

Yunjia He

5/21/25

IT FND 110 A

Assignment05

<https://github.com/Yunjia13/IntroToProg-Python-Mod05.git>

Advanced Collections and Error Handling

Introduction

This week, I learned the using dictionaries, json files, and exception handling. I created a script with menu, and user can make selection. User can register students for course or save data to a file. This script can save data in dictionaries and json file. The following information is a breakdown of how I wrote this program.

Creating the Program

I start my program off by defining constants and variables with type hints. I set "MENU" and "FILE_NAME" as string type data constant. I set others data variables.(Figure 1.1)

```
# 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
-----
"""

FILE_NAME: str = "Enrollments.json"

# Define the Data Variables
student_first_name: str = ""
student_last_name: str = ""
course_name: str = ""
file: object = None
menu_choice: str = ""
student_data: dict
students: list = []
```

Figure 1.1: Define the data

Next step, I need to load data from file to this program. First, I import json to process the contents of a file. Second, I use “try-except” to handle program error. When file is not exist, the program will show error info and stop running. Third, I use “json.load” to load json file data to “students” variable. Last, I use “close()” to close file.(Figure 1.2)

```
#Load data from file
import json

try:
    file = open(FILE_NAME, "r")
    students = json.load(file)
    file.close()
except FileNotFoundError as e:
    print("File must exist before running this program!\n")
    print("Built-In Python error info: ")
    print(e, e.__doc__, type(e), sep= '\n')
    print("program ended!")
    quit()
except Exception as e:
    print("There was a non-specific error!\n")
    print("-- Technical Error Message -- ")
    print(e, e.__doc__, type(e), sep= '\n')
    print("program ended")
    quit()
```

Figure 1.2: Load the data

Now I need to present the menu of choices and capture the selection. I use “while” to keep user in program until they choose exit.(Figure 1.3)

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

Figure 1.3: Present the menu of choices

After getting selection, I use “if” and “elif” in different choices. If user enter “1”, user can input their name and course name. I use “name.isalpha()” to check user’s input. The program will end input step and remind user to enter name correctly, if user’s name contain number. If name is alpha, the data will be saved in “student_data” dictionary. If user enter

"2", the program will present the current data from "students". If user enter "3", the program will write the "students" into json file and show the current data that in the "students" variable. I use "try-except" to handle program error in data saving step. If user enter "4", the "while" loop will break, and the program will end. If user enter other things, the program will remind user to choose option.(Figure 1.4~1.7)

```
# Input user data
if menu_choice == "1":
    try:
        student_first_name = input("Enter the student's first name? ")
        if not student_first_name.isalpha():
            raise ValueError("First name should not contain numbers!")
    except ValueError as e:
        print(e)
        print("-- Technical Error Message -- ")
        print(e, e.__doc__, type(e), sep= '\n')
        print()
        continue
    try:
        student_last_name = input("Enter the student's last name? ")
        if not student_last_name.isalpha():
            raise ValueError("Last name should not contain numbers!")
    except ValueError as e:
        print(e)
        print("-- Technical Error Message -- ")
        print(e, e.__doc__, type(e), sep= '\n')
        print()
        continue
    course_name = input("Please enter the course's name? ")
    student_data:dict = {"First_name": student_first_name,
                        "Last_name": student_last_name,
                        "Course": course_name}
    students.append(student_data)
    print()
```

Figure 1.4: Menu_choice 1: Input user data

```
# Present the current data
elif menu_choice == "2":
    print("The current data is:")
    for student in students:
        print(f'{student["First_name"]}, '
              f'{student["Last_name"]}, '
              f'{student["Course"]}')
    print()
```

Figure 1.5: Menu_choice 2: Present the current data

```

# Save the data to a file
elif menu_choice == "3":
    try:
        file = open(FILE_NAME, "w")
        json.dump(students, file, indent = 2)
        file.close()
        print("Data saved! The current data is:")
    except FileNotFoundError as e:
        print("File must exist before running this program!\n")
        print("Built-In Python error info: ")
        print(e, e.__doc__, type(e), sep='\n')
        print("program ended!")
        quit()
    except Exception as e:
        print("There was a non-specific error!\n")
        print("-- Technical Error Message -- ")
        print(e, e.__doc__, type(e), sep='\n')
        print("program ended!")
        quit()
    for student in students:
        print(f'{student["First_name"]}, '
              f'{student["Last_name"]}, '
              f'{student["Course"]}')
    print()

```

Figure 1.6: Menu_choice 3: Save data

```

# Stop the loop
elif menu_choice == "4":
    break
else:
    print("Please only choose option 1~4")
    print()
print("Program Ended")

```

Figure 1.7: Menu_choice 4: Exit program

Testing the Program

Now that the code is complete. It's time to run and test it. First, I save the script and run it in Pycharm and CMD. The following information is test result.(Figure 2.1~2.5)

```
{ } Enrollments.json X
C: > Users > HYJLO > Documents > Python > PythonCourse > a05 > { } Enrollments.json
1  [
2      {
3          "First_name": "Yunjia",
4          "Last_name": "He",
5          "Course": "Python 100"
6      },
7      {
8          "First_name": "Yuan",
9          "Last_name": "Li",
10         "Course": "Python 100"
11     }
12 ]
```

Figure 2.1: Enrollments.json

```
---- 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? Bob
Enter the student's last name? He
Please enter the course's name? Math 101

---- 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? Tow
Enter the student's last name? He
Please enter the course's name? English 100
```



```
What would you like to do? 2

The current data is:
Yunjia, He, Python 100
Yuan, Li, Python 100
Bob, He, Math 101
Tow, He, English 100

---- 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? 3

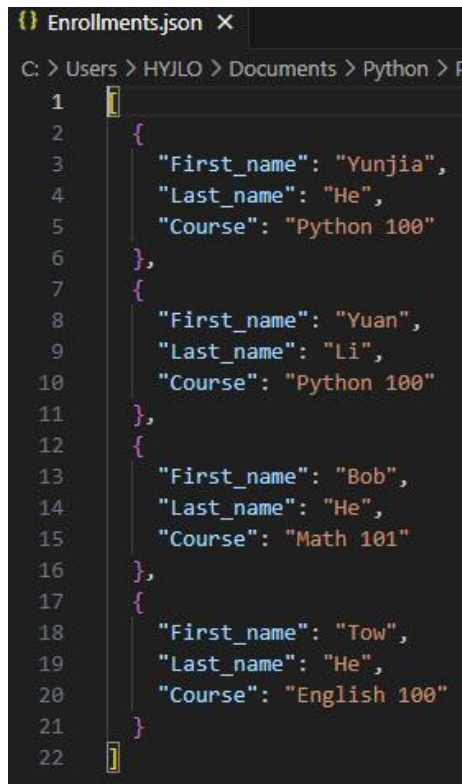
Data saved! The current data is:
Yunjia,He, Python 100
Yuan,Li, Python 100
Bob,He, Math 101
Tow,He, English 100

---- 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? 4

Program Ended
```

Figure 2.2: Pycharm test result

A screenshot of a code editor window titled 'Enrollments.json'. The file path is 'C:\Users\HYJLO\Documents\Python\Python 100'. The JSON content is as follows:

```
1  {
2      "First_name": "Yunjia",
3      "Last_name": "He",
4      "Course": "Python 100"
5  },
6  {
7      "First_name": "Yuan",
8      "Last_name": "Li",
9      "Course": "Python 100"
10 },
11 {
12     "First_name": "Bob",
13     "Last_name": "He",
14     "Course": "Math 101"
15 },
16 {
17     "First_name": "Tow",
18     "Last_name": "He",
19     "Course": "English 100"
20 }
21 }
22 }
```

Figure 2.3: Enrollments.json(after Pycharm test)

```
---- 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? Jay
Enter the student's last name? Chou
Please enter the course's name? Python 100

---- 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

The current data is:
Yunjia, He, Python 100
Yuan, Li, Python 100
Bob, He, Math 101
Tow, He, English 100
Jay, Chou, Python 100
```

```

---- 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? 3

Data saved! The current data is:
Yunjia,He, Python 100
Yuan,Li, Python 100
Bob,He, Math 101
Tow,He, English 100
Jay,Chou, Python 100

---- 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? 4

Program Ended

```

Figure 2.4: CMD test result

```

{} Enrollments.json X
C: > Users > HYJLO > Documents > Python >
1  [
2    {
3      "First_name": "Yunjia",
4      "Last_name": "He",
5      "Course": "Python 100"
6    },
7    {
8      "First_name": "Yuan",
9      "Last_name": "Li",
10     "Course": "Python 100"
11   },
12   {
13     "First_name": "Bob",
14     "Last_name": "He",
15     "Course": "Math 101"
16   },
17   {
18     "First_name": "Tow",
19     "Last_name": "He",
20     "Course": "English 100"
21   },
22   {
23     "First_name": "Jay",
24     "Last_name": "Chou",
25     "Course": "Python 100"
26   }
27 ]

```

Figure 2.5: Enrollments.json(after CMD test)

Summary

I review using dictionaries, json files, and exception handling by writing this program. I learn that dictionaries have “keys” and “values” can help data management. It is convenient to processes files contents with dictionaries. I grasp using “try-except” to handle or customs error. I also review lists using, file handling, and looping in Python. Lastly, this practice helped me start programming with Python and review knowledge.