File

File is the collection of data that is availabel to a program. We can retrieve and use data stored in a file whenever we required.

Adavantages:

Store data is permanent unles someone remove it. Store data can be shared. It is possible to update when requried

Two types of file:

Textfile: It store in form of chaarcters

Binary file: IT stores data in the form of bytes, a group of 8 bit

Python File Open File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

File Handling:

The key function for working with files in Python is the open() function.

The open() function takes two parameters; filename, and mode.

There are four different methods (modes) for opening a file:

"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

In addition you can specify if the file should be handled as binary or text mode

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

Syntax To open a file for reading it is enough to specify the name of the file:

f = open("demofile.txt") The code above is the same as:

f = open("demofile.txt", "rt") Because "r" for read, and "t" for text are the default values, you do not need to specify them.

```
f = open("demofile.txt", mode='w')
f.write("Hello how are you \n")
```

```
f.write("Hello medi students final year \n")
f.close()

f=open('demofile.txt', mode='r')
data=f.read()
print(data)
f.close()
```

```
Hello how are you
Hello medi students final year
```

```
f=open('demofile.txt', mode='x')
data=f.read()
print(data)
f.close()
```

```
---------------------------------------------------------------------------
-----
FileExistsError                           Traceback (most recent call
last)
C:\PROGRA~1\KMSpico\temp/ipykernel_18308/1798509067.py in <module>
----> 1 f=open('demofile.txt', mode='x')
      2 data=f.read()
      3 print(data)
      4 f.close()

FileExistsError: [Errno 17] File exists: 'demofile.txt'
```

```
f=open('demofile11.txt', mode='x')
f.write("Hello how are you\n")
f.write("Hello medi \n")
f.close()

f=open('demofile11.txt', mode='r')
data=f.read()
print(data)
f.close()
```

```
Hello how are you
Hello medi
```

```
f = open("demofile.txt", mode='a')
f.write("Happy to see both of you")
f.close()

f=open('demofile.txt', mode='rt')
data=f.read()
print(data)
f.close()
```

```
Hello how are you
Hello medi students final year
Happy to see both of you
```

```
f=open('demofile.txt', mode='rb')
data=f.read()
print(data)
f.close()
```

```
b'Hello how are you \r\nHello medi students final year \r\nHappy to
see both of you'
```

File object Method

readable()

This method is used to check whether file is readable or not.It returns TRue if file is readable else returns False

Syntax: file_object.readable()

writable()

This method is used to check whether file is writable or not. It returns True if file is writable else return False

Syntax: file_object.writable()

```
f = open('student.txt',mode='w')
f.write("Hello dear one\n")
f.write("How are you students \n")
print(f.name)
print(f.mode)
f.close()
```

```
student.txt
w
```

```
f=open('student.txt', mode='w')
print(f.readable())
print(f.writable())
f.close()
print(f.closed)
```

```
False
True
True
```

```
f=open('student.txt', mode='r')
print(f.closed)
f.close()
```

```
False
```

```
f= open('student.txt',mode='r')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : r
File Readable : True
File Writeable : False
File Closed : False
File Closed : True
```

```
f= open('student.txt',mode='w')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : w
File Readable : False
File Writeable : True
File Closed : False
File Closed : True
```

```
f= open('student.txt',mode='x')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
---------------------------------------------------------------------------
-----
FileExistsError                           Traceback (most recent call
last)
```

```
C:\PROGRA~1\KMSpico\temp/ipykernel_18308/2683115200.py in <module>
----> 1 f= open('student.txt',mode='x')
      2 print("File Name :",f.name)
      3 print("File Mode :",f.mode)
      4 print("File Readable :",f.readable())
      5 print("File Writeable :",f.writable())

FileExistsError: [Errno 17] File exists: 'student.txt'
```

```
f= open('student.txt',mode='a')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : a
File Readable : False
File Writeable : True
File Closed : False
File Closed : True
```

r+ Open for reading and then writing w+ Open for writing and then reading.It will overwrite existing data.

a+ Open for appending then reading.It won't overwrite existing data.

```
f= open('student.txt',mode='r+')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : r+
File Readable : True
File Writeable : True
File Closed : False
File Closed : True
```

```
f= open('student.txt',mode='w+')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
```

```
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : w+
File Readable : True
File Writeable : True
File Closed : False
File Closed : True
```

```
f= open('student.txt',mode='a+')
print("File Name :",f.name)
print("File Mode :",f.mode)
print("File Readable :",f.readable())
print("File Writeable :",f.writable())
print("File Closed :",f.closed)
f.close()
print("File Closed :",f.closed)
```

```
File Name : student.txt
File Mode : a+
File Readable : True
File Writeable : True
File Closed : False
File Closed : True
```

Manupulating of pointer using seek & tell method

The seek() function in Python is used to move the file cursor to the specified location. When we read a file, the cursor starts at the beginning, but we can move it to a specific position by passing an arbitrary integer (based on the length of the content in the file) to the seek() function The tell() method returns the current file position in a file stream.

```
f3=open ("studentcse.txt",mode="w")
data3=f3.write("iam_a_student_of_cse")

f3=open("studentcse.txt",mode="r")
data=f3.read()
print(data)
```

```
iam_a_student_of_cse
```

```
f=open("studentcse.txt",mode="r")
print(f.tell())
```

```
f.seek(7)
print(f.tell())
data1=f.read()
print(data1)
print(f.tell())
f.seek(2)
print(f.tell())
data2=f.read()
print(data2)
```

```
0
7
tudent_of_cse
20
2
m_a_student_of_cse
```

```
f =open("student.txt", mode='a+')
f.write('Youtube SAFDAR final ')
data =f.read()
print(data)
```

```
f=open("student.txt", mode='a+')
print(f.tell())
f.write('Youtube')
print(f.tell())
data =f.read()
print(f.tell())
print(data)
```

```
49
56
56
```

```
f=open("student11.txt", mode='a+')
print(f.tell())
f.write('Youtube')
print(f.tell())
f.seek(0)
print(f.tell())
data =f.read()
print(f.tell())
print(data)
```

```
44
51
```

```
0
51
Hello dear one
How are you
YoutubeYoutubeYoutube
```

How to copy file contents in python

```
f1 = open("student.txt",mode='r')
f2 = open("student24.txt",mode='w')
data = f1.read()
f2.write(data)
print(data)
f1.close()
f2.close()
```

```
YoutubeYoutube SAFDARYoutubeYoutube SAFDAR final Youtube
```

```
f2=open("student24.txt",mode="r")
data=f2.read()
print(data)
```

```
YoutubeYoutube SAFDARYoutubeYoutube SAFDAR final Youtube
```

```
f3=open("student2.txt",mode="r")
data=f2.read()
print(data)
```

```
f1 = open("student.txt",mode='r')
f3 = open("student2.txt",mode='w')
data = f1.read()
f3.write(data)
print(data)
f1.close()
f3.close()
```

```
YoutubeYoutube SAFDARYoutubeYoutube SAFDAR final Youtube
```

```
f3=open("student2.txt",mode="r")
data=f3.read()
print(data)
```

```
YoutubeYoutube SAFDARYoutubeYoutube SAFDAR final Youtube
```

with statement in python

```
with open("student.txt",mode="r") as f5:
    data = f5.read()
    print(data)
    print(f5.closed)
print(f5.closed)
```

```
YoutubeYoutube SAFDARYoutubeYoutube SAFDAR final Youtube
False
True
```

writelines()

```
f=open("student_writelines.txt", mode='w')
lst=["raj",'mohan','rahul','rani','safdar']
f.writelines(lst)
f.close()

f=open("student_writelines.txt", mode='w')
lst=['raj \n','mohan \n','rahul \n','rani \n']
f.writelines(lst)
f.close()

t=(1,2,4,3,8,9)
[t[i]for i in range(0,len(t),2)]
```

```
[1, 4, 8]
```

```
print('abcdef12'.replace('cd','12'))
```

```
ab12ef12
```

```
stre = "saf"
m = stre.reverse
print(m)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\PROGRA~1\KMSpico\temp/ipykernel_18308/884492296.py in <module>
      1 stre = "saf"
----> 2 m = stre.reverse
      3 print(m)

AttributeError: 'str' object has no attribute 'reverse'
```

```
l =[2,3]
l.add(7)
print(l)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\PROGRA~1\KMSpico\temp/ipykernel_18308/2189205904.py in <module>
      1 l =[2,3]
----> 2 l.add(7)
      3 print(l)

AttributeError: 'list' object has no attribute 'add'
```