Spencer Neiffer

ITFDN110A

Module 05

2/25/25

https://github.com/SBNEIFFER/Python110_2025.git

Dictionaries, JSON, Exception Handling

Summary

This document outlines the steps I took in creating a Python program that works with dictionaries, JSON files, and exception handling. The program initially opens a JSON file and extracts its contents into a list variable. A user can then enter information about a student, which is stored in respective string variables, parsed into a dictionary variable, then appended to the existing list. The list can then be written back to the JSON file with the additional information entered by the user.

Constants and Variables

The first step I took in creating this Python program is ensuring that my constants are declared correctly. I made sure that my first constant was called ***MENU***, and it's a string that contains the options available to the user within the program. The second and final constant in my program is a string called ***FILE_NAME***, and it contains the name of the file that the program will write to. This variable was initially set to ***Enrollments.csv***, but since my program will be using JSON, I changed its variable to ***Enrollments.json***.

After ensuring my constants were declared correctly, I looked at my variables. The first variable is called ***student_first_name***, and it's an empty string that holds a student's first name entered by the user. The second variable is ***student_last_name***, which is also an empty string that holds a student's last name entered by the user. My next variable is called ***course_name***, and it's an empty string that stores the course name for a student entered by the user. After this, I had a string variable called csv_data that stores the formatted data to be written to a CSV file. Since my program works with JSON files and not CSV files, I removed this variable. Next, I had a variable called ***student_data*** that was initially set to an empty list. My program works with dictionaries instead of lists, so I changed this variable from a list to a dictionary using curly brackets. Below this, I have an empty list variable called ***students***. This was correct so I made no changes. Next, I had an object variable called ***file***, which was correctly set to ***None***. The final variable was an empty string called ***menu_choice***, and it stores the menu number chosen by the user that navigates through the program.

Reading From JSON File

Since my program reads and writes data to and from a JSON file, I first need to tell it to use JSON. To do this, I added the ***import json*** command at the very beginning of the program. After this, I added a code block that tells the program to open ***FILE_NAME***, which is set to ***Enrollments.json***, in read mode. I then added code to load the contents of the file ***Enrollments.json*** into the ***students*** list variable. Finally, I added code to close the file.

Menu Option 1 – Capturing User Input

To keep the program functioning after a task was completed by the user, I added a ***while True*** loop and began writing my code. The first task that my program performs is displaying the ***MENU*** constant, which presents the user with options for navigating through the program. After this, an input function is used that displays the message: "***What would you like to do***", and captures the user's input, storing it within the variable ***menu_choice***. Next, I added an ***if*** statement that checks the ***menu_option*** variable's value, and performs a task based on it. The first part of the ***if*** statement checks to see if ***menu_option*** is equal to '1'. If it is, the program initiates three input functions that capture user input. The first function asks the user to enter a student's first name, then stores the input with the ***student_first_name*** variable. The second input function asks the user to enter a student's last name and stores the captured input within the ***student_last_name*** variable. Finally, the third function asks the user to enter a course name that the student is registering for. This captured input is stored in the ***course_name*** variable. After this input is captured, I pass those three variable values to the ***student_data*** dictionary, specifying the keys as ***FirstName***, ***LastName***, and ***CourseName***. Once the ***student_data*** dictionary contained the student's first and last name, as well as the course name, I appended it to the ***students*** list, essentially creating a list of dictionaries. So, every subsequent student and course captured from the user is added to the ***students*** list. Finally, I have the program display a message to the user informing them that the student information they just entered was registered, by using the variables stored within ***student_first_name***, ***student_last_name***, and ***course_name***.

Menu Option 2 – Printing List Data

The second part of the ***if*** statement performs a task if the value in menu_choice is equal to ‘2’. If this is true, the program first prints 50 dashes to create a visual barrier between the rest of the output. Then, a ***for*** loop prints the first name, last name, and course name for each dictionary contained in the ***students*** list. Both the keys and values are displayed during this print function in a list format. Another 50 dashes are printed, forming a barrier between the above output and any following output, and the program returns to the beginning of the if statement and displays ***MENU*** once again.

Menu Option 3 – Writing Data to JSON File

The third portion of the ***if*** statement executes if ***menu_option*** is equal to ‘3’. If this is true, the file object is opened using the value of ***FILE_NAME*** in write mode. Once the file is open, I used the dump command to write the contents of the ***students*** list to the JSON file. Once the data is written, a message is displayed to the user showing what data was written. This is accomplished through a ***for*** loop that prints the first name, last name, and course name for every dictionary in the ***students*** list. After this, the file is closed.

Menu Option 4 – Exiting the Program

The fourth portion of the ***for*** loop initiated if ***menu_option*** is equal to ‘4’. If this happens, the ***break*** keyword is used to exit from the original ***while*** loop and end the program. A message is then displayed that reads ‘***Program Ended***’. Finally, the end of the ***if*** statement prints a message that tells the user to select only options 1 – 4 if ***menu_option*** is equal to any other value than 1 through 4.

Error Handling

I added error handling code to the portion of the program where the JSON file is initially loaded into the ***students*** list. The error handling code will catch any errors that happen while loading the file, and if the error relates to the file not existing, an error message is raised that reads: ***Text file must exist before running this script!*** The default Python error message and doc are also displayed. If any other error occurs while loading the JSON file, a message is printer that reads: ***There was a non-specific error!***

The Python error message and docs are also displayed when this happens. If any error occurs, the program checks to see if the JSON file is closed and closes it if it's open.

I also added error handling code to the portion of the program where the user is prompted to enter student and course information. The program checks to see if user's input only contains alphabetical characters. If characters are entered that are not alphabetic, an error is thrown that reads: ***The first name should contain only letters***. This also happens when capturing the last name input.

Lastly, I added error handling code to the portion of the program that writes the ***students*** list back to the JSON file. If a type error occurs, a message is displayed that reads: ***Please check that the data is a valid JSON format***. The built-in error message is also displayed. If any other error occurs, the built-in Python error message is displayed. Finally, the code checks to see if the file is closed and will close it if it's open.

Summary

In conclusion, my program loads data from a JSON file into a list variable. Error handling will raise an error if the file doesn't exist. The program then captures user input regarding student and course information and stores it within the respective variables. Error handling ensures the user only enters alphabetic characters or an error is raised. The user input is stored within a dictionary and appended to a list, creating a list of dictionaries. The program can also display each dictionary within the list variable and write the contents of the list back to the JSON file. Error handling will throw an error if the writing fails. Finally, the user can exit the program. These options are all navigable by the user via menu choices. If any option other than 1 – 4 is chosen by the user, a message is displayed asking the user to only select a valid menu option.