Files:

File is a named location on disk to store the related information. A collection of data or information that has a name is called the file name. All information stored in a computer must be in a file.

Types of files:

i) Data files

ii) Text files

iii) Program files

iv) Directory files

i) Text files

Text file is a sequence of characters stored on a permanent medium like hard drive, flash memory (or) CD-ROM. In addition to printable characters, the text file contains non printing characters. So a text file contain letters (a-z/A-Z), numbers (0-9), special symbols ($,#),non printing characters(\n,\t,\b) etc. Text file should be saved with the extension **.txt**.

ii) File operations

In Python, a file operation takes place in the following order.

3.1.ii.1 Opening the file

3.1.ii.2 Reading and Writing (Perform operation)

3.1.ii.3 Closing the file

Opening the file

When the user wants to read or write to a file, user needs to open it first. Python has a built-in function open () to open a file.

Syntax:

fileobject=open("filename", "access mode")

The parameters are explained below:

filename ❼The filename argument is a string value that contains the name of the file to access.

access mode❼ The access mode denotes the mode in which the file has to be opened (read,write, append, etc).

Example:

```
>>>f=open(“test.txt”,”w”)
```

File opening modes:

There are various modes while opening a file. They are,

Modes	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file.
rb	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer placed at the beginning of the file.
rb+	Opens a file for both reading and writing in binary format. The file pointer placed at the beginning of the file.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
wb	Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
wb+	Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. If the file does not exist, it creates a new file for writing.
ab	Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. If the file does not exist, it creates a new file

	for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. If the file does not exist, it creates a new file for reading and writing.
ab+	Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. If the file does not exist, it creates a new file for reading and writing.

The file object attributes:

Attribute	Description
file.closed	Returns true if the file is closed, otherwise false.
file.mode	Returns the file access mode with which file was opened.
file.name	Returns name of the file.

Closing the file

The function close() of a file object flushes if there is any unwritten information and closesthe file object when there is no more writing can be done. Python closes a file automatically whenthe reference object of a file is reassigned with another file. The syntax of close () function is givenbelow.

Syntax:

```
fileobject.close()
```

Example Program:

```
f1=open("test.txt","w")
print("Name of file is:",f1.name)
print("closed or not:",f1.closed)
print("opening mode is:",f1.mode)
f1.close()
```

Output:

```
Name of file is:test.txt

closed or not:False

opening mode is:w
```