



EXPLORER

OPEN EDITORS

User Settings C:\Users\p...
percipio40_writing_data...

PYTHON

Automate-Boring-Stuff
my_code
Percipio_Python3-Course
01_Start
02_Data-Sequence Types
03_Collections-Mapping-...
04_Modules-Functions
05_Classes
06_Working-with-Files
 loremipsum - Shortcut.I...
 percipio36_docstrings.py
 percipio37_code_comm...
 percipio38_documentat...
 percipio39_reading_text...
 percipio40_writing_data...
 percipio41_writing_larg...
 percipio42_reading_bin...
 percipio43_writing_bina...
 percipio44_exercise_cre...
 reading_text_files - Sho...
 reading_text_files.txt
 writing_data - Shortcut...
 writing_data.txt
07_Comprehensions
08_Iterables-and-Generat...
09_Exceptions
Python Projects_2014
CMD_Python_Set-Path.txt
Python_Basics.txt
Python_Clear-Window-Com...
python_exercises_00.py
python_exercises_01.py
Python_Tutorial_Running-Sc..
Python_Tutorials.md

User Settings

percipio40_writing_data.py X

```
1 '''
2 percipio40_writing_data.py
3 Percipio video: Working with Files; Writing Data
4
5 Demonstrates writing, reading, finding the position location, and seeking to the beginning & end
6 of a text file.
7 This code will write to an existing text file, or create the file and write to it if not in the
8 directory.
9
10 The write method (='w') allows writing text, or strings, to a file
11 '''
12 nl = '\n'
13
14 from io import UnsupportedOperation
15 ''' [io module](https://docs.python.org/3/library/io.html?highlight=io#module-io)
16 Using the open() function uses the standard library IO module implicitly.
17 Since an error may occur (known as an 'UnsupportedOperation'), from the IO module, the
18 UnsupportedOperation class is imported to use the exception in the code.
19
20 When creating a file with the open() function, there are a few important things;
21 * file path is correct
22 * acceptable permission to write to a file in that location
23 * Must specify the correct mode (w = write)
24
25 I think the 3 'with open' statements should be writing to the text file, but they are only
26 writing to the teminal. The 'lines' tuple statement below is writing directly into the
27 text file.
28 '''
29 print('Default Windows 10 encoding "cp1252".', nl) # code page 1252
30 with open('writing_data.txt', mode='w') as out_file: # out_file object is created,...
31     out_file.write('This is the first line of text written from Python code. \n') # ...method
32     .write is called to write the first line of text to the file.
33
34 with open('writing_data.txt', mode='a') as append_file:
35     append_file.write('This is an appended line of text written from Python code. \n')
36
37 with open('writing_data.txt', mode='r') as in_file: # creates an 'in_file' object
38     # do not need to specify 'r' (mode='r' [read-only]) as that is the default.
```





EXPLORER	
OPEN EDITORS	
User Settings	C:\Users\p...
percipio40_writing_data...	
PYTHON	
Automate-Boring-Stuff	
my_code	
Percipio_Python3-Course	
01_Start	
02_Data-Sequence Types	
03_Collections-Mapping-...	
04_Modules-Functions	
05_Classes	
06_Working-with-Files	
loremipsum - Shortcut.I...	
percipio36_docstrings.py	
percipio37_code_comm...	
percipio38_documentat...	
percipio39_reading_text..	
percipio40_writing_data...	
percipio41_writing_larg...	
percipio42_reading_bin...	
percipio43_writing_bina...	
percipio44_exercise_cre...	
reading_text_files - Sho...	
reading_text_files.txt	
writing_data - Shortcut...	
writing_data.txt	
07_Comprehensions	
08_Iterables-and-Generat...	
09_Exceptions	
Python Projects_2014	
CMD_Python_Set-Path.txt	
Python_Basics.txt	
Python_Clear-Window-Com...	
python_exercises_00.py	
python_exercises_01.py	
Python_Tutorial_Running-Sc...	
Python_Tutorials.md	

User Settings		percipio40_writing_data.py
33		# do not need to specify 'r' (mode='r' [read-only]) as that is the default.
34		print('in_file.read() = \n', in_file.read()) # print out what's inside the 'in_file' object by calling its read() method. (It's printing out the .write and .append)
35		
36		# Video 02:50 to 03:56 extended to 05:00_ Discusses 'with' context manager & how the 'try/finally' code block does the same thing. IT'S UNCLEAR WHAT THEIR TALKING ABOUT
37		
38		lines = (# a tuple list
39		'Line 1\n', 'Line 2\n',
40		'Line 3\n', 'Line 4\n',
41		'This is the only section actually writing to the text file.'
42) #
43		
44		''' Using a file without the "with" context '''
45		# 'with' context needs to be in an indented block
46		try:
47		out_file = open('writing_data.txt', mode='w', buffering=1, encoding='us') # try to open the file in write mode, buffering enabled, & a specified encoding
48		# buffering is normally not enabled using a (minus) -1 value
49		out_file.writelines(lines) # writelines method takes a sequence, here the tuple 'lines' with the 4 Line elements
50		out_file.flush() # with buffering enabled, the file will not be written to until the flush method on the file object is called, or the file is closed ('out_file.close()')
51		finally:
52		out_file.close() # file is closed
53		'''
54		The try/finally code block essentially do the same thing as the 'with' context manager used above
55		'''
56		
57		# Video 05:00
58		try: # code to read the text file, this block handles an exception
59		# BOF = beginning of file
60		# EOF = end of file
61		in_file = open('writing_data.txt', encoding='us') # trys to open the file specifying the encoding
62		
63		print(in_file.tell(), end=' <- Current opening position of file before it is read\nShould always open at position zero\n\n')





EXPLORER	
OPEN EDITORS	
{} User Settings C:\Users\p...	64
percipio40_writing_data...	
PYTHON	
▶ Automate-Boring-Stuff	
▶ my_code	
▶ Percipio_Python3-Course	
▶ 01_Start	
▶ 02_Data-Sequence Types	66
▶ 03_Collections-Mapping...	67
▶ 04_Modules-Functions	68
▶ 05_Classes	69
▶ 06_Working-with-Files	70
≡ loremipsum - Shortcut.l...	71
percipio36_docstrings.py	72
percipio37_code_comm...	73
percipio38_documentat...	74
percipio39_reading_text...	
percipio40_writing_data...	75
percipio41_writing_larg...	76
percipio42_reading_bin...	77
percipio43_writing_bina...	
percipio44_exercise_cre...	78
≡ reading_text_files - Sho...	79
≡ reading_text_files.txt	80
≡ writing_data - Shortcut...	
≡ writing_data.txt	81
▶ 07_Comprehensions	82
▶ 08_Iterables-and-Generat...	83
▶ 09_Exceptions	84
▶ Python Projects_2014	85
≡ CMD_Python_Set-Path.txt	
≡ Python_Basics.txt	86
≡ Python_Clear-Window-Com...	87
percipio40_writing_data...	88
python_exercises_00.py	
python_exercises_01.py	
Python_Tutorial_Running-Sc...	
Python_Tutorials.md	

{} User Settings

percipio40_writing_data.py x

```
always open at position zero\n\n')
print('printing "in_file.read(7)" equals:', in_file.read(7)) # in_file read method with '7'
is called, so 7 characters will be read.
print('NOTE: The first 7 characters are read of the "lines" tuple, which are "Line 1\\n".
However \\n is read as 1 character so the actual 1st 7 characters are "Line 1\\n". The
computer still reads 8 bytes of the file however & therefore logs 8 bytes read, shown in
the call result of the next .tell() function', nl, nl, 'NOTE: as you read along in the
file, the position automatically moves forward.', nl)

print(in_file.tell(), end=' <- File position after first read of 7 characters/8 bytes\n')
print('printing "in_file.read(7)" equals:', in_file.read(7), nl) #

print(in_file.tell(), end=' <- Current position in file before second read\n')
# Video 07:33 goes into seek
'''
Can not seek realtive
What can seek is a position, or offset, from either from the beginning, specified by zero,
or the end, specified by two (WHY TWO?)
'''
in_file.seek(0, 0) # seek(offset, from_what) from_what=0 is BOF
''' seek(0,0) (is same as 'seek(0)') will automatically move the position to the beginning
of the file '''
print(nl)

print(in_file.tell(), end=' <- BOF after seek(0,0), which tells the position is 0, which is
at the beginning of file')
in_file.seek(0, 2) # from_what=2 is EOF
''' '2' specifies the end of the file (WHY?) '''
print(nl)

print(in_file.tell(), end=' <- EOF after seek(0,2), which tells the position is 32, which is
at the end of the file (WHY IS "2" THE END OF THE FILE?)\n\n')

print('printing "in_file.read()" equals:', in_file.read())
print('The read() function reads from the current position to the end of the file. Since
the current position is at the end of the file, this read() results in a blank or
"nothing to read" because it can not read beyond the end of the file.\n') # print from
position to end
```




EXPLORER

1

OPEN EDITORS 1 UNSAVED

User Settings C:\Users\p...
percipio40_writing_data...

PYTHON

Automate-Boring-Stuff

my_code

Percipio_Python3-Course

01_Start

02_Data-Sequence Types

03_Collections-Mapping-...

04_Modules-Functions

05_Classes

06_Working-with-Files

loremipsum - Shortcut.I...

percipio36_docstrings.py

percipio37_code_comm...

percipio38_documentat...

percipio39_reading_text...

percipio40_writing_data...

percipio41_writing_larg...

percipio42_reading_bin...

percipio43_writing_bina...

percipio44_exercise_cre...

reading_text_files - Sho...

reading_text_files.txt

writing_data - Shortcut...

writing_data.txt

07_Comprehensions

08_Iterables-and-Generat...

09_Exceptions

Python Projects_2014

CMD_Python_Set-Path.txt

Python_Basics.txt

Python_Clear-Window-Com...

python_exercises_00.py

python_exercises_01.py

Python_Tutorial_Running-Sc...

Python_Tutorials.md

User Settings

percipio40_writing_data.py

position to end

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

```
print(in_file.tell(), end=' <- EOF after read()\n')
''' This next line causes the error because Python 3 does not support relative position with
    text files. (Video 09:04)'''
in_file.seek(7, 1) # from_what=1 is relative to position
''' This next tell() function is not run because the error (except) block was called '''
print(in_file.tell(), end=' <- Current position in file after relative seek\n')
print(nl)

# Video 05:20 to 05:55
except UnsupportedOperation: # Python 3 does not support move (or seek) in a negative direction
    or a positive direction from your current position. So if code trys to seek from a relative
    position, this exception is raised. UNCLEAR!
    print('Text files do not support relative seeks')
finally:
    in_file.close() # closes the text file
'''
RESULTS_in Terminal:
Default Windows 10 encoding "cp1252".

in_file.read() =
This is the first line of text written from Python code.
This is an appended line of text written from Python code.

0 <- Current opening position of file before it is read
Should always open at position zero

printing "in_file.read(7)" equals: Line 1

NOTE: The first 7 characters are read of the "lines" tuple, which are "Line 1\n". However \n is
      read as 1 character so the actual 1st 7 characters are "Line 1\n". The computer still reads
      8 bytes of the file however & therefore logs 8 bytes read, shown in the call result of the
      next .tell() function

NOTE: as you read along in the file, the position automatically moves forward.

8 <- File position after first read of 7 characters/8 bytes
printing "in_file.read(7)" equals: Line 2
```




EXPLORER	
1	OPEN EDITORS 1 UNSAVED
{}	User Settings C:\Users\p...
•	percipio40_writing_data...
PYTHON	
▶	Automate-Boring-Stuff
▶	my_code
▶	Percipio_Python3-Course
▶	01_Start
▶	02_Data-Sequence Types
▶	03_Collections-Mapping...
▶	04_Modules-Functions
▶	05_Classes
▶	06_Working-with-Files
≡	loremipsum - Shortcut.I...
•	percipio36_docstrings.py
•	percipio37_code_comm...
•	percipio38_documentat...
•	percipio39_reading_text...
•	percipio40_writing_data...
•	percipio41_writing_larg...
•	percipio42_reading_bin...
•	percipio43_writing_bina...
•	percipio44_exercise_cre...
≡	reading_text_files - Sho...
≡	reading_text_files.txt
≡	writing_data - Shortcut...
≡	writing_data.txt

User Settings	
percipio40_writing_data.py	
120	printing "in_file.read(7)" equals: Line 2
121	
122	
123	16 <- Current position in file before second read
124	
125	
126	0 <- BOF after seek(0,0), which tells the position is 0, which is at the beginning of file
127	
128	91 <- EOF after seek(0,2), which tells the position is 32, which is at the end of the file (WHY IS "2" THE END OF THE FILE?)
129	
130	printing "in_file.read()" equals:
131	The read() function reads from the current position to the end of the file. Since the current position is at the end of the file, this read() results in a blank or "nothing to read" because it can not read beyond the end of the file.
132	
133	91 <- EOF after read()
134	Text files do not support relative seeks
135	
136	RESULTS_in 'writing_data.txt' File:
137	Line 1
138	Line 2
139	Line 3
140	Line 4
141	This is the only section actually writing to the text file.
142	'''

