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 when we're using Python (and other programming languages) because it allows you to read data from files and write data to files, thus preserving data even after the program has finished execution. If we didn't store local files all data would be erased and become unavailable when we run the application again.

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 convert complex data into a packaged stream of bytes, known as "pickle", and then write this into a binary file. I would choose to use pickles for storing (more) complex data structures, such as dictionaries, because in this situation it's difficult to retain the structure of data in the form of regular text.

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?**
   In Python, we use the **os.getcwd()** command to find out which file directory we're currently working on. The **os.chdir()** command lets the Python interpreter change its current working directory. To use it follow this syntax:

   os.chdir(“<path to desired folder>”)

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?**
   I would use the **try-except** block and the finally block. The **try** block of code (first block) is where you would expect the error to occur. If no error occurs the rest of the code is executed; if an error occurs the (output) of **except** block of code may be used to notify the user of the error and provide guidance on fixing the error, or do other tasks. I would mark the end of my program using a **finally** block because this block will still run, regardless of whether an error occurred or not, and even if a return statement occurs before it (normally Python ignores any statements following return).

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.**
   Everything is going well. I have been able to apply concepts explained in the exercises (and that I found on other resources).
   I’m having difficulty understanding what is asked in some steps. Sometimes it’s hard to visualize what will be the final result/output of the task as we go through the steps.
   I need more practice with file handling and lambda functions. I also need to explore more the concepts of mutable/ immutable variables and functions with arguments vs. functions with keyword arguments.