

Name: Parker Henry

Date: 11/04/24

Course: IT FDN 110: Foundations Of Programming: Python

Github Link: <https://github.com/parkerduolingo/IntroToProg-Python-Mod05>

Assignment 05: Writing a Program Using Dictionaries, Files, and Exception Handling

Introduction

This version of the course registration program I've been writing for my assignments introduces JSON files to read and write data. This change offers a more structured way to handle data compared to the CSV format I used in previous assignments. I also added *try* and *except* code to take account of possible errors that could occur. This makes the code more robust and less prone to stopping suddenly. Overall, this assignment repeats much of the functionality of Assignment 04 but in a way that seems safer and firmer.

Step 1: Import the JSON Module

To handle JSON files, I first imported the ``json`` module at the top of the script. This is crucial for using JSON data in Python.

Step 2: Define Data Constants and Variables

The constants and variables were adjusted slightly compared to Assignment 04 to account for JSON. The ``FILE_NAME`` is set to "Enrollments.json" as opposed to "Enrollments.csv", indicating that the data will be stored in a JSON format. Additionally, "student_data" is now a dictionary as opposed to a list. JSON files seem to like to read dictionaries more so than lists.

Step 3: Read Existing Data from JSON File

At the beginning of the program, I attempted to open and read data from the JSON file. This is where I put my first *try-except-finally* in case the file didn't exist or if the data couldn't be read. I put specific instructions for what to print out if the `FileNotFoundError` exception occurred, and I put general instructions for what to print if any other exceptions occurred. This ensures that even if the file didn't exist when the program started, a new file would be opened so that the data could be saved.

Step 4: Modify User Data Input

In the option for registering a student (menu choice 1), the program now creates a `'student_data'` dictionary (not a list) for each student entry. This step ensures that records fit neatly into the JSON structure. I also added another 'try-except-finally' block here in case the user inputted non-alphabetical characters for the first name or last name. One thing that was a little tricky though was that in the demo video on try/except (<https://www.youtube.com/watch?v=O46YoSo477Y>) (external link), the instruction was to continue if the user put in an incorrect input. In this case the data wouldn't save and the user would have to reinput the information. That didn't seem immediately obvious to me, but after running through the program a few times, I understood it.

Step 5: Display Current Data

To display the data in a meaningful format, I iterated through the *students* list of dictionaries and printed each entry. The use of dictionaries makes this print statement a little easier to format because I could use each of the keys (*FirstName*, *LastName*, and *CourseName*) in the code which felt more intuitive than the last assignment where I used the indices for a list.

Step 6: Save Data Back to the JSON File

In the menu choice for saving data (option 3), I used the *json.dump()* method which writes the list of student dictionaries to the file *Enrollments.json*. This is wrapped in another try-except-finally block just in case there are any errors in saving to the file. I made sure to put the code to close the file in the finally block for both try-except-finally structures in this program so that regardless of what happens the file gets closed.

Conclusion

Overall, this enhancement to the program not only provided the capability to use JSON files for data storage but also improved the efficiency of the program. This version takes into account several errors that could occur and accounts for them. Finally, this assignment allowed me to dip my toes into Github, which is a common space for programmers to collaborate on projects.