Fan Yang

August 7, 2025

Course Name: Foundations Of Programming: Python

GitHub link: https://github.com/Fyang712/IntroToProg-Python-Mod05

Assignment 05

Intro

In this lesson, I learned the differences between Lists and Dictionaries; how to write data to a file from a Dictionary and how to read data from a file into a Dictionary; and how to handle errors in Python.

Programming Steps

I first defined the data constant and variables. Since this assignment deals with dictionaries and reading and writing data into a json file, I added the "import json" code in the beginning of my codes.

```
import json

# Define the Data Constants
MENU: str = '''
---- 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
-------------------------------------
'''
# Define the Data Constants
FILE_NAME: str = "Enrollments.json"

# Define the Data Variables and constants
student_first_name: str = ''  # Holds the first name of a student entered by the user.
student_last_name: str = ''  # Holds the last name of a student entered by the user.
course_name: str = ''  # Holds the name of a course entered by the user.
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.
student_data: dict = {}  # one row of student data (TODO: Change this to a Dictionary)
students: list = []  # a table of student data
```

Once the data constants and variables were defined, I wrote the code to read the content of an existing json file into the list. I used the Enrollments.JSON file that came with the .py starter file and added in error handling codes.

```
# When the program starts, read the file data into a list of lists (table)
# Extract the data from the file
try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError as e:
    print("Text file must exist before running this script!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
except Exception as e:
    print("There was a non-specific error!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep='\n')
finally:
    if file.closed == False:
        file.close()
```

Once the file was read into the list, I proceeded to write the codes to present the menu choices. There are four different menu choices, and I used conditional loops to:

1. Register a Student for a Course.
2. Show current data.
3. Save data to a file.
4. Exit the program.

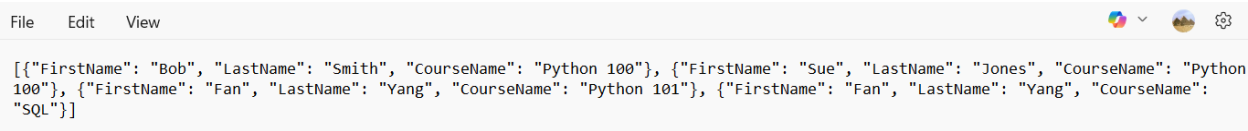
I added in the error handling codes under menu choice 1 and 3 to flag any errors as requested by the assignment.

```python
# Present and Process the data
while (True):

    # Present the menu of choices
    print(MENU)
    menu_choice = input("What would you like to do: ")

    # Input user data
    if menu_choice == "1":  # This will not work if it is an integer!
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_first_name.isalpha():
                raise ValueError("The first name should not contain numbers.")

            student_last_name = input("Enter the student's last name: ")
            if not student_last_name.isalpha():
                raise ValueError("The last name should not contain numbers.")

            course_name = input("Please enter the name of the course: ")

            student_data = {"FirstName": student_first_name,
                            "LastName": student_last_name,
                            "CourseName": course_name}
            students.append(student_data)
        except ValueError as e:
            print(e)  # Prints the custom message
            print("-- Technical Error Message -- ")
            print(e.__doc__)
            print(e.__str__())
        except Exception as e:
            print("There was a non-specific error!\n")
```

```python
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
        print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
        continue

    elif menu_choice == "2":

        # Process the data to create and display a custom message
        print("-" * 50)
        for student in students:
            print(f'Student {student["FirstName"]} '
                  f'{student["LastName"]} is enrolled in {student["CourseName"]}')
        continue
```

```python
    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(students, file)
            file.close()
            continue
        except TypeError as e:
            print("Please check that the data is a valid JSON format\n")
            print("-- Technical Error Message -- ")
            print(e, e.__doc__, type(e), sep='\n')
        except Exception as e:
            print("-- Technical Error Message -- ")
            print("Built-In Python error info: ")
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if file.closed == False:
                file.close()

    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, 3, or 4")

print("Program Ended")
```

I tested out the codes in both Pycharm and Command Prompt to test out the codes and it worked both ways.



Summary

This week, I learned the differences between Lists and Dictionaries; how to write data to a file from a Dictionary and how to read data from a file into a Dictionary; and how to handle errors in Python. The codes built upon the lessons and concepts learned previous week and expanded the capability by incorporating dictionaries, reading and savings data from and to json file, and building in error handling codes.