

## Data File Handling in Python

### Data Files

The data files are the files that store data pertaining to specific application, for later use. The data files can be stored in two ways:

- i) Text files
- ii) Binary files

### Text Files

A text file store information in ASCII or Unicode characters. In text file each line of text is terminated, (delimited) with a special character known as EOL(End of Line) character.

### Binary Files

A file is just a file that contains information in the same format in which the information is held in the memory, i.e. the file content that is returned to you is raw. In binary file there is no delimiter for a line. Also no translation occur in binary files. As a result binary files are faster and easier for a program to read and write than are text file.

### Opening and Closing Files

In order to work with a file from within a Python program, you need to open it in a specific mode.

It is done using `open()` function as per one of the following syntaxes:

```
<file_objectname> = open(filename)
```

```
file_objectname > = open(<filename>, <mode>)
```

for e.g.

```
myFile = open("taxes.txt")
```

The above statement opens file "taxes.txt" in file mode as read mode(default mode) and attaches it to file object namely myfile.

Consider another statement:

```
file2 = open("data.txt", "r")
```

The above statement opens file "data.txt" in read mode(because of "r" given as mode) and attaches it to file object namely file2.

Consider another statement:

```
file3 = open("e:\\main\\result.txt", "w")
```

The above statement opens file "result.txt" (stored in folder e:\main\) in write mode and attaches it to file object namely file3.

The Open() function work in following manner:

- i) Python's open() function crates a file object which serves as a link to a file residing on your computer.
- ii) The first parameter for the open() function is path to the file. If just the file name is given that Python searches for the file in the current folder.
- iii) The second parameter of the open function corresponds to a mode which is typically read('r'), write('w'), or append('a'). If no second parameter is given, then by default it opens it in read('r') mode.

As we observed the slashes in the path are doubled. This is because the slashes have special meaning and to suppress that special meaning escape sequence for slash i.e., \\ is given. However if you want to write with single slash, you may write in raw string as:

```
f = open(r"c:\temp\data.txt")
```

The prefix r in front of a string makes it raw string that mean there is no special meaning attached to any character.

Another way of Opening Files - There is one more file-open function which is `file()`. This function works identically as that of `open()` and has the same syntax as that of `open()`-- just you need to replace keyword `open` with keyword `file`.

## File Object

File objects are use to read and write data to a file on disk. The File object is used to obtain a reference to the file on disk and open it for a number of different task.

When you use file `open()`, Python stores the reference of mentioned file in the file-object. A file-object of Python is a stream of bytes where the data can be read either byte by byte or line by line or collectively.

## File Access Modes

When Python opens a file, it need to know the file-mode in which the file is being opened. A file-mode governs the type of operations(such as read or write or append) possible in the opened file ie it refers to how the file will be used once it's opened.

Text File Mode	Binary File Mode	Description	Notes
'r'	'rb'	Read only	File must exist already, otherwise Python raises I/O error.
'w'	'wb'	Write only	If the file does not exist, file is created. If the file exists, Python will truncate existing data and overwrite in the file. So this mode must be used with caution.
'a'	'ab'	Append	File is in write only mode. If the file exists, the data in the file is retained and new data being written will be appended to the end. If the file does not exist Python will create new file.
'r+'	'r+b' or 'rb+'	Read and write	File must exist otherwise error raised. Both read and writing operations can take place.

'w+'	'w+b' or 'wb+'	Write and read	File is created if does not exist. If file exists, file is truncated(past data is lost)
'a+'	'a+b' or 'ab+'	Write and read	File is created if does not exist. If file exists, file's existing data is retained. Both reading and writing operations can take place.

## Closing Files

An open file is closed by calling the `close()` method of its file object.

The `close()` function accomplishes this task and it takes the following general form:

```
fileHandle.close()
```

For instance, if a file `Master.txt` is opened via file-handle `outfile`. It may be closed by the following statement:

```
outfile.close()
```

`open()` is a in built function(used standalone) while `close()` is the method used with file handle object.