FERNANDO TAMAYO GRADOS
November 13, 2023
IT FDN 110 B
Assignment 5
https://github.com/fernandotg123/IntroToProg-Python

Assignment 5

1. Intro

This week we learned about dictionaries in Python, which are key value pairs and are more efficient for data processing and retrieving than lists. We also learned about JSON files and how do they differ from CSV files. JSON files work great with lists, as they also consist of key value pairs. We will re-run the previous assignment

2. Problem Statement

The assignment requires creating a program that asks a user to select one of the four menu options in order to register, visualize and store details of a student. The assignment also requires a JSON file to start the program and for names to do not have any numbers on their values. Hence, structured error handling is required throughout the program.

3. JSON set up

To kick off the code, I stated “import json” in the Python script, so that it is easier to work with JSON files.

4. Define data constants and variables

Then, I continued the assignment by defining the data constants. I made sure that I created these constants in CAPITAL LETTERS, as well as added the data types for easy understanding of the reader. Since the first string is large, I used “”” “”” in order to incorporate the full menu. See Figure 1.

```
# 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"
```

Figure 1: Data constants

When I finished with the data constants, I created the variables with null values by using " ", although I could have also used ' '. I also added one empty dictionary called *student_data* and a list of dictionaries called *students*. See Figure 2.

```
# 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.
student_data: dict = {} #one row of student data
students: list = []  # a table of student data
file_data: str = ''  # Holds combined string data separated by a comma.
file = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.
```

Figure 2: Data variables

5. Opening the JSON file

Before asking the user for a choice, we open the file. Since a file may not exist, then we present an exception and we ask the user to create a file before running the program.

```
# Extract the data from the file
try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError:
    print("There must be a file before starting the program.")
    print("")
    print("Create a json file and start the program again.")
    exit()
```

Figure 3: Opening file

For example, if there is no file, this exception will happen:

```
/usr/local/bin/python3.12 /Users/gradosf/Documents/PythonScripts/_Module05/Assignment/Assignment05-Starter.py
There must be a file before starting the program.

Create a json file and start the program again.

Process finished with exit code 0
```

Figure 4: Exception when there is no file

6. Present the menu of choices

To present the menu of four choices, I started with the while loop statement, so that whenever an input finishes, then the menu can appear again in addition to requesting the user to input an option of the menu. A user can only select options 1 to 4. If any other input is provided, the program will ask for the input again.

If choice 1 is chosen, then we help the user to input a correct value for the first and last name, using the “try” and “except” block. Student_data creates the dictionary by providing keys to the values entered by the user:

```
# 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():
            print("The first name should not contain numbers.")
            student_first_name = input("Enter the student's first name: ")
    except:
        continue
    try:
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            print("The last name should not contain numbers.")
            student_last_name = input("Enter the student's last name: ")
    except:
        continue
    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)
    print(f"You have registered {student_first_name} {student_last_name} for
{course_name}.")
    continue
```

Figure 5: Option 1 code

If there is an error with the name, then the program will ask the user to input the name again:

```
What would you like to do: 1
Enter the student's first name: Fernand0
The first name should not contain numbers.
Enter the student's first name: Fernando
Enter the student's last name: Tama10
The last name should not contain numbers.
Enter the student's last name: Tamayo
Please enter the name of the course: Python 100
You have registered Fernando Tamayo for Python 100.
```

Figure 6: Option 1 exception output

If choice 2 is chosen, then we convert the values in the dictionary into a string separated by commas:

```
# Present the current data
elif menu_choice == "2":

    # Process the data to create and display a custom message
    print("-"*50)
    for student in students:

print(f"{student['FirstName']},{student['LastName']},{student['CourseName']}"
)
    print("-"*50)
    continue
```

Figure 7: Option 2 code

Hence, the output will read in strings as follows:

```
What would you like to do: 2
--------------------------------------------------
fer,tam,python
stef,vidarte,python
amelia,tamayo,python
juan,fernando,python
juan,tamayo,python
bob,axe,billions
Fernando,Tamayo,Python 100
--------------------------------------------------
```

Figure 8: Option 2 output

If choice 3 is chosen, then we open the JSON file and *dump* all the rows into the file.

```
# Save the data to a file
elif menu_choice == "3":
    try:
        file = open(FILE_NAME, "w")
```

```
        json.dump(students,file)
        file.close()
        print("The following data was saved to file!")
        for student in students:
            print(f"{student['FirstName']} {student['LastName']} is enrolled
in {student['CourseName']}")
        continue
    except Exception as e:
        print("Technical error",e)
    finally:
        if file.closed == False:
            file.close()
```

Figure 9: Option 3 code

```
What would you like to do: 3
The following data was saved to file!
fer tam is enrolled in python
stef vidarte is enrolled in python
amelia tamayo is enrolled in python
juan fernando is enrolled in python
juan tamayo is enrolled in python
bob axe is enrolled in billions
Fernando Tamayo is enrolled in Python 100
```

Figure 10: Option 3 output

Finally, we close the program:

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

print("Program Ended")
```

Figure 10: Option 4 code

6. Conclusion

Overall, this has been a fun exercise to learn about dictionaries. These are more manageable than lists. I will share this assignment in Github.