Python File Handling

Introduction:

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.

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 like a comma (,) or a newline character. Python executes the code line by line. So, it works in one line and then asks the interpreter to start the new line again. This is a continuous process in Python.

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

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

Opening a file

A file operation starts with the file opening. At first, open the File then Python will start the operation. File opening is done with the open() function in Python. This function will accepts two arguments, file name and access mode in which the file is accessed. When we use the open() function, that time we must be specified the mode for which the File is opening. The function returns a file object which can be used to perform various operations like reading, writing, etc.

Syntax:

The syntax for opening a file in Python is given below -

file object = open(<file-name>, <access-mode>, <buffering>)

Program code for read mode:

It is a read operation in Python. We open an existing file with the given code and then read it. The code is given below -

#opens the file file.txt in read mode

```
fileptr = open("file.txt","r")

if fileptr:

    print("file is opened successfully")
```

Program code for Write Mode:

It is a write operation in Python. We open an existing file using the given code and then write on it. The code is given below -

```
file = open('file.txt','w')

file.write("Here we write a command")

file.write("Hello users of JAVATPOINT")

file.close()
```

The close() Method

The close method used to terminate the program. Once all the operations are done on the file, we must close it through our Python script using the close() method. Any unwritten information gets destroyed once the close() method is called on a file object.

The syntax to use the close() method is given below.

Syntax

The syntax for closing a file in Python is given below -

fileobject.close()

Program code for Closing Method:

Here we write the program code for the closing method in Python. The code is given below -

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

Program code 1 for Write Method:

Here we write the program code for write method in Python. The code is given below -

# open the file.txt in append mode. Create a new file if no such file exists.

```
         fileptr = open("file2.txt", "w")
         # appending the content to the file
         fileptr.write('''''''Python is the modern programming language. It is done
any kind of program in shortest way.''')
          fileptr.close()
```

Program code 2 for Write Method:

Here we write the program code for write method in Python. The code is given below -

```
         with open(test1.txt', 'w') as file2:
            file2.write('Hello coders')
            fil2.write('Welcome to javaTpoint')
```

Program code 3 for Write Method:

Here we write the program code for write method in Python. The code is given below -

```
#open the file.txt in write mode.
```

```
fileptr = open("file2.txt","a")

#overwriting the content of the file

fileptr.write(" Python has an easy syntax and user-friendly interaction.")

#closing the opened file

fileptr.close()
```

Program code for read() Method:

Here we write the program code for read() method in Python. The code is given below -

```
#open the file.txt in read mode. causes error if no such file exists.

fileptr = open("file2.txt","r")

#stores all the data of the file into the variable content

content = fileptr.read(10)

print(type(content))

print(content)

fileptr.close()
```

Program code 1 for Read File using For Loop:

Here we give an example of read file using for loop. The code is given below -

```
fileptr = open("file2.txt","r");

        for i in fileptr:

            print(i) # i contains each line of the file
```