File Handling

- Sometimes, it is not enough to only display the data on the console. The data to be displayed may be very large, and only a limited amount of data can be displayed on the console, and since the memory is volatile, it is impossible to recover the programmatically generated data again and again.
- However, if we need to do so, we may store it onto the local file system which is volatile and can be accessed every time. Here, comes the need of file handling.

Opening a file

- Python provides the **open()** function which **accepts two arguments**, **file name and access mode** in which the file is accessed. The function returns a file object which can be used to perform various operations like reading, writing, etc.
- The **syntax** to use the **open() function** is given below.
- file object = open(<file-name>, <access-mode>)
- The **files** can be **accessed** using various **modes** like **read, write, or append**. The following are the details about the access mode to open a file.

SN	Access Mode	Description
1	R	It opens the file to **read-only**. The file pointer exists at the beginning. ***The file is by default open in this mode if no access mode is passed.***
2	Rb	It opens the file to read only in binary format. The file pointer exists at the beginning
3	r+	It opens the file to read and write both. The file pointer exists at the beginning of the file
4	rb+	It opens the file to read and write both in binary format. The file pointer exists at thefile.
5	W	It opens the file to **write only**. It overwrites the file if previously exists or createsa new one if no file exists with the same name. The file pointer exists at the beginning of the file.
6	Wb	It opens the file to write only in binary format. It overwrites the file if it exists previously or creates a new one if no file exists with the

		same name. The file pointer exists at the beginning of the file.
7	w+	It opens the file to write and read both. It is different from r+ in the sense that it overwrites the previous file if one exists whereas r+ doesn't overwrite the previously written file. It creates a new file if no file exists. The file pointer exists at the beginning of the file.
8	wb+	It opens the file to write and read both in binary format. The file pointer exists at the beginning of the file.
9	A	It opens the file in the append mode. The file pointer exists at the end of the previously written file if exists any. It creates a new file if no file exists with the same
10	Ab	It opens the file in the append mode in binary format. The pointer exists at the end of the previously written file. It creates a new file in binary format if no file existsname.
11	a+	It opens a file to append and read both. The file pointer remains at the end of the filecreates a new file if no file exists with the same name.
12	ab+	It opens a file to append and read both in binary format. The file pointer remainsat the end of the file.

- Example(open the file file.txt in read mode.):
  - fileptr = open("E:\\SAPALOGY\\file2.txt","r")
  - print("File created..")

The close() method

- Once all the operations are done on the file, we must close it through our python script using the **close()** method. Any unwritten information gets destroyed once the close() method is called on a file object.
- We can perform any operation on the file externally in the file system is the file is opened in python, hence it is good practice to close the file once all the operations are done.
- The syntax to use the close() method is given below.
- fileobject.close()