Python Programming

by Narendra Allam

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 permenent 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>)
```

In [8]:

```
f = open('abc.txt', 'r')
```

```
----------------------------------------------------------------------
-----
FileNotFoundError                         Traceback (most recent call
 last)
<ipython-input-8-c7e2003e559e> in <module>()
----> 1 f = open('abc.txt', 'r')

FileNotFoundError: [Errno 2] No such file or directory: 'abc.txt'
```

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

Note: In the above statement abct.txt is not exisiting, so IOError

Modes:
Text Modes

```
r or rt - read mode, if file not exists throws IOError
w or wt - write mode, if file not exists creats 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 - Binray 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 [9]:

```
f = open('abc.txt', 'w')

f.write("Once upon a time in India, there was a king called Tippu.")
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 '!'.

Open file with context manager:

In [10]:

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

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

In [11]:

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

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

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

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

In [12]:

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

```
/Users/nikky/Google Drive/Latest Notebooks
```

Reading an existing file

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

In [13]:

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

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

Reading n charcters, read(n)

In [14]:

```
f = open('abc.txt', 'r')
s = f.read(5)
print(s)
f.close()
```

```
Once
```

Writing Multiline text

In [15]:

```
f = open('abc.txt', 'w')
f.write("""Once upon a time in India, there was a king called Tippu.
Tippu was soo tall and handsom and brave. He was looking for a brave and beatiful
bride and sent the messgae to all of his citizens.""")
f.close()
```

Check the file content

In [16]:

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

```
Once upon a time in India, there was a king called Tippu.
Tippu was soo tall and handsom and brave. He was looking for a brave a
nd beatiful
bride and sent the messgae to all of his citizens.
```

Reading text into a list of strings

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

In [17]:

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

```
['Once upon a time in India, there was a king called Tippu.\n', 'Tippu
was soo tall and handsom and brave. He was looking for a brave and bea
tiful\n', 'bride and sent the messgae to all of his citizens.']
```

Line by line file reading

In [18]:

```
f = open('abc.txt', 'r')

for line in f:
    print(line, end='')

f.close()
```

```
Once upon a time in India, there was a king called Tippu.
Tippu was soo tall and handsom and brave. He was looking for a brave a
nd beatiful
bride and sent the messgae to all of his citizens.
```

Writing multiple lines

In [19]:

```
f = open('abc.txt', 'w')
l = ['Once upon a time in India, there was a king called Tippu.\n',
 'Tippu was soo tall and handsom and brave. He was looking for a brave and beatiful\
 'bride and sent the messgae to all of his citizens.\n',
 'He was waiting for years...']
f.writelines(l)
f.close()
```

In [20]:

```
!cat abc.txt
```

```
Once upon a time in India, there was a king called Tippu.
Tippu was soo tall and handsom and brave. He was looking for a brave a
nd beatiful
bride and sent the messgae to all of his citizens.
He was waiting for years...
```

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

In [21]:

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

In [22]:

```
!cat abc.txt
```

```
Apple is sweet !!!
```

In [23]:

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

In [24]:

```
!cat abc.txt
```

```
Apple is sweet !!!Orange is sour!
```

After writing into a file, file pointer moves to the end, so read ing after writing wont read anything. Check the below example.

In [25]:

```
f = open('abc.txt', 'a+')

f.write("Sky is blue!\nMilk is White.")

s = f.read()

print ("File content = ", s)

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 charcater from start of the file

In [26]:

```
!cat abc.txt
```

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

In [27]:

```
import io
f = open('abc.txt', 'a+')
f.seek(20, io.SEEK_SET)
s = f.readline()
print(s)
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 [28]:

```
f = open('abc.txt')
f.seek(5, io.SEEK_SET)
print (f.tell())

print (f.readline())
print (f.tell())

f.seek(0, io.SEEK_CUR)
print (f.readline())
print (f.tell())

f.seek(0, io.SEEK_CUR)
print (f.readline())
print (f.tell())

f.seek(0, io.SEEK_END)
print (f.readline())
print (f.tell())
```

```
5
 is sweet !!!Orange is sour!

34
Sky is blue!

47
Milk is White.
61

61
```

In [29]:

```
!cat abc.txt
```

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

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

In [30]:

```
from collections import Counter

f = open('abc.txt')
s = f.read()
f.close()

letters = [char for char in s if char.isalnum() or char == ' ']
words =  ''.join(letters).split()
print ('Most frequently occured word:', Counter(words).most_common(1))
```

```
Most frequently occured word: [('is', 4)]
```

Storing data as CSV file

In [31]:

```
import datetime

l = [(10001, datetime.date(1953, 9, 2), 'Georgi', 'Facello', 'M', datetime.date(1986
(10002, datetime.date(1964, 6, 2), 'Bezalel', 'Simmel', 'F', datetime.date(1985, 11,
(10003, datetime.date(1959, 12, 3), 'Parto', 'Bamford', 'M', datetime.date(1986, 8,
(10004, datetime.date(1954, 5, 1), 'Chirstian', 'Koblick', 'M', datetime.date(1986,
(10005, datetime.date(1955, 1, 21), 'Kyoichi', 'Maliniak', 'M', datetime.date(1989,
(10006, datetime.date(1953, 4, 20), 'Anneke', 'Preusig', 'F', datetime.date(1989, 6,
(10007, datetime.date(1957, 5, 23), 'Tzvetan', 'Zielinski', 'F', datetime.date(1989,
(10008, datetime.date(1958, 2, 19), 'Saniya', 'Kalloufi', 'M', datetime.date(1994, 9
(10009, datetime.date(1952, 4, 19), 'Sumant', 'Peac', 'F', datetime.date(1985, 2, 18
(10010, datetime.date(1963, 6, 1), 'Duangkaew', 'Piveteau', 'F', datetime.date(1989,
(10011, datetime.date(1953, 11, 7), 'Mary', 'Sluis', 'F', datetime.date(1990, 1, 22)
(10012, datetime.date(1960, 10, 4), 'Patricio', 'Bridgland', 'M', datetime.date(1992
(10013, datetime.date(1963, 6, 7), 'Eberhardt', 'Terkki', 'M', datetime.date(1985, 1
(10014, datetime.date(1956, 2, 12), 'Berni', 'Genin', 'M', datetime.date(1987, 3, 11
(10015, datetime.date(1959, 8, 19), 'Guoxiang', 'Nooteboom', 'M', datetime.date(1987
(10016, datetime.date(1961, 5, 2), 'Kazuhito', 'Cappelletti', 'M', datetime.date(199
(10017, datetime.date(1958, 7, 6), 'Cristinel', 'Bouloucos', 'F', datetime.date(1993
(10018, datetime.date(1954, 6, 19), 'Kazuhide', 'Peha', 'F', datetime.date(1987, 4,
(10019, datetime.date(1953, 1, 23), 'Lillian', 'Haddadi', 'M', datetime.date(1999, 4
(10020, datetime.date(1952, 12, 24), 'Mayuko', 'Warwick', 'M', datetime.date(1991, 1
```

In [32]:

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

In [33]:

```
!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
10011,1953-11-07,Mary,Sluis,F,1990-01-22
10012,1960-10-04,Patricio,Bridgland,M,1992-12-18
10013,1963-06-07,Eberhardt,Terkki,M,1985-10-20
10014,1956-02-12,Berni,Genin,M,1987-03-11
10015,1959-08-19,Guoxiang,Nooteboom,M,1987-07-02
10016,1961-05-02,Kazuhito,Cappelletti,M,1995-01-27
10017,1958-07-06,Cristinel,Bouloucos,F,1993-08-03
10018,1954-06-19,Kazuhide,Peha,F,1987-04-03
10019,1953-01-23,Lillian,Haddadi,M,1999-04-30
10020,1952-12-24,Mayuko,Warwick,M,1991-01-26
```

In [34]:

```
f = open('data.csv')
for rec in f:
    rec = rec.rstrip('\n')
    l = rec.split(',')
    l[0] = int(l[0])
    print(l)
f.close()
```

```
[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']
[10011, '1953-11-07', 'Mary', 'Sluis', 'F', '1990-01-22']
[10012, '1960-10-04', 'Patricio', 'Bridgland', 'M', '1992-12-18']
[10013, '1963-06-07', 'Eberhardt', 'Terkki', 'M', '1985-10-20']
[10014, '1956-02-12', 'Berni', 'Genin', 'M', '1987-03-11']
[10015, '1959-08-19', 'Guoxiang', 'Nooteboom', 'M', '1987-07-02']
[10016, '1961-05-02', 'Kazuhito', 'Cappelletti', 'M', '1995-01-27']
[10017, '1958-07-06', 'Cristinel', 'Bouloucos', 'F', '1993-08-03']
[10018, '1954-06-19', 'Kazuhide', 'Peha', 'F', '1987-04-03']
[10019, '1953-01-23', 'Lillian', 'Haddadi', 'M', '1999-04-30']
[10020, '1952-12-24', 'Mayuko', 'Warwick', 'M', '1991-01-26']
```

In [35]:

```
rec1 = '1234 John 23000 Male'
rec2 = '1235 Samantha 34000 Female'

f = open('data.csv', 'w')

l = rec1.split()
rec = ','.join(l) + '\n'
f.write(rec)

l = rec2.split()
rec = ','.join(l) + '\n'
f.write(rec)

f.close()
```

In [36]:

```
!cat data.csv
```

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

Reading data from CSV file

In [37]:

```
f = open('data.csv')
for line in f:
    l = line.split(',')
    l[-1] = l[-1].rstrip('\n')
    print(l)
    print
f.close()
```

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

With: opening a file with context manager

In [38]:

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

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

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

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.tx') as f:
    s = f.read()
```

What happens if ‘abc.txt’ doesn’t exist?