File Handling in Python

File handling is an integral part of programming. File handling in Python is simplified with built-in methods, which include creating, opening, and closing files.

While files are open, Python additionally allows performing various file operations, such as reading, writing, and appending information.

Opening file in Python

The **open()** Python method is the primary file handling function. The basic syntax is:

```
object_name = open('file_name', 'mode')
```

The **open()** function takes two elementary parameters for file handling:

- The ***file_name*** includes the file extension and assumes the file is in the current working directory. If the file location is elsewhere, provide the absolute or relative path.
- The **mode** is an optional parameter that defines the file opening method. The table below outlines the different possible options:

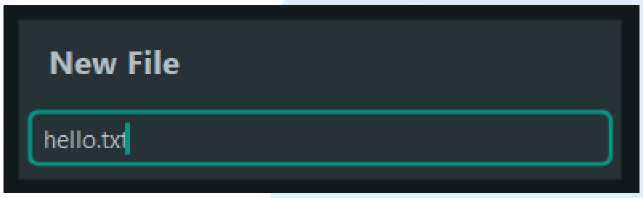
Mode	Description
‘r’	Reads from a file and returns an error if the file does not exist **(default)**.
‘w’	Writes to a file and creates the file if it does not exist or overwrites an existing file.
‘x’	Exclusive creation that fails if the file already exists.
‘a’	Appends to a file and creates the file if it does not exist or overwrites an existing file.
‘b’	Binary mode. Use this mode for non-textual files, such as images.
‘t’	Text mode. Use only for textual files (default).
‘+’	Activates read and write methods.

Read Mode

The read mode in Python opens an existing file for reading, positioning the pointer at the file's start.

Reading whole file with FOR loop

Create a new file named **hello.txt** in the same folder.



```
hello.txt
```

Write some dummy text into **hello.txt.**

```
Good morning
This is my dummy text
Hello I am the best developer
```

Now, we will be reading this file within our python program. To do so, write down the following code.

```python
# Opening file in r mode
f = open('hello.txt', 'r')

# Printing whole file line by line
for i in f:
    print(i)
```

Output

```
Good morning

This is my dummy text

Hello I am the best developer
```

Reading whole file and saving it in a variable using read() method

```python
# Opening file in r mode
f = open('hello.txt', 'r')

text = f.read()
print(text)
```

Output

```
Good morning
This is my dummy text
Hello I am the best developer
```

Storing each lines into a list using readlines() method

This will store each line at every position in the list

```python
# Opening file in r mode
f = open('hello.txt', 'r')

text = f.readlines()
print(text)
```

Output

```
['Good morning\n', 'This is my dummy text\n', 'Hello I am the best developer']
```

Reading single line using readline() method

```python
# Opening file in r mode
f = open('hello.txt', 'r')

line1 = f.readline()
print(line1)
line2 = f.readline()
print(line2)
```

Output

```
Good morning

This is my dummy text

```

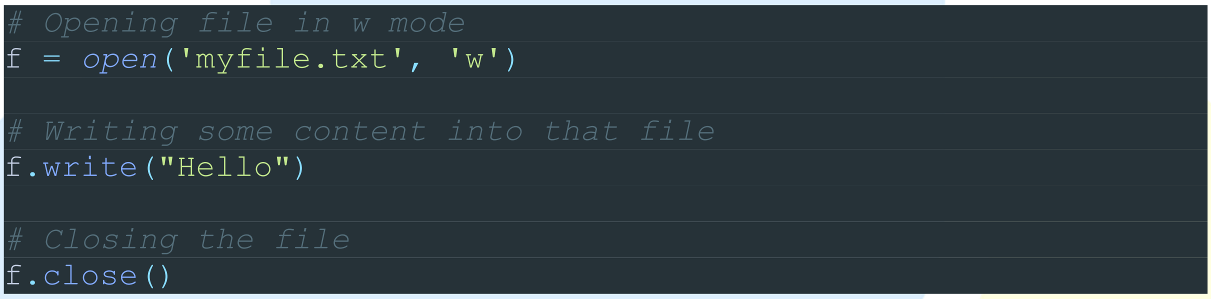
Write Mode

Write mode creates a file for writing content and places the pointer at the start. If the file exists, write truncates (clears) any existing information.

Write mode deletes existing content immediately. Check if a file exists before overwriting information by accident.

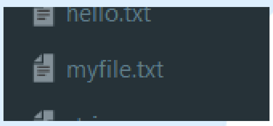
Writing some content into the file.

We will write some content into the file using our python code.



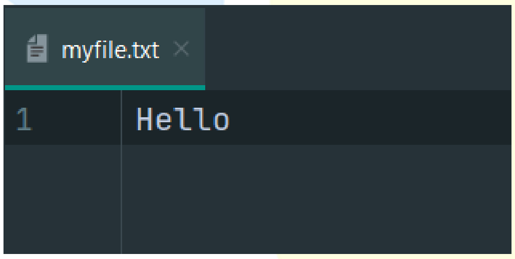
```python
# Opening file in w mode
f = open('myfile.txt', 'w')

# Writing some content into that file
f.write("Hello")

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

We know ***myfile.txt*** does not exists in our folder. As soon as we run our python code, we can see ***myfile.txt*** gets created.



```
hello.txt
myfile.txt
```

Content in myfile.txt



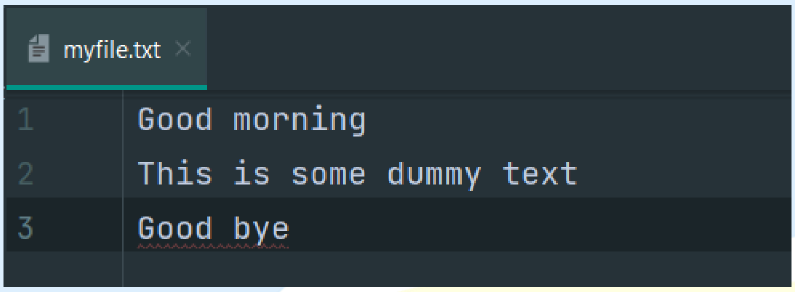
```
myfile.txt
Hello
```

Let us run our python code and write something else into the same file. See the code below.

```python
# Opening file in w mode
f = open('myfile.txt', 'w')

# Writing some content into that file
f.write("Good morning\nThis is some dummy text\nGood bye")

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

As soon as we run our python code, the content in ***myfile.txt*** will first be deleted and overwritten by the new content. This is because we opened file in **w (write)** mode.

Content of ***myfile.txt***:



```
Good morning
This is some dummy text
Good bye
```

Append Mode

Append mode adds information to an existing file, placing the pointer at the end. If a file does not exist, append mode creates the file.

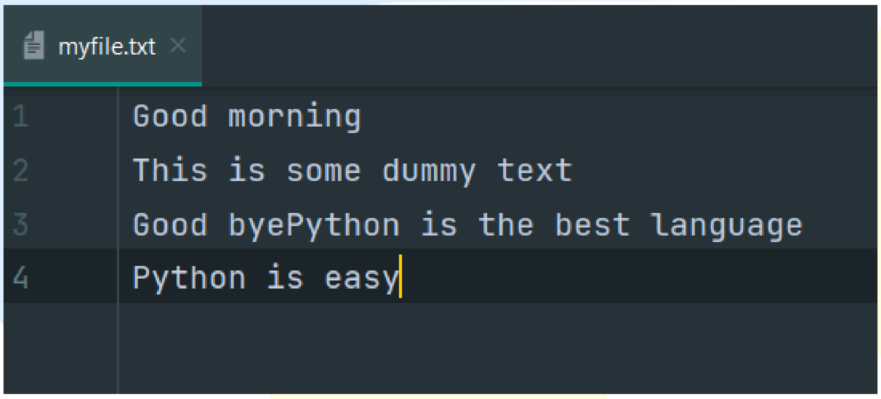
Note: The key difference between write and append modes is that append does not clear a file's contents.

Let us use the same file we did in write mode.

Now we are opening ***myfile.txt*** in **a(append)** mode.

```
# Opening file in a mode
f = open('myfile.txt', 'a')

# Writing some content into that file
f.write("Python is the best language\nPython is easy")

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

Output



```
Good morning
This is some dummy text
Good byePython is the best language
Python is easy
```

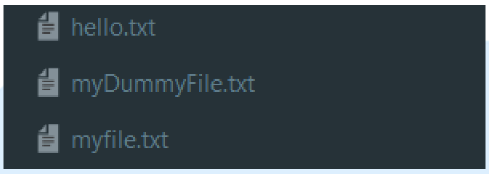
As we can see, when writing something in **append** mode, the content of file persists and is not overwritten which was the case in **write** mode.

Create a file using (x) Mode

In some case, where you just want to create a file and do nothing, **x** mode comes handy.

```python
# Opening file in X mode
# This will just create a new file
f = open('myDummyFile.txt', 'x')

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

After running this code, when we see in our directory, we have a ***myDummyFile.txt*** created.



hello.txt
myDummyFile.txt
myfile.txt

Deleting a file

Removing files in Python requires establishing communication with the operating system. Import the **os** library and delete a file with the following:

```python
import os

os.remove("myDummyFile.txt")
```

Note: If file does not exist and we try to delete it, it will throw an error.

Practice Examples (File Handling)

1. Write a Python program to read an entire text file.
2. Write a Python program to read a file line by line and store it into a list.
3. Write a python program to find the longest words.
4. Write a Python program to count the number of lines in a text file.
5. Write a Python program to count the frequency of words in a file.
6. Write a Python program to copy the contents of a file to another file.
7. Write a Python program to read a random line from a file.

For solutions, check the end of the book.