Python - Files I/O

Opening and Closing Files

In Python, a physical file must be mapped to a built-in file object with the help of built-in function open().

Access Modes	Description
r	Opens a file for reading only.
rb	Opens a file for reading only in binary format.
r+	Opens a file for both reading and writing.
rb+	Opens a file for both reading and writing in binary format.
w	Opens a file for writing only.
wb	Opens a file for writing only in binary format.
w+	Opens a file for both writing and reading.
wb+	Opens a file for both writing and reading in binary format.
a	Opens a file for appending.
ab	Opens a file for appending in binary format.
a+	Opens a file for both appending and reading.
ab+	Opens a file for both appending and reading in binary format.

Writing to a File

The following example writes to a physical file.

Example: Writing to File

```
f=open("D:\myfile.txt","w")
f.write("Hello! Learn Python.")
f.close()
```

Reading from a File

Three different methods are provided to read data from file.

- readline(): reads the characters starting from the current reading position up to a newline character.
- read(chars): reads the specified number of characters starting from the current position.
- readlines(): reads all lines until the end of file and returns a list object.

Example: Reading Lines

```
f=open("myfile.txt","r")
line=f.readline()
print(line)
f.close()
```

THE FILE OBJECT ATTRIBUTES

Sr.No.	Attribute & Description
1	file.closed Returns true if file is closed, false otherwise.
2	file.mode Returns access mode with which file was opened.
3	file.name Returns name of the file.
4	file.softspace Returns false if space explicitly required with print, true otherwise.

The Close () Method

The close() method of a file object flushes any unwritten information and closes the file object, after which no more writing can be done.

THE WRITE () METHOD

The write() method writes any string to an open file. It is important to note that Python strings can have binary data and not just text.

THE READ() METHOD

The read() method reads a string from an open file. It is important to note that Python strings can have binary data. apart from text data.

File Positions

Tell():

- The **tell()** method tells you the current position within the file; in other words, the next read or write will occur at that many bytes from the beginning of the file.

seek(offset[, from]) :

- The **seek(offset[, from])** method changes the current file position. The offset argument indicates the number of bytes to be moved. The from argument specifies the reference position from where the bytes are to be moved.