

File Handling in Python

Tushar B. Kute,
<http://tusharkute.com>

What is a file ?

- File is a named location on disk to store related information. It is used to permanently store data in a non-volatile memory (e.g. hard disk).
- Since, random access memory (RAM) is volatile which loses its data when computer is turned off, we use files for future use of the data.
- When we want to read from or write to a file we need to open it first. When we are done, it needs to be closed, so that resources that are tied with the file are freed.
- Hence, in Python, a file operation takes place in the following order.
 - Open a file
 - Read or write (perform operation)
 - Close the file

Opening a file

- Python has a built-in function `open()` to open a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.

```
>>> f = open("test.txt")
```

- We can specify the mode while opening a file.
- In mode, we specify whether we want to read 'r', write 'w' or append 'a' to the file.
- We also specify if we want to open the file in text mode or binary mode.
- The default is reading in text mode.

Python file modes

Mode	Description
'r'	Open a file for reading. (default)
'w'	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)

Examples

- `f = open("test.txt")`
equivalent to 'r' or 'rt'
- `f = open("test.txt", 'w')`
write in text mode
- `f = open("img.bmp", 'r+b')`
read and write in binary mode
- `f = open("test.txt", mode = 'r', encoding = 'utf-8')`
Specified with encoding

Reading functions

- **read (n)**
 - Read at most n characters from the file. Reads till end of file if it is negative or None.
- **readable ()**
 - Returns True if the file stream can be read from.
- **readline (n=-1)**
 - Read and return one line from the file. Reads in at most n bytes if specified.
- **readlines (n=-1)**
 - Read and return a list of lines from the file. Reads in at most n bytes/characters if specified.

read()

```
>>> f = open('test.txt')
>>> print f.read()
Hello All,
This is Tushar Kute,
MITU Skillologies, Pune
tushar@tusharkute.com
```

readline()

```
>>> f = open('test.txt')
>>> print f.readline()
Hello All,

>>> print f.readline()
This is Tushar Kute,

>>> print f.readline()
MITU Skillologies, Pune
```


readlines()

```
>>> f = open('test.txt')
>>> data = f.readlines()
>>> for line in data:
...     print line
...
Hello All,

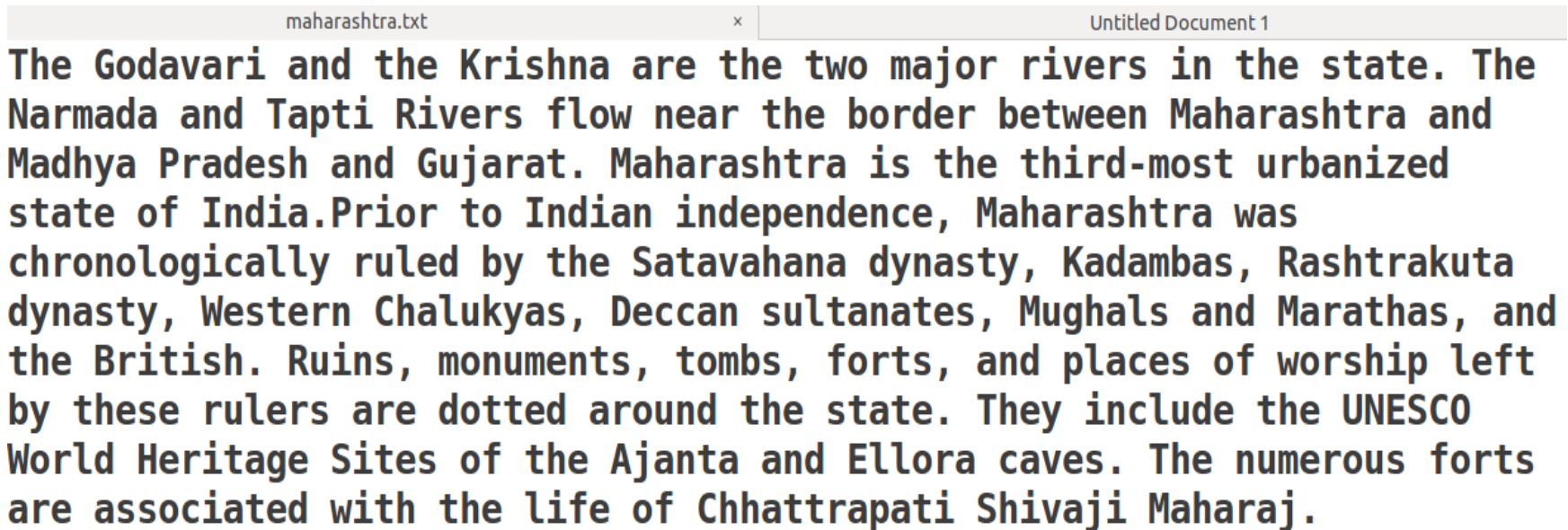
This is Tushar Kute,

MITU Skillologies, Pune

tushar@tusharkute.com
```

Example:

- Read a given file and count total number of 'the' in the given file.



maharashtra.txt x Untitled Document 1

The Godavari and the Krishna are the two major rivers in the state. The Narmada and Tapti Rivers flow near the border between Maharashtra and Madhya Pradesh and Gujarat. Maharashtra is the third-most urbanized state of India. Prior to Indian independence, Maharashtra was chronologically ruled by the Satavahana dynasty, Kadambas, Rashtrakuta dynasty, Western Chalukyas, Deccan sultanates, Mughals and Marathas, and the British. Ruins, monuments, tombs, forts, and places of worship left by these rulers are dotted around the state. They include the UNESCO World Heritage Sites of the Ajanta and Ellora caves. The numerous forts are associated with the life of Chhatrapati Shivaji Maharaj.

Solution

```
f = open('maharashtra.txt')  
data = f.read()  
words = data.split()  
print words.count('the')
```

Problem:

- Given with three text files, count the total of all the numerical elements from these files.

- file1

23
45
65
18
23
92
69
26
74
12

- file2

23 87
45 10 52 84 55
65
1889 12 10
23
92 64
69 12 66
26 44
74
12 90 42 44 73

- file3

23 87
45 qs 52 84 55
65
1889 12 10
23 abc knh
92 64
69 12 x 66
26 44
74
12 90 42 xy 73

Problem:

- Given the following file, count average of the marks find the name of topper student.

data.txt

```
1 amit 56.77
2 ajit 67.55
3 anita 75.44
4 anil 63.44
5 ajinkya 65.55
```

Closing a file

- When we are done with operations to the file, we need to properly close the file.
- Closing a file will free up the resources that were tied with the file and is done using Python `close()` method.
- Python has a garbage collector to clean up unreferenced objects but, we must not rely on it to close the file.

```
f = open("test.txt")  
# perform file operations  
f.close()
```

File using 'with' statement

- The with statement ensures that the file is closed when the block inside with is exited.
- We don't need to explicitly call the close() method. It is done internally.

```
with open("test.txt") as f:  
    print f.read()
```

Writing the file

- In order to write into a file in Python, we need to open it in write 'w', append 'a' or exclusive creation 'x' mode.
- We need to be careful with the 'w' mode as it will overwrite into the file if it already exists. All previous data are erased.
- Writing a string or sequence of bytes (for binary files) is done using write() method. This method returns the number of characters written to the file.

Example:

```
with open("test.txt", 'w') as f:  
    f.write("my first file\n")  
    f.write("This file\n\n")  
    f.write("contains three lines\n")
```

File writing methods

- **writable()**
 - Returns True if the file stream can be written to.
- **write(s)**
 - Write string *s* to the file and return the number of characters written.
- **writelines(lines)**
 - Write a list of lines to the file.

Sample File copy operation

```
f = open('test.txt')  
w = open('newfile.txt', 'w')  
data = f.read()  
w.write(data)  
w.close()  
f.close()
```

Example:

- Read the given file and copy only the numbers from it into another file. So, perform the number filtering operation. Do not change the lines.

```
23 87
45 qs 52 84 55
65
1889 12 10
23 abc knh
92 64
69 12 x 66
26 44
74
12 90 42 xy 73
```

Program:

```
d1 = open("numbers.txt")
d2 = open("number2.txt", "w")

s = d1.readlines()

for val in s:
    val = val.split()
    for x in val:
        if x.isdigit():
            d2.write(x+" ")
    d2.write('\n')

d1.close()
d2.close()
```

File positions

- We can change our current file cursor (position) using the seek() method.
- Similarly, the tell() method returns our current position (in number of bytes).

```
>>> f = open('test.txt')
```

```
>>> f.readline()
```

```
'Hello All,\n'
```

```
>>> print f.tell()
```

```
11
```

```
>>> f.readline()
```

```
'This is Tushar Kute,\n'
```

```
>>> print f.tell()
```

```
32
```

Seeking

```
>>> f = open('test.txt')
>>> f.readline()
'Hello All,\n'
>>> f.readline()
'This is Tushar Kute,\n'
>>> f.seek(0)
>>> f.readline()
'Hello All,\n'
>>> f.seek(11)
>>> f.readline()
'This is Tushar Kute,\n'
```

Problem Statements:

- Copy the contents of a file and write it in another file by swapping case of every character.
- Read the data of student like roll number, name and marks, store in in a file by appending it.

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License



@mitu_skillologies



/mITuSkillologies



@mitu_group



/company/mitu-
skillologies

Web Resources

<http://mitu.co.in>

<http://tusharkute.com>

contact@mitu.co.in

tushar@tusharkute.com