

Python Programming

by Narendra Allam

Copyright 2019

Chapter 7

File IO

Topics Covering

- Creating file
- File reading
- File writing
- File modes
- Line by line file reading
- Writing multiple lines
- seek()
- tell()
- os.getcwd()
- os.mkdir()
- os.chdir()
- os.remove()
- os.rmdir()
- Use Case - CSV file reading and writing

Files is generally anything that generally is saved on permanent storage devices with a name. Content of file can be simple text, binary data(image, audio, video) etc. Only text files are dicussed in this chapter.

open(): This is the function in python to open a file.

Syntax:

```
file_handle = open(<filename>, <mode>)
```

Open function -- opens a file and returns a file object, through which we perform all operations on a file.

Note: In the above statement, we are trying to open abc.txt, if abc.txt is not exisiting, we get "IOError" in Python 2.x and we get "FileNotFoundError" in Python 3.x

Modes:

Text Modes

r or rt - read mode, if file not exists throws IOError
w or wt - write mode, if file not exists creates new one
a or at - append mode is write mode but starts writing, from the end of the file

r+ or rt+ - read write
w+ or wt+ - write read
a+ or at+ - append read

Binary Modes

rb - Binary read
wb - Binary write
ab - append
rb+ - read and write in binary
wb+ - read and write in binary
ab+ - read and append in binary

File creation and writing

In [1]:

```
1 f = open('abc.txt', 'w')
2
3 f.write("Once upon a time in India, there was a king called Tippu.")
4 f.close()
```

f is the file object, which holds a buffer in RAM, which will be synced to hard disk later. `_close()` function ensures the sync between content written to a file and memory buffer. It flushes all the content to a file on hard disk.

Let's check the content of the file, we can run OS commands from jupyter notebook. just prefix with the command with '!'.

In [2]:

```
1 f = open('abc.txt')
2 txt = f.read()
3 print(txt)
4 f.close()
```

Once upon a time in India, there was a king called Tippu.

Open file with context manager:

In [3]:

```
1 with open('abc.txt') as f:
2     txt = f.read()
3     print(txt)
```

Once upon a time in India, there was a king called Tippu.

In [4]:

```
1 !cat abc.txt # On windows run : !type abc.txt
```

Once upon a time in India, there was a king called Tippu.

In [5]:

```
1 !ls # On windows run : !dir
```

```
abc.txt
book1.json
book.json
books.json
book.xml
browser.png
'Chapter 10 Numpy.ipynb'
'Chapter 11 Pandas & Matplotlib.ipynb'
'Chapter 12 Object Orientation.ipynb'
'Chapter 13.1 MySQLDBSetupWalkthrough.ipynb'
'Chapter 13 DBConnection - MySQL.ipynb'
'Chapter 14 ExceptionHandling.ipynb'
'Chapter 17 Regular Expressions.ipynb'
'Chapter 1 Introduction.ipynb'
'Chapter 20 Email - SMTP.ipynb'
'Chapter 2 Strings.ipynb'
'Chapter 3 Control Structures.ipynb'
'Chapter 4 Data Structures.ipynb'
'Chapter 5 Functions.ipynb'
'Chapter 6 Modules.docx'
'Chapter 7 File IO.ipynb'
'Chapter 8 Comprehensions, Lambdas and Functional Programming.ipynb'
'Chapter 9 Serialization.ipynb'
Comprehensions.ipynb
'Control Structures.ipynb'
country.xml
CountWays.ipynb
create.png
cs.ipynb
data.csv
data.txt
double_decor.ipynb
drawing10.png
drawing11.png
drawing12.png
drawing13.png
drawing14.png
drawing1.png
drawing2.png
drawing3.png
drawing4.png
drawing5.png
drawing6.png
drawing7.png
drawing8.png
drawing9.png
employ_data.csv
employee.pickle
employee.xml
employ.pickle
file-8.png
FirstNotebook.ipynb
'Icon'$'\r'
large_data.csv
LibraryManagementSystem.ipynb
Lists.ipynb
mac.png
```

```
merge.ipynb
'MongoDB .ipynb'
nextBiggest.ipynb
notebook.tex
Operators.ipynb
output_1_1.png
password.py
password.pyc
Practice1.ipynb
__pycache__
'Python Complete Brochure.docx'
'Python Modules.docx'
random_data.csv
random_data.numbers
random_data.xlsx
'replace In.ipynb'
shot1.png
shot2.1.png
shot2.png
shot3.png
shot4.png
shot5.png
shot6.png
Shot8.png
start.png
tmp_shot1.png
'Trainers Presentation.pptx'
Untitled1.ipynb
Untitled.ipynb
windows.png
```

In [6]:

```
1 !pwd
```

```
/hdd/notebooks/VASU/updated Python Notebooks-final
```

Do you want to check, which folder your are in? This `getcwd()` function gives you current working directory.

```
import os
print os.getcwd()
```

In [7]:

```
1 import os
2 print(os.getcwd())
```

```
/hdd/notebooks/VASU/updated Python Notebooks-final
```

Reading an existing file

`read()` function reads entire file content as a string.

In [8]:

```
1 f = open('abc.txt', 'r')
2 s = f.read()
3 print(s)
4 f.close()
```

Once upon a time in India, there was a king called Tippu.

Reading n characters, read(n)

In [9]:

```
1 f = open('abc.txt', 'r')
2 s = f.read(10)
3 print(s)
4 f.close()
```

Once upon

Writing Multiline text

In [10]:

```
1 f = open('abc.txt', 'w')
2 f.write("""Once upon a time in India, there was a king called Tippu.
3 Tippu was so tall and handsome and brave. He was looking for a brave and beautiful
4 bride and sent the message to all of his citizens.""")
5 f.close()
```

Check the file content

In [11]:

```
1 !cat abc.txt # !type abc.txt for windows
```

Once upon a time in India, there was a king called Tippu.

Tippu was so tall and handsome and brave. He was looking for a brave and beautiful

bride and sent the message to all of his citizens.

Reading text into a list of strings

readlines() function returns all lines in the file as a list of strings

In [12]:

```
1 f = open('abc.txt', 'r')
2 l = f.readlines()
3 print(l)
4 f.close()
```

```
['Once upon a time in India, there was a king called Tippu.\n', 'Tippu\nwas so tall and handsome and brave. He was looking for a brave and beautiful\n', 'bride and sent the message to all of his citizens.']
```

Line by line file reading

In [13]:

```
1 f = open('abc.txt', 'r')
2
3 for line in f:
4     print(line, end='')
5
6 f.close()
```

Once upon a time in India, there was a king called Tippu.
Tippu was so tall and handsome and brave. He was looking for a brave and beautiful
bride and sent the message to all of his citizens.

Writing multiple lines

In [14]:

```
1 f = open('abc.txt', 'w')
2 l = ['Once upon a time in India, there was a king called Tippu.\n',
3     'Tippu was so tall and handsome and brave. He was looking for a brave and beautiful\n',
4     'bride and sent the message to all of his citizens.\n',
5     'He was waiting for years...']
6 f.writelines(l)
7 f.close()
```

In [15]:

```
1 !cat abc.txt
```

Once upon a time in India, there was a king called Tippu.
Tippu was so tall and handsome and brave. He was looking for a brave and beautiful
bride and sent the message to all of his citizens.
He was waiting for years...

A new write operation in existing file with some text discards the existing text, so we have to use append mode to add the text at the end.

In [16]:

```
1 f = open('abc.txt', 'w')
2 f.write("Apple is sweet !!!")
3 f.close()
```

In [17]:

```
1 !cat abc.txt
```

Apple is sweet !!!

In [18]:

```
1 f = open('abc.txt', 'a')
2 f.write("Orange is sour!\n")
3 f.close()
```

In [19]:

```
1 !cat abc.txt
```

Apple is sweet !!!Orange is sour!

After writing into a file, file pointer moves to the end, so reading after writing won't read anything. Check the below example.

In [20]:

```
1 f = open('abc.txt', 'a+')
2
3 f.write("Sky is blue! \n Milk is White.")
4
5 s = f.read()
6
7 print ("File content = ", s)
8
9 f.close()
```

File content =

seek(): Moving the file pointer in file

Syntax:

```
f.seek(<Offset>, <Whence>)
```

1. f.seek(n, io.SEEK_SET) reads from nth character from start of the file

In [21]:

```
1 !cat abc.txt
```

Apple is sweet !!!Orange is sour!
Sky is blue!
Milk is White.

In [22]:

```
1 import io
2 f = open('abc.txt', 'a+')
3 f.seek(20, io.SEEK_SET)
4 s = f.readline()
5 print(s)
6 f.close()
```

ange is sour!

io.SEEK_SET, io.SEEK_CUR and io.SEEK_END are the reference points from which offset needs to be considered in a file.

```
f.seek(5, io.SEEK_SET) # moves file pointer to 5th character from file start
f.seek(-10, io.SEEK_CUR) # moves file pointer to 10th character from current
```


position

f.seek(-10, io.SEEK_END) # moves file pointer to 50th character from end of the file

f.tell(): this function returns current offset position from start of the file

In [23]:

```
1 import io
2 f = open('abc.txt', 'rb')
3 f.seek(5, io.SEEK_SET)
4 print (f.tell())
5
6 print (f.readline())
7 print (f.tell())
8
9 f.seek(-3, io.SEEK_CUR)
10 print (f.readline())
11 print (f.tell())
12
13
14 f.seek(-6, io.SEEK_END)
15 print (f.readline())
16 print (f.tell())
17
18 f.seek(0, io.SEEK_SET)
19 print (f.readline())
20 print (f.tell())
21
22 f.seek(0, io.SEEK_END)
23 print (f.readline())
24 print (f.tell())
```

```
5
b' is sweet !!!Orange is sour!\n'
34
b'r!\n'
34
b'White.'
63
b'Apple is sweet !!!Orange is sour!\n'
34
b''
63
```

In [24]:

```
1 !cat data.txt
```

here are plenty of artificial intelligence (AI) trends emerging from China these days, but arguably the most intriguing are those that show just how close humans are to meeting their match at the hands of machines. This week, we saw 100Credit, a big data platform company, has launched a low-cost robot called "Little 100Credit" that reportedly collects bad debts over the phone 90% as often as its human counterparts. Yes, a robot calls up customers who owe money and helps people repay their debts almost as well as a human.

Program: Read text from a text file, find the word with most number of occurrences

In [25]:

```
1 from collections import Counter
2
3 f = open('data.txt')
4 s = f.read()
5 f.close()
6
7 letters = [char for char in s if char.isalnum() or char == ' ']
8 words = ''.join(letters).split()
9 print('Most frequently occurred word:', Counter(words).most_common(1))
```

Most frequently occurred word: [('a', 4)]

Storing data as CSV file

In [26]:

```
1 import datetime
2
3 l = [(10001, datetime.date(1953, 9, 2), 'Georgi', 'Facello', 'M', datetime.date(1985, 12, 1)),
4 (10002, datetime.date(1964, 6, 2), 'Bezalel', 'Simmel', 'F', datetime.date(1985, 12, 1)),
5 (10003, datetime.date(1959, 12, 3), 'Parto', 'Bamford', 'M', datetime.date(1986, 12, 1)),
6 (10004, datetime.date(1954, 5, 1), 'Chirstian', 'Koblick', 'M', datetime.date(1986, 12, 1)),
7 (10005, datetime.date(1955, 1, 21), 'Kyoichi', 'Maliniak', 'M', datetime.date(1986, 12, 1)),
8 (10006, datetime.date(1953, 4, 20), 'Anneke', 'Preusig', 'F', datetime.date(1986, 12, 1)),
9 (10007, datetime.date(1957, 5, 23), 'Tzvetan', 'Zielinski', 'F', datetime.date(1986, 12, 1)),
10 (10008, datetime.date(1958, 2, 19), 'Saniya', 'Kalloufi', 'M', datetime.date(1986, 12, 1)),
11 (10009, datetime.date(1952, 4, 19), 'Sumant', 'Peac', 'F', datetime.date(1985, 12, 1)),
12 (10010, datetime.date(1963, 6, 1), 'Duangkaew', 'Piveteau', 'F', datetime.date(1986, 12, 1))]
```

In [27]:

```
1 ','.join(['Apple', 'Orange', 'Banana', 'Peach'])
```

Out[27]:

'Apple,Orange,Banana,Peach'

In [28]:

```
1 f = open('data.csv', 'w')
2 for rec in l:
3     s = ','.join([str(rec[0]), rec[1].strftime('%Y-%m-%d'),
4                   rec[2], rec[3], rec[4], rec[5].strftime('%Y-%m-%d')])
5     s += '\n'
6     f.write(s)
7 f.close()
```

In [29]:

```
1 !cat data.csv
```

```
10001,1953-09-02,Georgi,Facello,M,1986-06-26
10002,1964-06-02,Bezalel,Simmel,F,1985-11-21
10003,1959-12-03,Parto,Bamford,M,1986-08-28
10004,1954-05-01,Chirstian,Koblick,M,1986-12-01
10005,1955-01-21,Kyoichi,Maliniak,M,1989-09-12
10006,1953-04-20,Anneke,Preusig,F,1989-06-02
10007,1957-05-23,Tzvetan,Zielinski,F,1989-02-10
10008,1958-02-19,Saniya,Kalloufi,M,1994-09-15
10009,1952-04-19,Sumant,Peac,F,1985-02-18
10010,1963-06-01,Duangkaew,Piveteau,F,1989-08-24
```

In [30]:

```
1 import datetime
2 f = open('data.csv')
3 for rec in f:
4     rec = rec.rstrip('\n')
5
6     l = rec.split(',')
7     l[0] = int(l[0])
8
9     year, month, day = [int(x) for x in l[1].split('-')]
10    l[1] = datetime.date(year, month, day)
11
12    year, month, day = [int(x) for x in l[5].split('-')]
13    l[5] = datetime.date(year, month, day)
14
15    print(l)
16 f.close()
```

```
[10001, datetime.date(1953, 9, 2), 'Georgi', 'Facello', 'M', datetime.
date(1986, 6, 26)]
[10002, datetime.date(1964, 6, 2), 'Bezalel', 'Simmel', 'F', datetime.
date(1985, 11, 21)]
[10003, datetime.date(1959, 12, 3), 'Parto', 'Bamford', 'M', datetime.
date(1986, 8, 28)]
[10004, datetime.date(1954, 5, 1), 'Chirstian', 'Koblick', 'M', dateti
me.date(1986, 12, 1)]
[10005, datetime.date(1955, 1, 21), 'Kyoichi', 'Maliniak', 'M', dateti
me.date(1989, 9, 12)]
[10006, datetime.date(1953, 4, 20), 'Anneke', 'Preusig', 'F', datetim
e.date(1989, 6, 2)]
[10007, datetime.date(1957, 5, 23), 'Tzvetan', 'Zielinski', 'F', datet
ime.date(1989, 2, 10)]
[10008, datetime.date(1958, 2, 19), 'Saniya', 'Kalloufi', 'M', datetim
e.date(1994, 9, 15)]
[10009, datetime.date(1952, 4, 19), 'Sumant', 'Peac', 'F', datetime.da
te(1985, 2, 18)]
[10010, datetime.date(1963, 6, 1), 'Duangkaew', 'Piveteau', 'F', datet
ime.date(1989, 8, 24)]
```

In [31]:

```
1 rec1 = '1234 John 23000 Male'
2 rec2 = '1235 Samantha 34000 Female'
3
4 f = open('data.csv', 'w')
5
6 l = rec1.split()
7 rec = ','.join(l) + '\n'
8 f.write(rec)
9
10 l = rec2.split()
11 rec = ','.join(l) + '\n'
12 f.write(rec)
13
14 f.close()
15
```

In [32]:

```
1 !cat data.csv
```

```
1234,John,23000,Male
1235,Samantha,34000,Female
```

Reading data from CSV file

In [33]:

```
1 f = open('data.csv')
2 for line in f:
3     l = line.rstrip('\n').split(',')
4     print(l)
5 f.close()
```

```
['1234', 'John', '23000', 'Male']
['1235', 'Samantha', '34000', 'Female']
```

With: Open a file with context manager

In [34]:

```
1 with open('abc.txt') as f:
2     print(f.read())
```

```
Apple is sweet !!!Orange is sour!
Sky is blue!
Milk is White.
```

Note: With closes file automatically, even an exception occurred.

Useful functions from os module

import os

`os.getcwd()`: Returns current working directory

`os.mkdir()`: Creates a directory

`os.chdir()`: Change directory

`os.remove()`: Removes a file

`os.rmdir()`: Removing directory

`os.listdir('.')`: Lists current directory

Interview questions

1. How do you read line by line from a file in python ?

```
with open('abc.txt') as f:  
    s = f.read()
```

2. What happens if 'abc.txt' doesn't exist?