Denesse Citlaly Gomez

May 26, 2024

IT FDN 110A

Assignment 05

Creating Python Script Using Programming Tools and Techniques: Advanced Collections and Error Handling

Introduction

For assignment 5, the task was to create a Python script with similar outputs as assignment 4; however, this assignment required the use of exception handling for data processing. By using structured error handling, we can provide a way for the program to respond to unexpected scenarios. In addition, JSON files were used for this script rather than the csv files that we were using for previous codes.

Creating Python Script for Task

JSON Files

I began the script by importing a JSON file. A JSON file is a JavaScript Object Notation file, which is used to store structured data in a simple format. In previous assignments, I was using csv files to store data, but I found that it was easier to structure data within a JSON file than in a csv file. Compared to the csv file, a JSON file includes key-value pairs, which have a unique identifier and a data value associated with it. This is what makes it easier to structure data in a JSON file. In a JSON file, you can have different data types, whereas in a csv file you can only have strings and data including numbers. I enjoyed learning about this type of file because it was easy to structure my data with it. To import the JSON file, I just used to write “import JSON” as seen in the code below. JSON files often contain data structured in a similar way as dictionaries.

Exception Error Handling

For the next part of the code, exception error handling statements were used. The “try” statement is used for exception error handling and is often paired with “except” clauses. These statements are valuable because they help you include a way to deal with unexpected scenarios when the user inputs data when using the code. The “except” clauses must be included because if it is not included then the code itself may terminate since it does not know how to deal with an unexpected scenario. However, with an “except” clause, if an exception does happen in the code, these clauses will catch it. As seen in the code below, “finally” was also used. All code under this “finally” statement will be executed even if the exception does not occur. In this section of the code, you will also see a line with “print(e.__doc__)”. This “__doc__” is an attribute to access a documentation string, which is a string used in the first statement, in this case, that line will return the documentation string of the variable “e”.

Opening and Saving Data in JSON File

In the code, the open() function was used to open a JSON file in a read format as indicated by the "r" inside the exception error handling "try" statement. The json.load() module was also used. This module is used to convert the JSON file information into a Python object. In addition, the json.dump() function does the opposite effect of the json.load() function. This json.dump() function insert Python objects into the JSON file. For the menu 3 choice, the code was supposed to save the user input into the JSON, so the json.dump() function was used in this case.

Summary

In conclusion, through this assignment, I learned about JSON files and exception handling. I verified the code using both PyCharm and Command Prompt, and I have included the outputs in Figure 2 (down below). I was able to successfully run the code and save various data that I input into the JSON file. I did run into a couple of errors during the process, which demonstrated to me how case-sensitive Python is. Overall, I liked how easy it was to modify and retrieve data from the JSON file.

Link to GitHub Repository

Resources

1. Mod05-Notes

Python Script

Figure 1: Python Script Reference

```python
# ------------------------------------------------------------------------------------------ #
# Title: Assignment05
# Desc: This assignment demonstrates using dictionaries, files, and exception handling
# Change Log: (Who, When, What)
#   DGomez,5/26/24,Created Script
#   <Denesse Citlaly Gomez>,<5/26/24>, <Assignment 05>
# ------------------------------------------------------------------------------------------ #
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.
student_data: dict = {} # one row of student data
students: list = []  # a table of student data
# csv_data: str = '' # csv_data: str = ''  # Holds combined string data separated by a comma.
json_data: str = '' # Holds combined string data in a json file
```

```python
file = 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)
# Extract the data from the file

try:
    file = open(FILE_NAME, "r")
    students = json.load(file)

    file.close()

except Exception as e:
    print("Error opening file")
    print("Please check you have entered the correct json")
    print(e.__doc__)
finally:
    if file.closed == False:
        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":  # 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("Student First Name must be alphabetic")
                student_last_name = input("Enter the student's last name: ")
                if not student_last_name.isalpha():
                    raise ValueError("Student Last Name must be alphabetic")
                course_name = input("Please enter the name of the course: ")
                student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
                students.append(student_data)
                print(f"You have registered {student_first_name} {student_last_name} for {course_name}.")
            except ValueError as e:
                print(e)
            except Exception as e:
                print("Error with entered data")
                print(e.__doc__)
            continue

        # 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['first_name']} {student['last_name']} is enrolled in {student['course_name']}")
            print("-" * 50)
            continue

        # 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 a file!")
                for student in students:
                    print(f"Student {student['first_name']} {student['last_name']} is enrolled in {student['course_name']}")
            except Exception as e:
                if file.closed == False:
                    file.close()
                print('Error saving data to file')
                print(e)
                print(e.__doc__)
            continue

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

    print("Program Ended")
```

Python Script Verification

Figure 2: Python Script Verification

```
C:\Python\Python3.x\python.exe "C:\Python\Python3.x\UW Foundations to Python Homeworks\_Module05\Assignment\Assignment05.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: Denesse
Enter the student's last name: Gomez
Please enter the name of the course: Python 100
You have registered Denesse Gomez for 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: 1
Enter the student's first name: Kevin
Enter the student's last name: Martinez
Please enter the name of the course: Business 101
You have registered Kevin Martinez for Business 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: 2
--------------------------------------------------
Student Denesse Gomez is enrolled in Python 100
Student Kathy  is enrolled in Psychology
Student Paolo Salazar is enrolled in Biology
Student Denesse Gomez is enrolled in Python 100
Student Kevin Martinez is enrolled in Business 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: 2
--------------------------------------------------
Student Denesse Gomez is enrolled in Python 100
Student Kathy  is enrolled in Psychology
Student Paolo Salazar is enrolled in Biology
Student Denesse Gomez is enrolled in Python 100
Student Kevin Martinez is enrolled in Business 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: 3
The following data was saved to a file!
Student Denesse Gomez is enrolled in Python 100
Student Kathy  is enrolled in Psychology
Student Paolo Salazar is enrolled in Biology
Student Denesse Gomez is enrolled in Python 100
Student Kevin Martinez is enrolled in Business 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: 4
Program Ended

Process finished with exit code 0
```