Exercise 1.4: File Handling in Python

Learning Goals

- Use files to store and retrieve data in Python

Reflection Questions

1. Why is file storage important when you're using Python? What would happen if you didn't store local files?

File storage is important because it is a necessary step to preserve data after a program or application is closed. In Python, if data is not written to a storage file it will all be lost after the script finishes running. In order to preserve data, it must be written to a binary file and then that file can be read the next time a script is run to preserve the data from one use to another.

2. In this Exercise you learned about the pickling process with the **pickle.dump()** method. What are pickles? In which situations would you choose to use pickles and why?

Pickles are a package of streamed bytes that have been converted from complex data. Since a binary file stores complex data in a way that can only be read by a machine, pickles are needed to translate the information. Pickles allow you to retrieve complex data and display it in a readable fashion with *.load()*, they also allow you to write data into a binary file by using *.dump()*.

3. In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?

You can determine what directory you are working in by using the *os.getcwd()* command. If you want to change your current working directory you can do so with the *os.chdir()* command to navigate to the directory of your choice.

4. Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?

If you are encountering a section of code that might throw an error, you can write it in a try/except/else block. Putting the code you want to run in the try section, if it is successful the code will simply continue on to the else section and subsequently the rest of the script. However, if there is an error it will be caught by the except section informing the user of the error before moving along to the else section.

5. You're now more than halfway through Achievement 1! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? Feel free to use these notes to guide your next mentor call.

So far I think Achievement 1 is going well. I am proud of the scripts written for this exercise, as well as the overall knowledge and understanding I have gained regarding python so far. I feel much more comfortable loops, and try/except/else as well. I would definitely like more practice with error handling(especially handling specific errors), structuring data, and using pickles within scripts.