Python File Handling: Create, Open, Append, Read, Write

In Python, there is no need for importing external library to read and write files. Python provides an inbuilt function for creating, writing and reading files.

In this tutorial, we will learn

- How to Create a Text File
- How to Append Data to a File
- How to Read a File
- How to Read a File line by line
- File Modes in Python

How to Create a Text File

With Python you can create a .text files (guru99.txt) by using the code, we have demonstrated here how you can do this

Step 1)

```
        f= open("guru99.txt","w+")
```

- We declared the variable f to open a file named guru99.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.
- The available option beside "w" are, "r" for read, and "a" for append

Step 2)

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

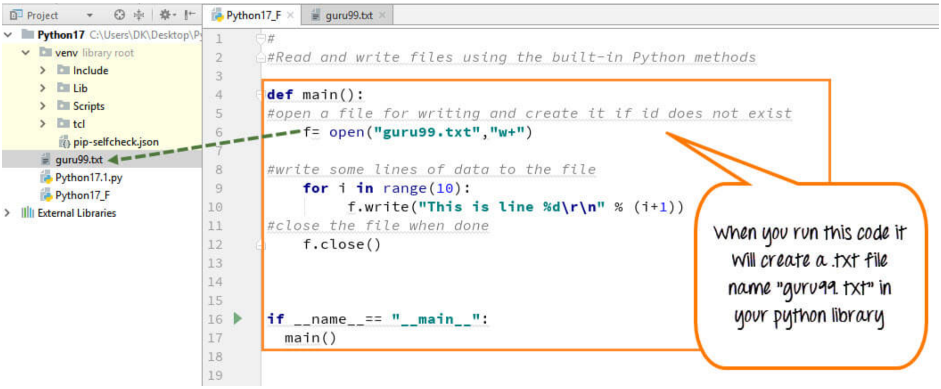
- The output we want to iterate in the file is "this is line number", which we declare with write function and then percent d (displays integer)
- So basically we are putting in the line number that we are writing, then putting it in a carriage return and a new line character

Step 3)

```
f.close()
```

- This will close the instance of the file guru99.txt stored

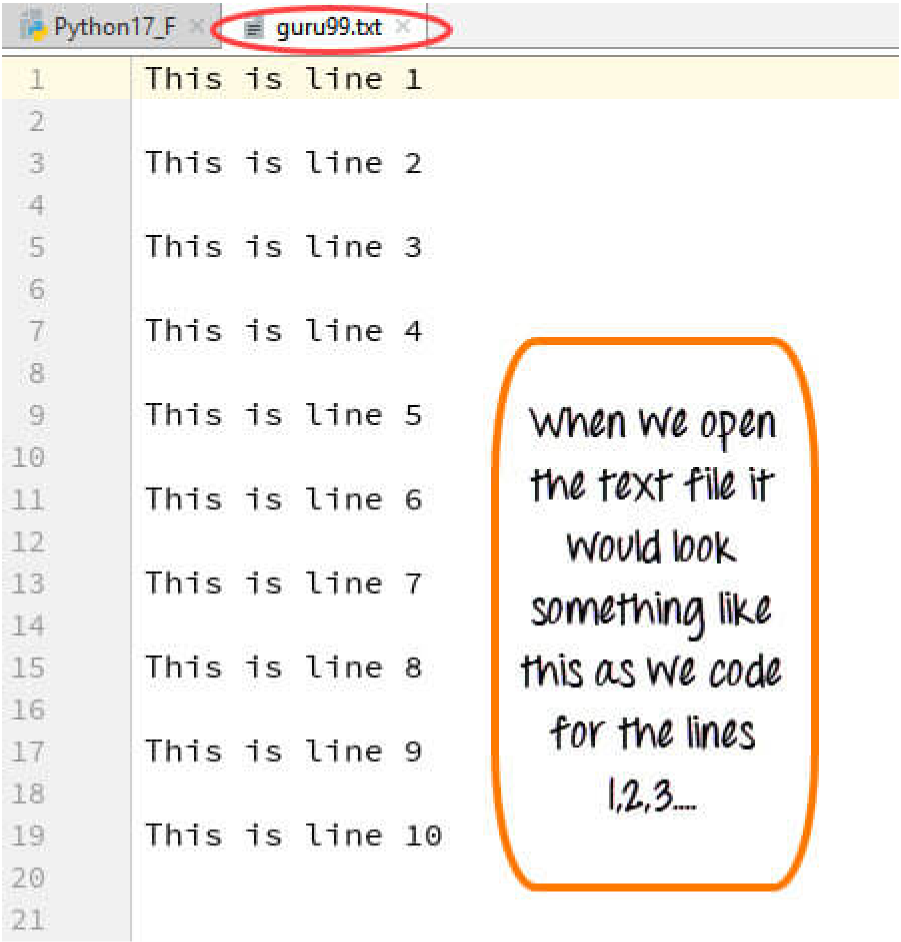
Here is the result after code execution



```
#
#Read and write files using the built-in Python methods

def main():
#open a file for writing and create it if id does not exist
    f= open("guru99.txt","w+")

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


if __name__== "__main__":
  main()
```

When you run this code it will create a .txt file name "guru99. txt" in your python library

When you click on your text file in our case "guru99.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

```

When we open the text file it would look something like this as we code for the lines 1,2,3...

How to Append Data to a File

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

Step 1)

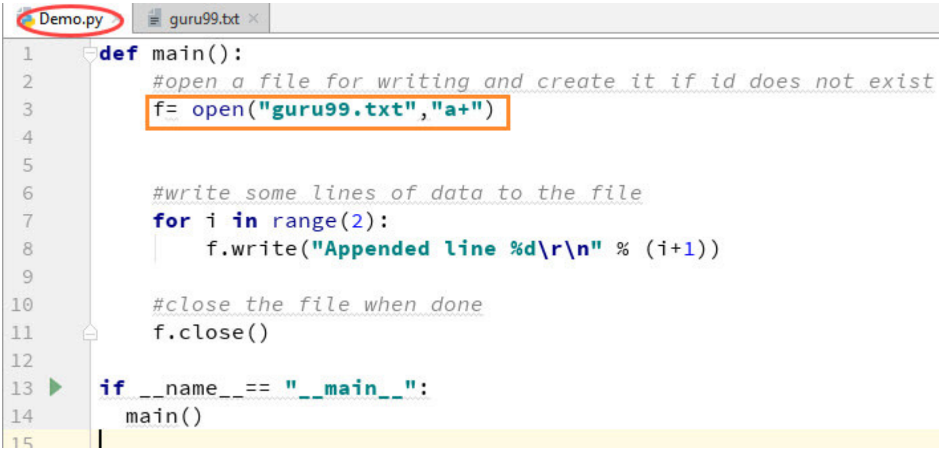
```
f=open("guru99.txt", "a+")
```

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)

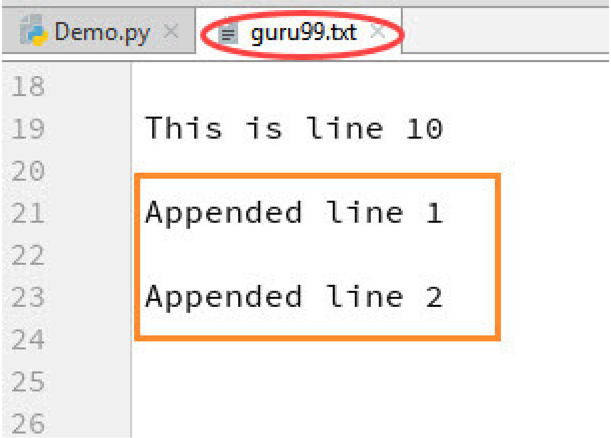
```
for i in range(2):
     f.write("Appended line %d\r\n"
% (i+1))
```

This will write data into the file in append mode.



```python
def main():
    #open a file for writing and create it if id does not exist
    f= open("guru99.txt","a+")


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

    #close the file when done
    f.close()

if __name__== "__main__":
  main()
```

You can see the output in "guru99.txt" file. The output of the code is that earlier file is appended with new data.



```

This is line 10

Appended line 1

Appended line 2
```

How to Read a File

Not only you can create .txt file from Python but you can also call .txt file in a "read mode" (r).

Step 1) Open the file in Read mode

```
f=open("guru99.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

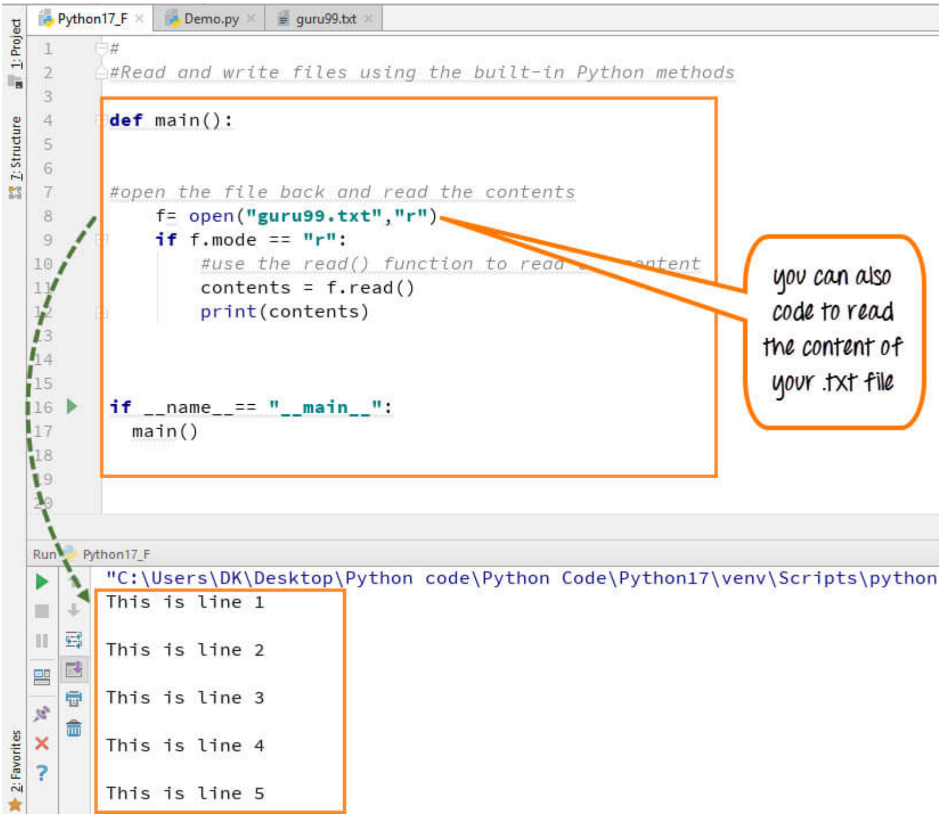
```
if f.mode == 'r':
```

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



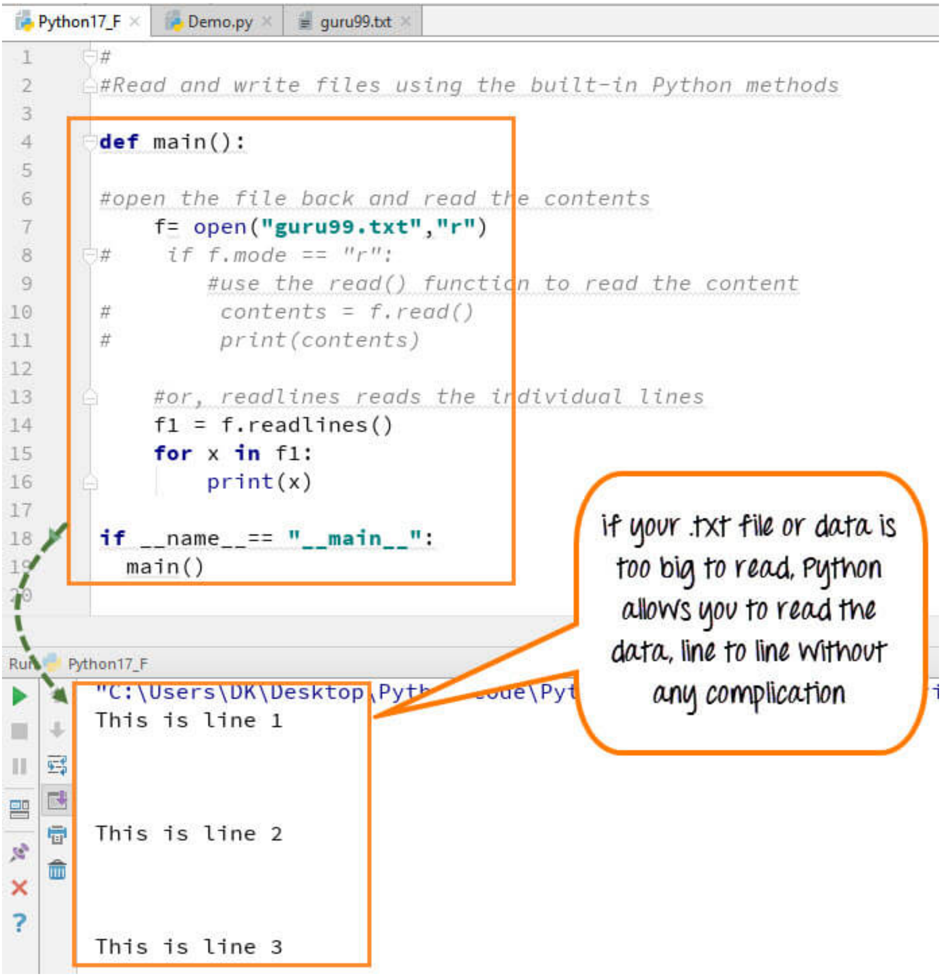
```
#
#Read and write files using the built-in Python methods

def main():


#open the file back and read the contents
    f= open("guru99.txt","r")
    if f.mode == "r":
        #use the read() function to read the content
        contents = f.read()
        print(contents)



if __name__== "__main__":
  main()
```

you can also code to read the content of your .txt file

```
"C:\Users\DK\Desktop\Python code\Python Code\Python17\venv\Scripts\python
This is line 1

This is line 2

This is line 3

This is line 4

This is line 5
```

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. This code will segregate your data in easy to ready mode



```python
#
#Read and write files using the built-in Python methods

def main():

#open the file back and read the contents
    f= open("guru99.txt","r")
#     if f.mode == "r":
        #use the read() function to read the content
#         contents = f.read()
#         print(contents)

    #or, readlines reads the individual lines
    f1 = f.readlines()
    for x in f1:
        print(x)

if __name__== "__main__":
  main()
```

```
"C:\Users\DK\Desktop\Pyth          ode\Pyt
This is line 1


This is line 2


This is line 3
```

if your .txt file or data is too big to read, Python allows you to read the data, line to line without any complication

When you run the code (**f1=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

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)

Here is the complete code

Python 2 Example

```
def main():
     f= open("guru99.txt","w+")
     #f=open("guru99.txt","a+")
     for i in range(10):
         f.write("This is line %d\r\n" % (i+1))
     f.close()
     #Open the file back and read the contents
     #f=open("guru99.txt", "r")
     #   if f.mode == 'r':
     #     contents =f.read()
     #     print contents
     #or, readlines reads the individual line into a list
     #fl =f.readlines()
     #for x in fl:
     #print x
if __name__== "__main__":
  main()
```

Python 3 Example

```
def main():
    f= open("guru99.txt","w+")
    #f=open("guru99.txt","a+")
    for i in range(10):
         f.write("This is line %d\r
\n" % (i+1))
    f.close()
    #Open the file back and read th
e contents
    #f=open("guru99.txt", "r")
    #if f.mode == 'r':
    #   contents =f.read()
    #    print (contents)
    #or, readlines reads the indivi
dual line into a list
    #fl =f.readlines()
    #for x in fl:
    #print(x)
if __name__== "__main__":
  main()
```

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.