by default the mode is set as read mode
  + we can also give encoding as UTF-8 as argument in open() function
  + to create a file but it do nothing we use exclusive creation
  + example: fp=open(“path”,”x”)
  + we can also create a file with write mode
  + whenever we open a file in write mode two things will happen
  + 1.if exist open in write mode
  + 2.if not exist create and open in erite mode
  + After we do operations in the file and we close the file
  + Example: fp.close()
  + We can create a file object only using with open() function
  + If we don’t close the file it will open in system interpreter
  + To get security to our files we must close the file
  + Without writing close() function we can also close file using “with”
  + Example: with open(“path”,”mode”) as fileobject:
    - * We don’t need to write the close() because when we use the with it automatically close the file.
* **Writing on files**: there are two methods to writing on files.
  + write()
  + writelines()
  + **write() :**
    - write function can only write single line .it doest provide new line in the file.
    - We used give new line by manually as frw.write(‘hello\n’)
  + **writelines()**  it is used to write collection of multiple lines like list.
    - Example: l\_list=[‘hello’,’hi’,’bye’]
    - fwr.writelines(l\_list)
    - if we open the above file using the write mode and write something then the previous data will be gone and new data will be updated.
    - To avoid this we use append.it add the cursor at the last
    - Append is used to write data with out moving the previous data .when the file is append for second time.
    - Example:fwr=open(‘path of file’,’a’)
* **Reading of files**  there are three methods in reading files.
  + read()
  + readline()
  + readlines()
  + read( ) with open(‘path’,’r’) as fr:’
  + fr.read() it print entire content of file
  + fr.read(10) it read upto 10 characters
  + fr.readline() it print the data from the Cursor to the \n
  + fr.readline(2) it read 2 characters
  + fr.readlines() it print the list of all the lines till \n
  + fr.readlines(4) it will print only 4 characters
* **tell() and seek():**
  + tell() and seek() are another special functions used to know the cursor position
  + fileobject.tell() gives the index or position of the cursor
  + it work on any mode
  + The method tell() returns the current position of the file read/write pointer within the file.
  + Syntax: fileobject.tell()
  + This method returns the current position of the file read/write pointer within the file.
  + Seek():
  + Seek() function is used to set the position in particular point on file
  + Seek() will take two arguments i.e offset and whence
  + Offsetis the position to which we want to change the cursor position
  + Syntax: fileobject.seek(“offset”,whence=[])
  + Seek() work on three modes
    - mode 0 start
    - mode 1 current
    - mode 2 end
  + if the cursor position is at the 5 line ie fr.tell()
  + fr.seek(10)->default it starts from 0 and it comes under mode 0.
  + fr.seek(10,1)->this is mode1 and in that from the current position that means from 5 it goes to 10 positions infront.so the pointer is at 15 position
  + fr.seek(10,2)->it is mode2 and it set the pointer at the end and then set to 10th position from the end.
  + There is no return value.
  + Note that if the file is opened for appending using either ‘a’ or ‘a+’ ,any seek() operation will be undone at the next write.
  + If the file is opened for writing in append mode using ‘a’ , this method is essentially a no-op.but it remains useful for files opened in append mode with reading enable(mode ‘a+’)
  + If the file is opened in text mode using ‘t’,only offset returns by tell() are legal .use of other offsets causes undefined behaviour.
  + Note that not all objects are seekable .
* **Directories :**
  + Directories is a collection of files .and folders having multiple files is a directory
  + If we want to make a directory as a package add the \_\_init\_\_.py file to the folder containing all the .py files then it is a package.
  + The \_\_init\_\_.py file is an empty file it consist nothing.
  + Directories are os depended to work with directories we import os
  + Os.getcwd->to get the current working directory .
  + Os.mkdir(‘xyz’)->it only create a directory .
  + Os.mkdirs(‘xyz\python’)-> it make directory in a directory .
  + Os.rmdir(‘xyz’)->it only removes empty directories.
  + Os.removedirs(‘xyz\python’)->it removes multiple directories which are empty.
  + Import shutil->it is used to remove directories which have content.
  + **Os.walk(‘path’):** it give 3 lists of
  + path
  + directories
  + file