Python for Class XII _- Part 6

Topics:

File handling: Need for a data file, Types of file: Text files, Binary files and CSV (Comma separated values) files. *(CSV will be discussed separately)*

Text File: Basic operations on a text file: Open (filename – absolute or relative path, mode) / Close a text file, Reading and Manipulation of data from a text file, Appending data into a text file, standard input /output and error streams, relative and absolute paths.

Binary File: Basic operations on a binary file: Open (filename – absolute or relative path, mode) / Close a binary file, Pickle Module – methods load and dump; Read, Write/Create, Search, Append and Update operations in a binary file. (continued)

How to Open a Text File

To open a file, you need to use the built-in `open` function. The open function returns a file object that contains methods and attributes to perform various operations on the file.

Syntax

```
file_object  = open("filename", "mode")
```

Here,

- **filename:** gives name of the file that the file object has opened.
- **mode:** attribute of a file object tells you which mode a file was opened in.

More details of these modes are explained below

How to Create a Text File

With Python you can create a .text files (student.txt) by using the code, we have demonstrated here

Step 1)

```
        f= open("student.txt","w+")
```

- We declared the variable f to open a file named student.txt. Open takes 2 arguments, the file that we want to open and a string that represents the kinds of permission or operation we want to do on the file
- Here, we used "w" letter in our argument, which indicates write and will create a file if it does not exist in library
- Plus sign indicates both read and write.

Step 2)

```
for i in range(10):
     f.write("This is line %d\n" % (i+1))
```

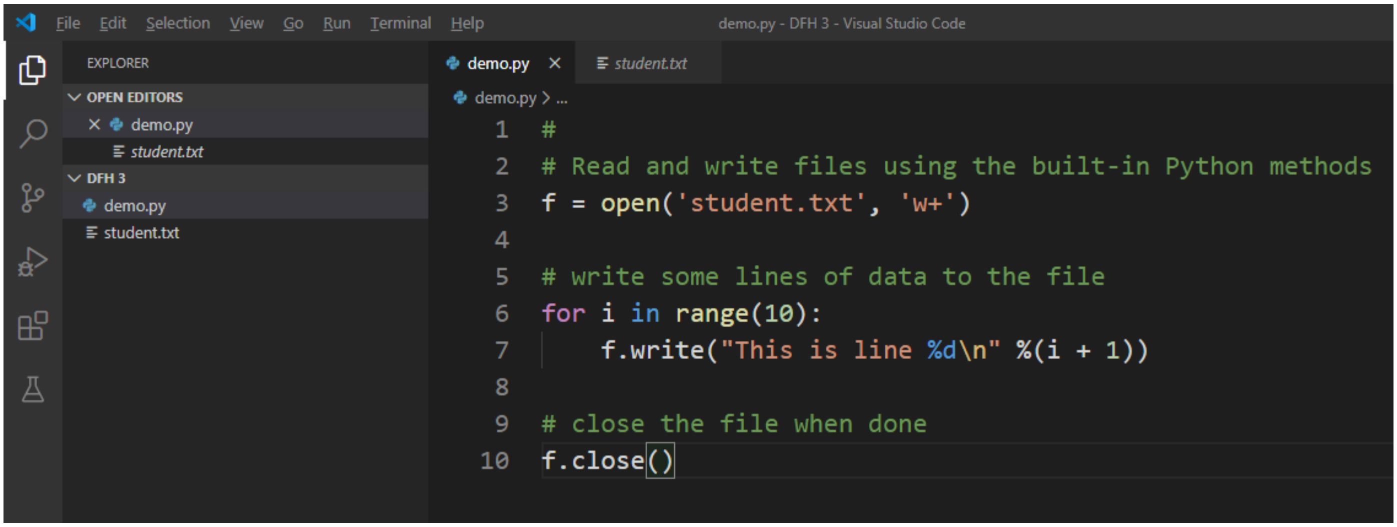
- We have a for loop that runs over a range of 10 numbers.
- Using the **write** function to enter data into the file.
- The output we want to iterate in the file is "this is line number", which we declare with write function and then percent d (displays integer)
- So basically we are putting in the line number that we are writing, then putting it in a carriage return and a new line character

Step 3)

```
        f.close()
```

- This will close the instance of the file student.txt stored

Here is the result after code execution

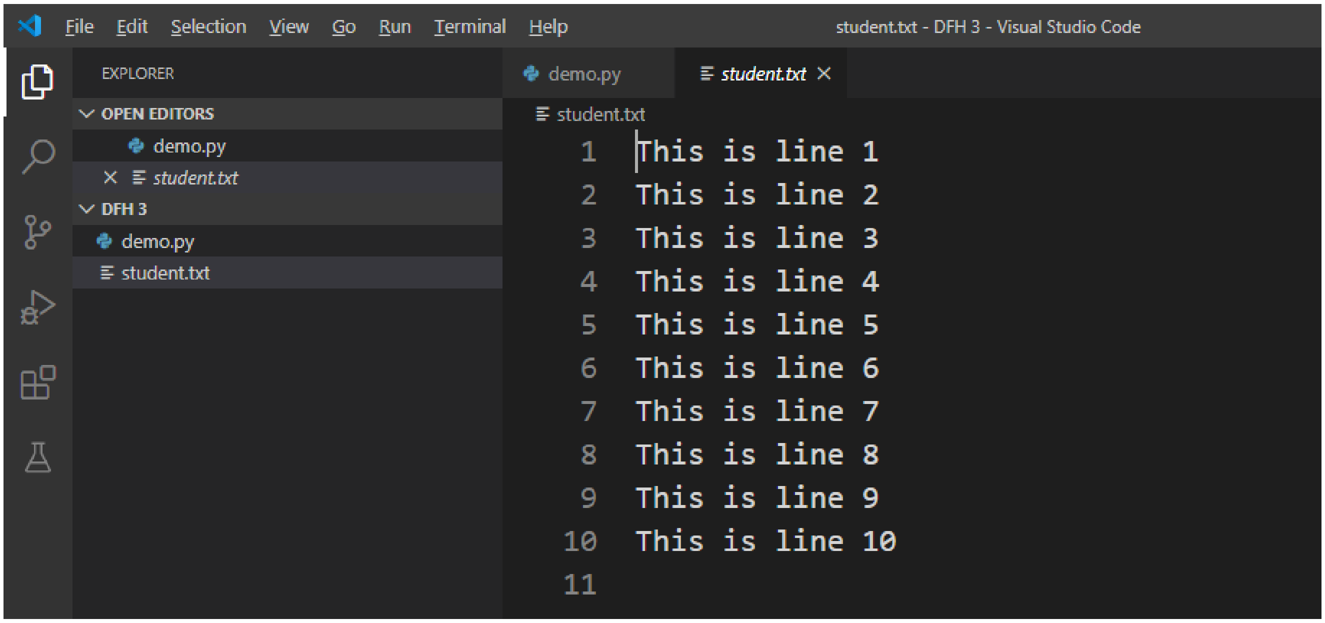


```
#
# Read and write files using the built-in Python methods
f = open('student.txt', 'w+')

# write some lines of data to the file
for i in range(10):
    f.write("This is line %d\n" %(i + 1))

# close the file when done
f.close()
```

When you click on your text file in our case "student.txt" it will look something like this



```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10

```

How to Append Data to a File

You can also append/add a new text to the already existing file or a new file.

Step 1)

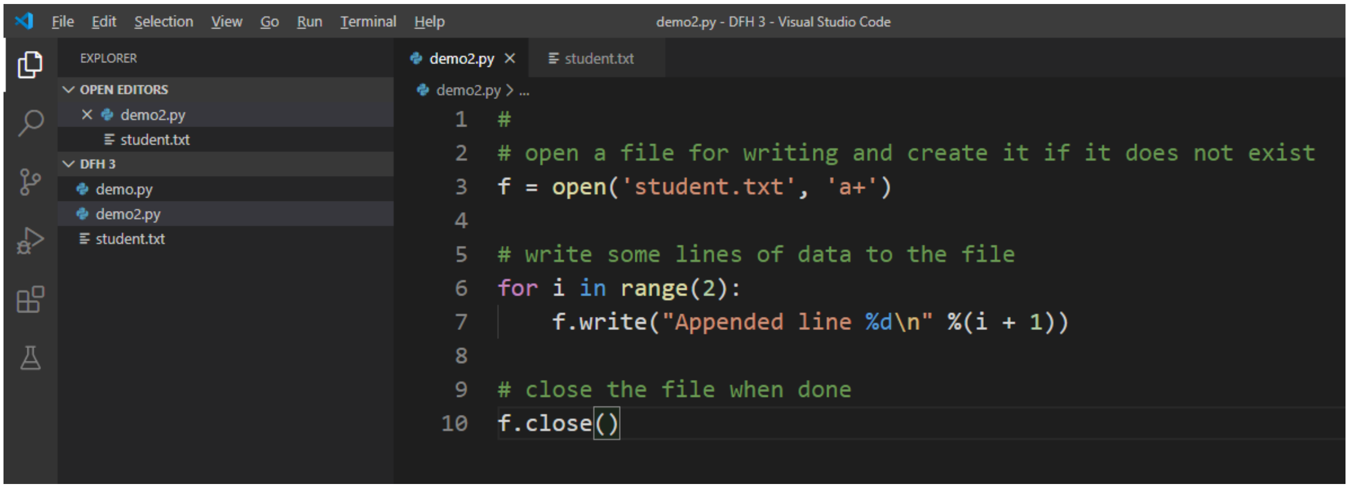
```
f=open("student.txt", "a+")
```

Once again if you could see a plus sign in the code, it indicates that it will create a new file if it does not exist. But in our case we already have the file, so we are not required to create a new file.

Step 2)

```
for i in range(2):
     f.write("Appended line %d\n" % (i+1))
```

This will write data into the file in append mode.

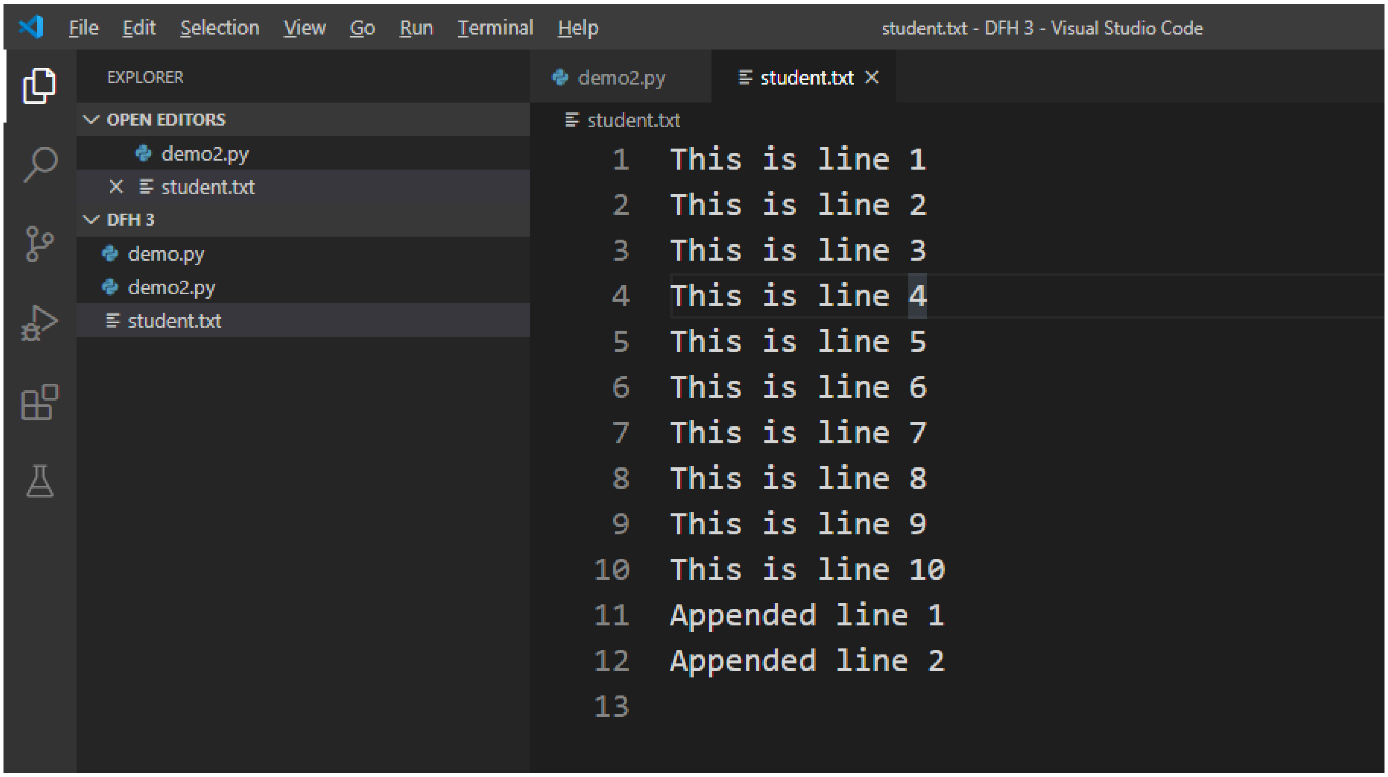


```
#
# open a file for writing and create it if it does not exist
f = open('student.txt', 'a+')

# write some lines of data to the file
for i in range(2):
    f.write("Appended line %d\n" %(i + 1))

# close the file when done
f.close()
```

You can see the output in "student.txt" file. The output of the code is that earlier file is appended with new data.



```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
Appended line 1
Appended line 2

```

How to Read a File

You can read a file in Python by calling .txt file in a "read mode"(r).

Step 1) Open the file in Read mode

```
        f=open("student.txt", "r")
```

Step 2) We use the mode function in the code to check that the file is in open mode. If yes, we proceed ahead

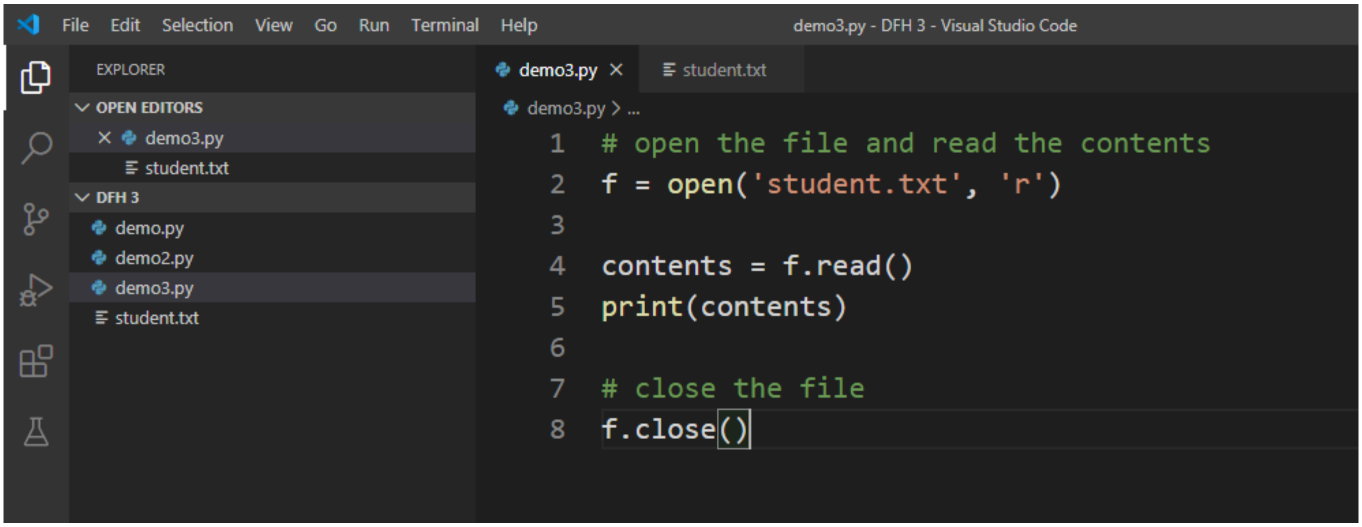
```
        if f.mode == 'r':
```

Step 3) Use f.read to read file data and store it in variable content

```
        contents =f.read()
```

Step 4) print contents

Here is the output



```
# open the file and read the contents
f = open('student.txt', 'r')

contents = f.read()
print(contents)

# close the file
f.close()
```

Output:

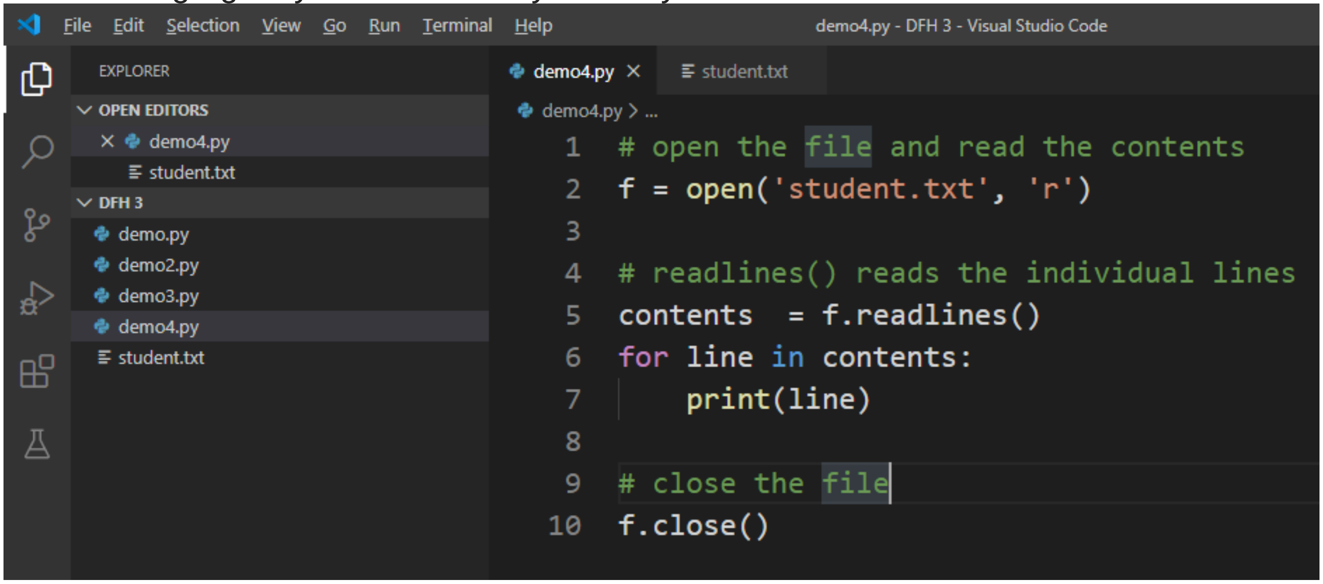
```
$ python demo3.py

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
Appended line 1
```

Appended line 2

How to Read a File line by line

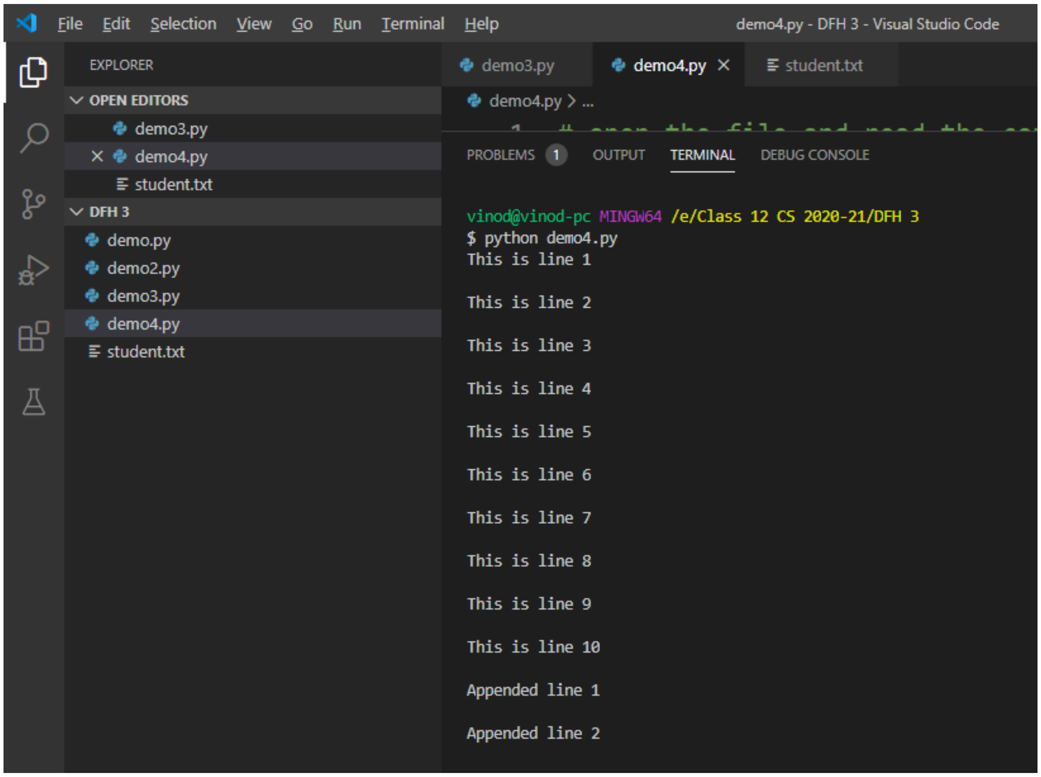
You can also read your .txt file line by line if your data is too big to read. readlines() code will segregate your data in easy to ready mode.



```python
# open the file and read the contents
f = open('student.txt', 'r')

# readlines() reads the individual lines
contents  = f.readlines()
for line in contents:
    print(line)

# close the file
f.close()
```



```
vinod@vinod-pc MINGW64 /e/Class 12 CS 2020-21/DFH 3
$ python demo4.py
This is line 1

This is line 2

This is line 3

This is line 4

This is line 5

This is line 6

This is line 7

This is line 8

This is line 9

This is line 10

Appended line 1

Appended line 2
```

When you run the code (**contents=f.readlines()**) for reading the file or document line by line, it will separate each line and present the file in a readable format. In our case the line is short and readable, the output will look similar to the read mode. But if there is a complex data file which is not readable, this piece of code could be useful.

File Modes in Python

Following are various modes in Python

Mode	**Description**
'r'	This is the default mode. It Opens file for reading.
'w'	This Mode Opens file for writing. If file does not exist, it creates a new file. If file exists it truncates the file.
'x'	Creates a new file. If file already exists, the operation fails.
'a'	Open file in append mode. If file does not exist, it creates a new file.
't'	This is the default mode. It opens in text mode.
'b'	This opens in binary mode.
'+'	This will open a file for reading and writing (updating)

Summary

- Python allows you to read, write and delete files
- Use the function open("filename","w+") to create a file. The + tells the python interpreter to open file with read and write permissions.
- To append data to an existing file use the command open("Filename", "**a**")
- Use the read function to read the ENTIRE contents of a file
- Use the readlines function to read the content of the file one by one.