Alexis Harris

February 26, 2025

Course: IT FDN 110 A

Assignment 05

Introduction

In this assignment, I learned about new file types and methods for coding. I have taken many beginner coding courses throughout undergrad and graduated school in R, C++ and MATLAB; and, I never learned about JSON files and try exception statements. After reviewing the Module 05 text, I had a better understand of the usefulness of these methods. Using the module text, the module labs, and videos, I was able to utilize the assignment starter python file to create a program that saves user input into a JSON file and utilizes the try exception statements. I also refreshed on pushing to GitHub for code collaboration and submitting this assignment.

GitHub Module 05 repository link: https://github.com/ahar9/IntroToProg-Python-Mod05

Module 05 Review

I reviewed JavaScript Object Notation (JSON) files. This is a new type of data file I have never worked with. JSON seems useful for developers for accessing data using keys. Often, I work with csv data for analysis and don't yet anticipate needing JSON files. However, I noticed in the review that JSON files are used with APIs, and I use utilize APIs to download csv data from hospital surveillance databases. So, I am curious to learn more about the connection between JSON files and APIs.

The try exception methos was new material for me. The try exception method must make program packages more user friendly. I appreciate the example in Demo Video 3 using the try exception programming for reading and writing a JSON file. In past assignments I would have to create data files separate from running the Python program, but with this example I can have the program either read the existing file to the program or create it if not present.

Assignment05.py

For the next section, I compared the acceptance criteria for assignment 05 to Assignment05-starter.py

File Name/Header/Constants/Variables

- Assignment05-starter.py needs to be Assignment05.py for submission
- FILE_NAME needs to change from "Enrollments.csv" to "Enrollments.json"
- Student_data changed to dictionary rather than list

Input/Output

- Change menu 1 student_data variable from list of list to list of dictionaries. With that use json function to dump data. Import json function. Referencing Lab02
- Change menu 2 print statement to output list based off dictionary keys (first name, last name, course) rather than list indexes. Referencing Lab02.

Processing

- Starter file reads enrollment file as csv/text file. Changed this data read to reading a JSON file. Checked to see Enrollments JSON file in folder has data, and fortunately it does! Assuming that reading in JSON file will require try except code, I referenced Lab03 for reading in JSON files.
- For menu choice 3 requirements, we need to upload data from the students list to the enrollments JSON file. The starter file uploads to a cvs dataset, so I referenced Lab02 for reading writing to a JSON file. Then I updated the print statement to reference keys rather than indexes in the list.
- Menu 4 exits the program, which the starter file executes.

Error Handling

- To address error handling when file is read into the list of dictionary rows, I referenced Lab03. The error handling addressed FileNotFoundError and other non-specific errors and closes the opened file.
- To address error handling when the user enters a first name and last name, I referenced Lab03 menu choice 2 to raise ValueError for names containing numbers. I edited the starter files menu choice 1, and added the technical error messages.
- Referenced Lab03 menu option 3 for error handling when the dictionary rows are written in the file in the starter menu option 03

Troubleshooting

After my edits, I noticed that the enrollment JSON file keys and my code keys were different. So, I edited the code to add equivalent Keys to the JSON file. I also ran into errors for printing the dictionary data in a string (Figure 1). After troubleshooting, I figured that the f-string function in the case for printing dictionary keys needs to use single quotations around the whole string, and the dictionary keys use double quotations.

```
C:\Users\Alexis\AppData\Roaming\rminiconda\nanopipeline_python\python.exe "C:\Users\Alexis\Documents\Python
  File "C:\Users\Alexis\Documents\Python course 2025\_Module05\Assignment\Assignment05.py", line 106
    print(f"Student {student["FirstName"]} {student["LastName"]} is enrolled in {student["Course"]}")
                            ^
SyntaxError: f-string: unmatched '['

Process finished with exit code 1
```

Figure 1 - Errors for printing student dictionary data

After fixing these issues, I tested my program in PyCharm and the command window based off the Test section in the acceptance criteria. Fortunately, the program ran as I was hoping for it. I also tested one of the try exceptions in the program. I entered a number for the student first name variable. The program prompted a message to dare not do this again (Figure 2).

```
What would you like to do: 1
What is the student's first name? c3po
The first name should not contain numbers.
-- Technical Error Message --
Inappropriate argument value (of correct type).
The first name should not contain numbers.

---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course.
    2. Show current data.
    3. Save data to a file.
    4. Exit the program.
----------------------------------------

What would you like to do:
```

Figure 2 - Try exception test for student first name

Conclusion

After reviewing Module 05 text, videos, and labs, I learned about JSON files and the functionality of keys and try exception coding. Using these materials, I completed a python program that takes user data and saves that data to a JSON file and alerts the user to anticipated errors. I refamiliarized myself with GitHub and uploaded the documentation.