Name: Diellza Mehmeti
Date: 11/15/2024
Course: Foundations of Programming: Python
Assignment: 05
GitHub

Dictionary Collections

Introduction

In this week, I started using dictionaries to collect, and present data. I was introduced to JavaScript Object Notation (JSON) files and to structured error handling (try-except). I continued working on my code from last week where I used dictionaries instead of lists to input and save data in the file. I used error handling codes when the user reads from the file, writes in it, and tries to enter the first name and last name. Finally, I created a repository in GitHub where I saved the script and the documentation.

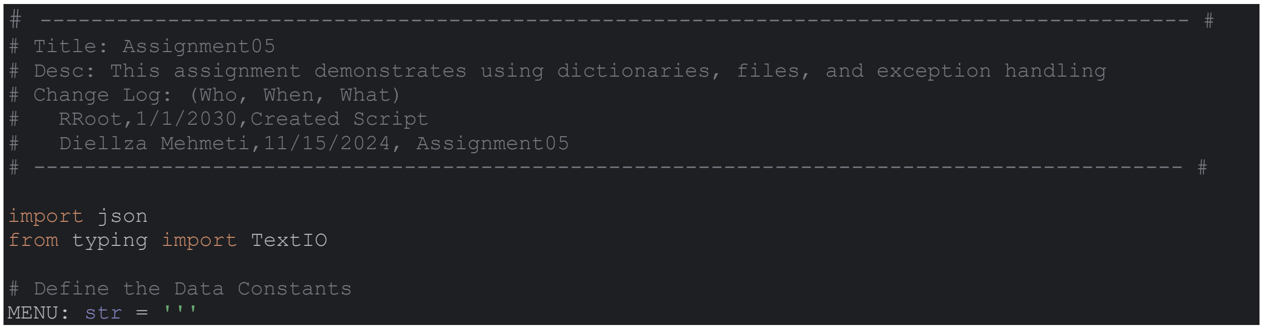
Script Header, Comments, Constants, and Variables

I started the assignment by utilizing the setup that Professor Randal had saved. I updated the name, title, and logs in the script header. Comments were set and I added a few to document my code. I then declared the two constants given in the assignment. The first constant will display the menu of choices for the user and then I defined the constant that will be used to name the JSON file. After that, I declared eight variables. This week I included a new data type, dictionary, to collect the list of data.

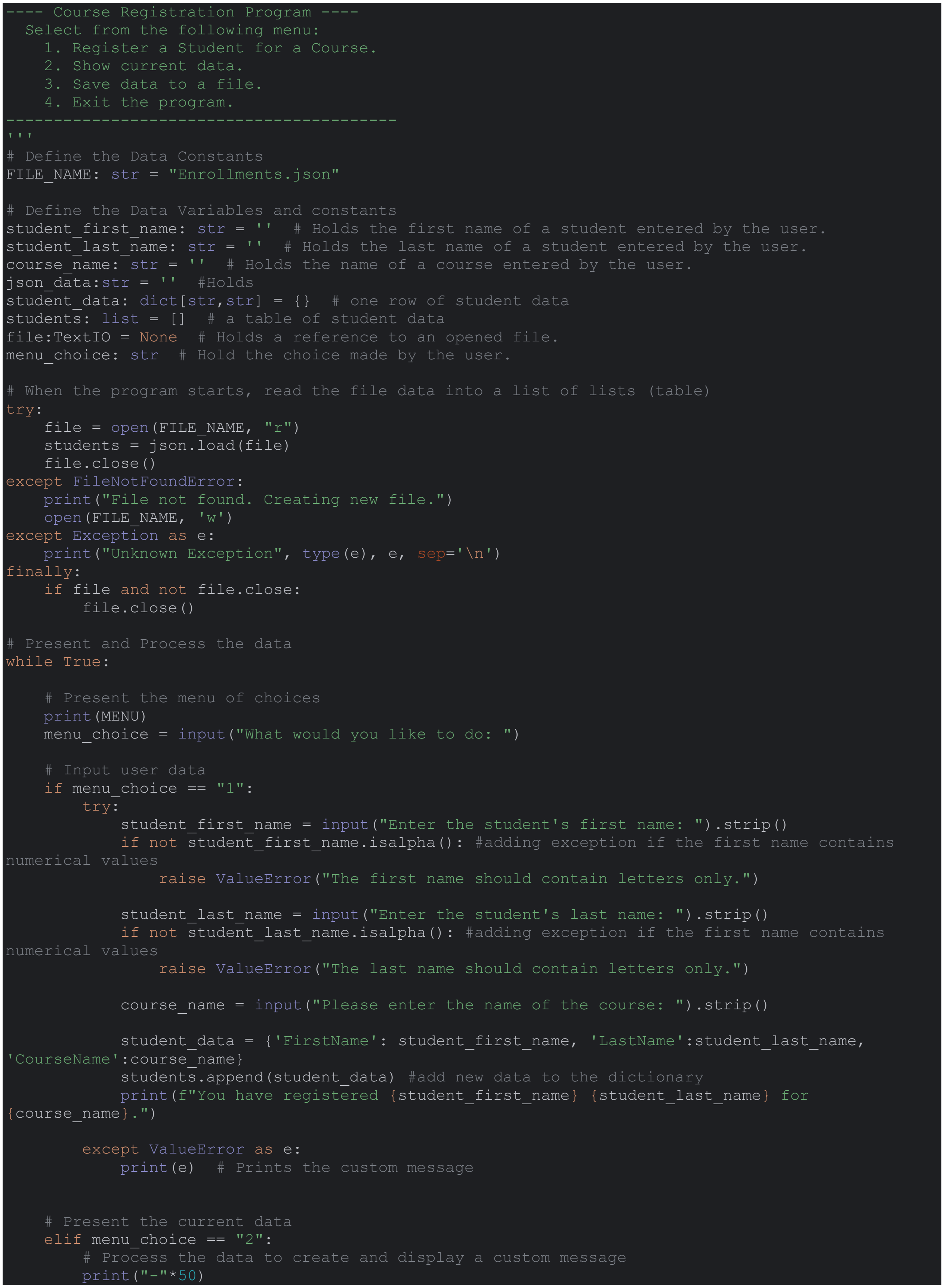
Input and Output

I started the assignment by using syntax to read the data from the json file. I also included a few errors exception handlings if the file does not exist when the program starts to read the file. I tested the errors in this part of the code, and they worked as intended. If the json file does not exist, the user gets an error, and a new file is created. Similar to last week, I used the first three variables (student_first_name, student_last_name, and course_name) to create the input. This week, I also included codes to handle errors if the user inputs a numerical value for the first name and last name. If the value is numerical, the user gets an error suggestion to include letters only. Once the user gets the display of the MENU constant to choose between the four choices, the while loops runs through the choices depending on what the user selects. Finally, after the while() function prints the MENU of choices, if the user enters 4, the program ends.

Here is the example code in PyCharm:



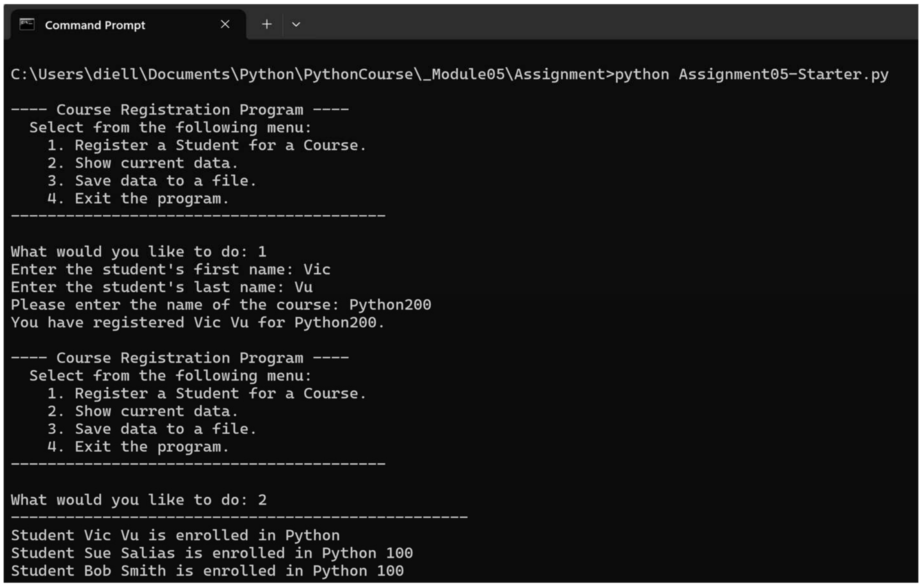
```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   RRoot,1/1/2030,Created Script
#   Diellza Mehmeti,11/15/2024, Assignment05
# ------------------------------------------------------------------------------------------ #

import json
from typing import TextIO

# 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.
json_data:str = ''  #Holds
student_data: dict[str,str] = {}  # one row of student data
students: list = []  # a table of student data
file:TextIO = None  # Holds a reference to an opened file.
menu_choice: str  # Hold the choice made by the user.

# When the program starts, read the file data into a list of lists (table)
try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError:
    print("File not found. Creating new file.")
    open(FILE_NAME, 'w')
except Exception as e:
    print("Unknown Exception", type(e), e, sep='\n')
finally:
    if file and not file.close:
        file.close()

# 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":
        try:
            student_first_name = input("Enter the student's first name: ").strip()
            if not student_first_name.isalpha(): #adding exception if the first name contains
numerical values
                raise ValueError("The first name should contain letters only.")

            student_last_name = input("Enter the student's last name: ").strip()
            if not student_last_name.isalpha(): #adding exception if the first name contains
numerical values
                raise ValueError("The last name should contain letters only.")

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

            student_data = {'FirstName': student_first_name, 'LastName':student_last_name,
'CourseName':course_name}
            students.append(student_data) #add new data to the dictionary
            print(f"You have registered {student_first_name} {student_last_name} for
{course_name}.")

        except ValueError as e:
            print(e)  # Prints the custom message


    # 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 {student['FirstName']} {student['LastName']} is enrolled in
{student['CourseName']}")
        print("-"*50)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(students, file)
            print("The following data was saved to file!")
            for student in students:
                print(f"Student {student['FirstName']} {student['LastName']} is enrolled in
{student['CourseName']}")
        except Exception as e:
            print("Error saving to file")
            print(e)
        finally:
            if file and not file.closed:
                file.close()

    # Stop the loop
    elif menu_choice == "4":
        break  # out of the loop
    else:
        print("Please only choose option 1, 2, or 3") #prints the message if user enters any
value other than 1,2,3

print("Program Ended")
```

Testing and running the script

Once I completed the code, and after numerous tests, I ran in some errors when trying to use the error handling syntax and when I was translating the data from list to dictionary. I was able to resolve these by checking the json file and updating the key “email” to “LastName. I tested various scenarios in the code and ensured that I was using the same key code throughout the code and in the file. Once I finalized the testing portion, I ran the code in PyCharm, and then in CMD. I saved the file with the “py” extension, uploaded it in the GitHub and shared it in the discussion board for my peers to review.



```
C:\Users\diell\Documents\Python\PythonCourse\_Module05\Assignment>python Assignment05-Starter.py

---- 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: 1
Enter the student's first name: Vic
Enter the student's last name: Vu
Please enter the name of the course: Python200
You have registered Vic Vu for Python200.

---- 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: 2
--------------------------------------------------
Student Vic Vu is enrolled in Python
Student Sue Salias is enrolled in Python 100
Student Bob Smith is enrolled in Python 100
```

Figure 1: Running Python code in CMD and displaying menu choice 1 and 2

Summary

In this week’s assignment, I continued to work from assignment four. I used dictionaries to save and process data instead of lists. The other two learning items for this week included working with json files where I used a code to read from the file when the program starts and write to a json file when the user wants to save the data. I added error handling codes (try-except) when the file is read, when the user enters their first name and last name, and when the file is written. Once the code was tested, I ran it in PyCharm and CMD. Finally, I saved the code in GitHub and shared it with my peers for review.