**DATE:22/08/2022**

**ASSIGNMENT NO:04.**

**TOPIC: DIFFERENT FILE MODES IN PYTHON.**

The file handling plays an important role when the data needs to be stored permanently into the file. A file is a named location on disk to store related information. We can access the stored information (non-volatile) after the program termination.

The file-handling implementation is slightly lengthy or complicated in the other programming language, but it is easier and shorter in Python.

In Python, files are treated in two modes as text or binary. The file may be in the text or binary format, and each line of a file is ended with the special character.

Hence, a file operation can be done in the following order.

* Open a file
* Read or write - Performing operation
* Close the file

**Opening File:**

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>, <access-mode>, <buffering>)

The files can be accessed using various modes like read, write, or append. The following are the details about the access mode to open a file.

**r:** It opens the file to read-only mode. The file pointer exists at the beginning. The file is by default open in this mode if no access mode is passed.

**Example:**

myfile=open("file.txt","r")

print(myfile.read())

myfile.close()

**OUTPUT:**

Hi,Goodafternoon

Hi,Goodafternoon

**w:** It opens the file to write only. It overwrites the file if previously exists or creates a new one if no file exists with the same name. The file pointer exists at the beginning of the file.

**Example 1**

myfile=open("file.txt","w")

myfile.write("This Is Written By User")

**OUTPUT 1:**

23

**Example 2:**

myfile=open("file.txt","w")

myfile.write("This Is Written By User")

myfile=open("file.txt","r")

print(myfile.read())

**OUTPUT 2:**

This Is Written By User

**a:** It opens the file in the append mode. The file pointer exists at the end of the previously written file if exists any. It creates a new file if no file exists with the same name.

**Example:**

myfile=open("file.txt","a")

myfile.write("\n This is Written By Programmer")

myfile=open("file.txt","r")

print(myfile.read())

**Output:**

This Is Written By User

This is Written By Programmer

**r+:** It opens the file to read and write both. The file pointer exists at the beginning of the file.

**Example:**

myfile=open("file.txt","r+")

myfile.write("\n Hi,Goodafternoon")

myfile=open("file.txt","r")

print(myfile.read())

**Output:**

Hi,Goodafternoon User

This is Written By Programmer

**w+:** It opens the file to write and read both. It is different from r+ in the sense that it overwrites the previous file if one exists whereas r+ doesn't overwrite the previously written file. It creates a new file if no file exists. The file pointer exists at the beginning of the file.

**Example:**

myfile=open("file.txt","w+")

myfile.write("\n Hi,Goodafternoon")

myfile=open("file.txt","r")

print(myfile.read())

**Output:**

Hi,Goodafternoon

**a+:** It opens a file to append and read both. The file pointer remains at the end of the file if a file exists. It creates a new file if no file exists with the same name

**Output:**

myfile=open("file.txt","a+")

myfile.write("\n Hi,Goodafternoon")

myfile=open("file.txt","r")

print(myfile.read())

**Output:**

Hi,Goodafternoon

Hi,Goodafternoon