File Handling

Data Files: - The data files are the files that store data pertaining to specific application, for later use. The data files can be store in two ways:  
• Text files  
• Binary files  
  
OPENING AND CLOSING FILES  
  
Opening files: -  
In data file handling through Python, the first thing that you do is open the file. It is done using open() function as per one of the following syntax:  
  
<file object name>=open(<file name>)  
<file object name>=open(<file name>,<mode>)  
  
For example:  
  
stu = open ("students.txt")  
The above statement open file "students.txt" in file mode as read mode (default mode) and attaches it to file object namely stu.  
  
Consider another statement:  
  
stu = open ("students.txt","r")  
The above statement opens the file "students.txt" in read mode (because of "r" given as mode) and attaches it to file object namely stu.  
  
Consider one more file open statement: -  
  
stu = open ("e:\\main\\students.txt","w")  
The above statement open file "students.txt"(stored in folder e:\main) in write mode (because of "w" given as mode) and attaches it to file object namely stu.  
  
• *Please note that when you open a file in read mode, the given file must exist in folder, otherwise Python will raise error.*

File Object/ File Handle: - A file object is a reference to a file on disk. It opens and makes it available for a number of different tasks.  
  
File Access Modes: -

|  |  |
| --- | --- |
| Mode | Description |
| r | Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode. If the specified file does not exist, it will generate FileNotFoundError. |
| 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 will be at the beginning of the file. |
| rb+ | Opens a file for both reading and writing in binary format. The file pointer will be 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, it 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 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. That is, the file is in the append mode. 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. That is, the file is in the append mode. 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. The file opens in the append mode. 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. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing. |

### operations on text file

A text file consists of a sequence of lines. A line is a sequence of characters, stored on permanent storage. In a text file, each line is terminated by a special character, known as End Of Line (EOL). Text file can be created using any text editor. Ex. Myfile.txt.

Opening a text file: Open ()

When we want to read or write a file, we must have to open is first. Open () function takes the name of a file as the first argument. The second argument indicates the mode of accessing the file.

Syntax:

<file variable>=open(file name, access mode)

Modes for opening a file:

read(r) : to read the file

write(w): to write the file

append(a):  to write at the end of file

Ex:

>>> f=open(‘myfile.txt’)

>>> print(f)

Output:

<\_io.TextIOWrapper name='myfile.txt' mode='r' encoding='cp1252'>

Closing file: close()

Syntax:  file\_Object.close()

Example:

f=open('myfile.txt')

print("The file which is to be open using the given command is:",f.name)

f.close()

Output:

The file which is to be open using the given command is: myfile.txt

Properties of File Object:

name: shows the file name of opened file

mode: shows Mode in which the file gets opened

readable: returns Boolean value, which indicates whether the file is readable or not

closed: returns Boolean value, which indicates whether the file is closed or not

Example:

f=open('myfile.txt', 'w')

print("Name of File:",f.name)

print("Mode of File:",f.mode)

print("File readable :",f.readable())

print("File closed:",f.closed)

f.close()

print("File closed:",f.closed, f.name)

Output:

Name of File: myfile.txt

Mode of File: w

File readable : False

File closed: False

File closed: True myfile.txt

Reading form a file:

We can read character data from text file by using the following read methods:

read(): To read the entire data from the file; starts reading from the cursor up to the end of the file.

Synatx:

file\_Object.read()

f=open('myfile.txt', 'r')

data=f.read()

print(data)

f.close()

Output:

== RESTART: C:/Users/ATC/AppData/Local/Programs/Python/Python38-32/filehandling.py =

Hello,

Welcome to the CBSE class 12 students

Having computer science with python

You are learning Python file handling

>>>

read(n): To read ‘n’ characters from the file, starting from the cursor.

Example: Here we will read only first 21 characters from the file.

f=open('myfile.txt', 'r')

data=f.read(21)

print(data)

f.close()

Output:

Hello,

Welcome to the

readline(): To read only one line from the file; starts reading from the cursor up to, and including, the end of line character.

readlines(): To read all lines from the file into a list; starts reading from the cursor up to the end of the file and returns a list of lines.