Justin Price

May 17, 2024

IT FDN 100 A

Assignment 5

Codegardium Pythonosa

Introduction:

“It’s pronounced Pythonosa, not Pythonosaaa” Professor Root and Professor Justin said in tandem. “Sorry, I am quite new to the Pyhton Arts,” said Professor Nahvese. “Be that as it may, If you would like to use my Python script for registering classes, then perhaps I should make the script more user-friendly,” said Professor Justin. “Brilliant!” Professor Root exclaimed. “Please annotate your work with great detail and distribute your findings to the rest of the faculty at Hogwarts University(HU).

Main Body:

As stated, I will treat each script as a continuous working file.

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   JP,13MAT24,Created Script
#   JP,16MAY24,Added Exception
# ------------------------------------------------------------------------------------------ #
```

Header figure 1.1

Below is the list of Variables and Constants that I will be using for this script.

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

Constants and Variables 2.1

A current issue that I have been facing is that when I run the file, it often does not know where to pull the Enrollments.json if it is in the same folder as the script. To fix that, I implemented a

code that creates the JSON file if there isn't one already. This is achieved by writing exceptions that first try to open the file and acknowledge the FileNotFound error, then display an error message that is easy to understand. This process ensures the creation of a JSON file that can be used with the script.

```
#create a dictionary
'''
This portion is to create a json file, assuming that there has not been a file that
has not been created or a defined pathway. >>>>
'''
import json

try:
    file = open(FILE_NAME, "r")
    student_table = json.load(file)
    for item in students:
        print(f"FirstName: {item['FirstName']}, LastName: {item['LastName']}, Course: {item['Course']}")
except FileNotFoundError as e:
    print("JSON 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')
try:
    print("Prof. Justin will create a JSON for you..")
    student_row1: dict = {"FirstName": "First Name", "LastName": "Last Name", "Course": "Course"}
    student_table: list = [student_row1]
    file = open("Enrollments.json", "w")
    json.dump(student_table, file)
    file.close()
finally:
    print("JSON successfully created")


'''
<<<< This portion is to create a json file, assuming that there has not been a file that
has not been created or a defined pathway.
'''
```

Exception before Main script 3.1

For menu choice one, I did something similar: I added an exception and raised a ValueError to notify the user that they had improperly input a wrong character. Once the data was correctly added, I added it as a new row in the dictionary.

For menu choice option 3, I did something similar to menu choice 1.

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

        # Process the data to create and display a custom message
        print("-"*50)
        print("\nCurrent registered students:")
        print(csv_data)
        print("-"*50)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            json.dump(student_table, file)
        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')
        else:
            file.close()
            print("The following data was saved to file!")
            # try to read data from json
        finally:
            file = open(FILE_NAME, "r")
            student_table = json.load(file)
            for item in student_table:
                print(f"FirstName: {item['FirstName']}, LastName: {item['LastName']}, Course: {item['Course']}")
            file.close()
        continue
```

Menu choice 2 with and menu choice 3 with exception 4.1

Menu choice 4 ends 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")
```

Constants and Variables 5.1

This time I tested the python script in IDLE rather than the command line.

```
JSON file must exist before running this script!

-- Technical Error Message --
[Errno 2] No such file or directory: 'Enrollments.json'
File not found.
<class 'FileNotFoundError'>
Prof. Justin will create a JSON for you..
JSON successfully created

---- 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:
Remus
Enter the student's last name:
Lupin
Please enter the name of the course:
Python100
You have registered Remus Lupin for Python100.

---- 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.
-----------------------------------------
```

IDLE Test 6.1

```
---- 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
--------------------------------------------------

Current registered students:
Remus,Lupin,Python100
Alastor,Moody,Python100

--------------------------------------------------

---- 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.
----------------------------------------
```

IDLE Test 6.2

```
What would you like to do:
3
The following data was saved to file!
FirstName: First Name, LastName: Last Name, Course: Course
FirstName: Remus, LastName: Lupin, Course: Python100
FirstName: Alastor, LastName: Moody, Course: Python100

---- 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


** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

IDLE Test 6.3

Summary:

"Thank you, Professor Justin; your Python script was a lifesaver," said Professor Nahvese. Not only was I able to register students for my Methods of Method Acting class, but I was also able to teach Professor Jack Black for his Rock class.