Week -12

Files Concept; features; file operations; Opening Files; Closing Files; Writing to Files; Reading to Files; File methods; Working with files using data frame.

Files:

- Files are named locations on disk to store related information.
- These are used to permanently store data in a non-volatile memory (e.g. hard disk).
- File handling in simple it means handling of files such as opening the file, reading, writing, and many other operations.
- A file operation takes place in the following order:
  - Open a file
  - Read or write
  - Close the file

Features of Files:

- A file always has a name.
- A file always takes up storage space.
- A file is always saved in a certain format (.txt, .pdf, .ppt, .doc, .jpeg, etc).
- A file contains information on when it was created and when it was last modified.
- Files usually have access rights**.**

File Operations:

Creating and Opening files:

 Python provides an open() function that 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.

 Syntax:

file_object = open(file_name, mode)

- ☞ Here, file_name is the name of the file or the location of the file that you want to open.
- ☞ The mode in the open function syntax will tell Python as what operation you want to do on a file.

A file can be opened in various modes:

'r' —> Read - Default value. Opens a file for reading, error if the file does not exist.

'a' —> Append - Opens a file for appending, creates the file if it does not exist.

'w' —> Write - Opens a file for writing, creates the file if it does not exist.

'x' —> Create - Creates the specified file, returns an error if the file exists.

'r+'-> Read & Write

'w+'-> Read and Append

Reading data from file:

To read the data from the file, python provides the following built-in methods.

1) read():

Syntax: file_handler. read([size])

- This method is used to read the contents of a file up to a size and return it as a string.
- The argument size is optional, and, if it is not specified, then the entire contents of the file will be read and returned.

2) readline():

Syntax:file_handler.readline()

- This method is used to read a single line from file.

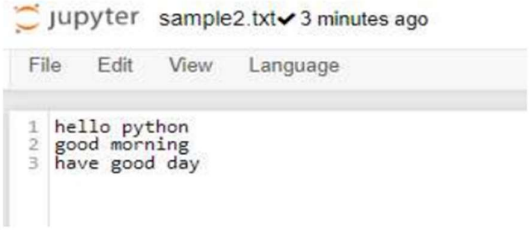
3) readlines():

Syntax: file_handler.readlines()

- This method is used to read all the lines of a file as list items.

Exmple:

```
fobj=open("sample2.txt") # file is opened in read only mode
print(fobj.read(2)) # reads the first two bytes from sample2.txt
print(fobj.readline()) # reads a single line from the file
print(fobj.readlines()) # reads all the lines from file
```



Output:

```
he
llo python

['good morning \n', 'have good day']
```

Writing data to a file:

In order to write data into a file, we must open the file in write mode or in append mode. We have two methods for writing data into a file as shown below.

1) write():

Syntax: file_handler. write(string)

This method will write the contents of the string to the file, returning the number of characters written. If you want to start a new line, you must include the new line character.

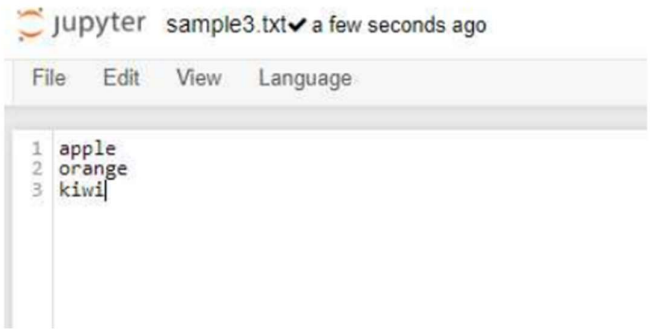
2) writelines():

Syntax: file_handler.writelines(sequence)

This method will write a sequence of strings to the file.

Example:

```
fh=open("sample3.txt","w")
fruits=["apple","\norange"]
fh.writelines(fruits)
fh.write('\nkiwi')
fh.close()
```



The close() method

Once all the operations are done on the file, we must close it through our Python script using the close() method. Once the file is closed we cannot perform any operations on it.

Syntax

fileobject.close()

Exmple:

```
# opens the file file.txt in read mode
fileptr = open("file.txt","r")
if fileptr:
 print("file is opened successfully")
#closes the opened file
```

fileptr.close()

File Object Attributes

When the Python open() function is called, it returns a file object called a file handler.

Using this file handler, you can retrieve information about various file attributes

List of File Attributes

1) file_handler.closed:

It returns a Boolean True if the file is closed or False otherwise.

2) file_handler.mode: It returns the access mode with which the file was opened.

3) file_handler.name: It returns the name of the file

File Methods:

1. read():This method is used for reading the content of the file.
2. readline():This method reads one entire line from the file.
3. readlines(): This method reads entire file and return a list containing the lines.
4. write():This method writes a string to the file.
5. seek(): We can change our current file cursor (position) using the seek() method.
6. tell(): This method returns the current file cursor position (in number of bytes).
7. readable(): This method returns True if the file is readable otherwise it will return False
8. writable(): This method returns True if the file is writable otherwise it will return False