

FILE HANDLING IN PYTHON

FILE INPUT/OUTPUT

- Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun.
- We use `open ()` function in Python to open a file in read or write mode. As explained above, `open ()` will return a file object. To return a file object we use `open ()` function along with two arguments, that accepts file name and the mode, whether to read or write.
- Syntax: **`open (filename, mode)`**
 - "r", for reading.
 - "w", for writing.
 - "a", for appending.
 - "r+", for both reading and writing

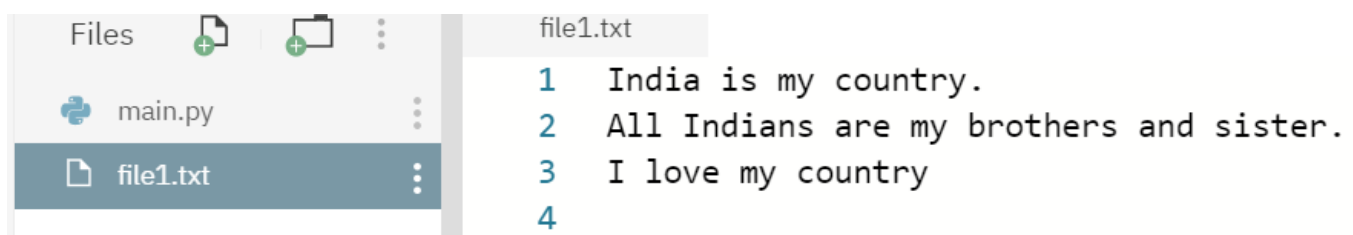
WORKING OF READ MODE

Approach-1

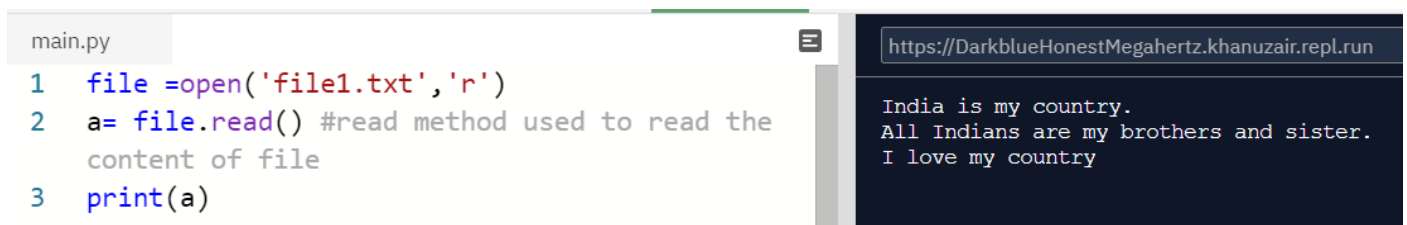
If you need to extract a string that contains all characters in the file then we can use

`file.open()`

- File given below is text file named as `file1.txt`



- Main.py is the file by which we can interact -



Approach-2

- Another way to read a file is to call a certain number of characters like in the following code the interpreter will read the first five characters of stored data and return it as a string:

```
1
2 file= open('file1.txt','r')
3 print(file.read(5))
4
```

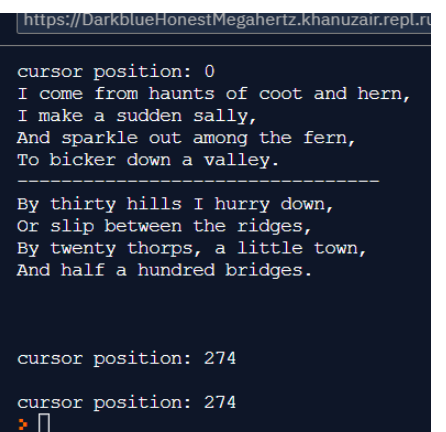


```
India
>
```

-The close () command terminates all the resources in use and frees the system of this particular program

```
am.py
file=open('file1.txt','r')
print(f'cursor position: {file.tell()}')
print(file.read())
print(f'cursor position: {file.tell()}')
print(file.read())
print(f'cursor position: {file.tell()}')

'''file.tell() gives the location of cursor.
here in example as we can see that as the file was open the
posotion was zero and when we read the file the cursor position
become 274'''
```



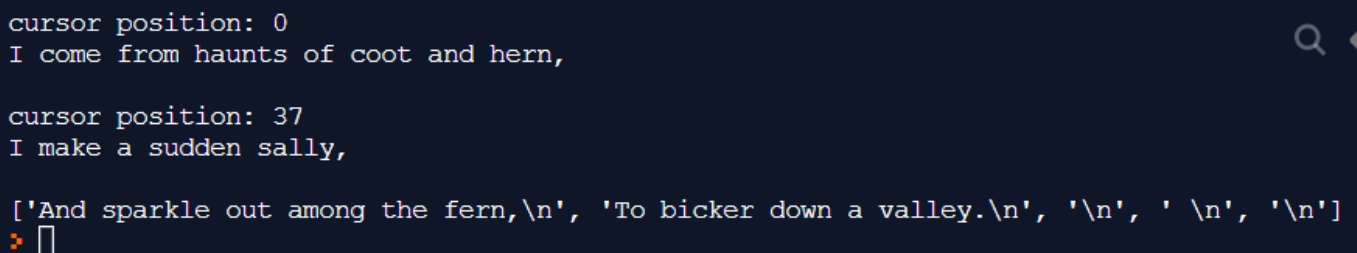
```
https://DarkblueHonestMegahertz.khanuzair.repl.re
cursor position: 0
I come from haunts of coot and hern,
I make a sudden sally,
And sparkle out among the fern,
To bicker down a valley.
-----
By thirty hills I hurry down,
Or slip between the ridges,
By twenty thorps, a little town,
And half a hundred bridges.

cursor position: 274
cursor position: 274
>
```

- **file.tell()** method used for knowing the location of cursor.
- **file.seek()** method is used to change the location of cursor.
- **File.readline ()** method read only one line at a time.
- **File.readlines ()** method give all lines in file as list.

file handling example.py > ...

```
1 file=open('file1.txt','r')
2 print(f'cursor position: {file.tell()}')
3 print(file.readline()) # to print single line at a time
4 print(f'cursor position: {file.tell()}')
5 print(file.readline())
6 print(file.readlines()) #it will give all lines of file in list
```



```
cursor position: 0
I come from haunts of coot and hern,

cursor position: 37
I make a sudden sally,

['And sparkle out among the fern,\n', 'To bicker down a valley.\n', '\n', ' \n', '\n']
>
```

- Python uses an idea of cursor to read the file. That's why we need to set cursor to 0 using seek method in order to read it again.
- It is better to close the file after doing all operations using-

```
file.close()
```

'with' STATEMENT

- there is no need to call file.close() when using with statement. The with statement itself ensures proper acquisition and release of resources.
- Thus, with statement helps avoiding bugs and leaks by ensuring that a resource is properly released when the code using the resource is completely executed.

```
with open('file1.txt') as file:
    print(file.readline())
```

WRITING INTO THE FILE

READ PLUS MODE (READ AND WRITE MODE)

with statement example file handling.py > ...

```
1 with open('file1.txt',mode='r+') as file: #r+ for writing into the file
2     data= file.write('Galgotias university')
3     print(data)
4
```

APPEND MODE:

h statement example file handling.py > ...

```
1 with open('file1.txt',mode='a') as file: #r+ for writing into the file
2     data= file.write('Galgotias university')
3     print(data) #this is giving the character which are written into the file
4
5 '''r+ overwrites the data which is already present
6 in the file as every time we do the cursor resets itself
7 in order to avoid this thing we use append mode'''
8
```

WRITE MODE:

ain.py

```
with open('happy.txt',mode='w') as file:
    data=file.write('My name is anthony')
    print(data) #shows character written into the file
```

```
'''GOOD THING ABOUT WRITE MODE IS THAT IT CAN CREATE
A FILE IN WHICH WE WANT TO WRITE EVEN IF IT IS NOT
PRESENT BEFORE'''
```

CODE EXERCISE

Code to make an offline translate, which takes file and provide you the translation in the asked language-

test.txt

```
1 The Brook by Lord Alfered Tennyson
2
3 I come from haunts of coot and hern
4 | I make a sudden sally,
5 And sparkle out among the fern,
6 | To bicker down a valley.
7
8 By thirty hills I hurry down,
9 | Or slip between the ridges,
10 By twenty thorps, a little town,
11 | And half a hundred bridges.
12
13 |
```

main.py

```
1 from translate import Translator
2 translate_to = Translator(to_lang='ur')
3 with open('test.txt',mode='r') as my_file:
4     data= my_file.read()
5     translation= translate_to.translate(data)
6     print(translation)
7
```

<https://CarefulSurefootedSystems.khanuzair.repl.run>

لیس کن اچا سیم سوہ ای آس سوڈا کے نہ روا ٹوک سیم کورب ہعیرڈ کے نسینیٹ ڈیرفل ای ڈرال
کے سوہڑاہپ سیت۔ وہ اڑگھج سیم یداو یسک ہکات ، اہٹا کمچ نایمرد کے نیرف روا ، سوانب ی
وہج کی ، سوروج سیب ، سوہ اتاج لسہپ نایمرد کے سورانک ای ، سوہ اترک یدلج سیم ہعیرڈ
-سے سولپ وس اہدآ روا ، رہش اس اٹ
➤ □

CODE EXERCISE:

You have text file in which data is given as state and capital

You have to segregate the data in other file as {state}'s capital is {capital}

```
segregate data in other file.py  test1.txt X
test1.txt
1  UP,Lucknow
2  MP,Bhopal
3  Rajasthan,Jaipur
4  Punjab,Chandhigarh
5  Assam,Dispur
6  Nagaland,Kohima
7  HP,Shimla
8

segregate data in other file.py X  test1.txt
segregate data in other file.py > ...
1  with open('test1.txt',mode='r') as read_file:
2  with open('output1.txt',mode='a') as write_file:
3  for line in read_file.readlines():
4      state, capital = line.split(',')
5      write_file.write(f'{state}\s capital is {capital}')
```

```
segregate data in other file.py  output1.txt X
output1.txt
1  UP's capital is Lucknow
2  MP's capital is Bhopal
3  Rajasthan's capital is Jaipur
4  Punjab's capital is Chandhigarh
5  Assam's capital is Dispur
6  Nagaland's capital is Kohima
7  HP's capital is Shimla
8  |
```