



Hyperiondev

Working with External Data Sources – Recap on File I/O

Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- ❑ No question is daft or silly - **ask them!**
- ❑ There are Q/A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- ❑ You can also submit questions here:
<http://hyperiondev.com/sbc4-se-questions>
- ❑ For all non-academic questions, please submit a query:
www.hyperiondev.com/support
- ❑ Report a safeguarding incident:
<http://hyperiondev.com/safeguardreporting>
- ❑ We would love your feedback on lectures:
<https://hyperiondev.wufoo.com/forms/zsgv4m40ui4i0g/>

Objectives

1. Recap on how to work with external data sources in Python
 - a. Writing data to text files.
 - b. Reading data from text files.
2. Learn how to read and write data without reopening the file
 - a. Using the 'r+', 'w+', 'a+' access modes
 - b. How to manipulate input data and output it

Let's quickly recap File I/O

Opening a File

- ★ Create a variable to store the name of the text file.
- ★ Create a variable to store the **file object** created by the **open()** function.
- ★ Specify the access mode eg. 'r' to allow user to read the data in the file object.

```
file_name = 'input.txt'  
  
file = open(file_name, 'r')
```

Reading Files

- ★ To properly read the `file object`, we will need to use one of the read methods: `.read()`, `.readline()`, `.readlines()`

```
# Use one of the read methods to read the contents
lines = file.read() # reads and stores all data as a string type
# Or
lines = file.readline() # reads and stores only the first line
# Or
lines = file.readlines() # reads and stores all data in a list

# Call the print function on the 'lines' variable to display contents
print(lines)
# Remember to close the file
file.close()
```

Writing to Files

- ★ Create a file object like before or use a with/as block
- ★ Set the access mode to 'w'.
- ★ If the file does not exist, it will be created.
- ★ Use the .write() method to write contents to a file.

```
with open('output.txt', 'w') as file:
```

```
    file.write("Mankind knew, that they cannot change society.\n")  
    file.write("So instead of reflecting on themselves. \n")  
    file.write("They blamed the beast")
```

```
file.close()
```

Things to Note

- ★ Remember that when the file is reopened and new data is written to the file, the previous data is then overwritten.
- ★ You can preserve the previous data by using the append (a) access mode. This will simply append the new data to the end of the file, instead of overwriting.

```
# Using the 'a' access mode will prevent data to be over written
# Open the file again
file_name = 'output.txt' # This is the original text file

file = open(file_name, 'a')

file.write("This is the new text")

file.close()
```


Reading and writing

- ★ There are ways to write/append contents to a file and read it without reopening the file.
- ★ You would need to use different access modes depending on the purpose:
 - 'w+' : To write and then read.
 - 'r+' : To read and then write.
 - 'a+' : To append and then read.

Example

```
with open('output.txt', 'w+') as file:

    file.write("Mankind new, that they cannot change society.\n")
    file.write("So instead of reflecting on themselves. \n")
    file.write("They blamed the beast")

    # Important: return to the top of the file before reading
    file.seek(0)

    line = file.read()
    print(line)
```

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any



Hyperiondev

**Thank you
for joining us**