Julia Westfall
May 07, 2024
Foundations of Programming: Python
Assignment04

Using Lists to Collect Data

Introduction

This assignment builds off of assignment 03 through its similar use of conditional logic, variables, and storing data in a file. This assignment differs by using lists to allow multiple entries in data menu one rather than just one at a time and also allows the program to write over previous datasets while adding new ones.

Starting the Assignment

Because this assignment was so similar to assignment03, I began by copying my script from assignment03 and then read through what assignment 04 had asked me to do. It appeared I would need to use lists which I was the least familiar with so I went ahead and watched the multidimensional list video to recall how to use and manipulate lists.

Defining Constants and variables

Next, I defined the desired constants and variables per the assignment instructions. Many were identical to the previous assignment. New ones added were “student_data” and “students” which were both lists. The following represents my variable/constants code at this point.

```
# Define the Data Constants
FILE_NAME: str = 'Enrollments.csv'
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 Variables
student_first_name: str = ''
student_last_name: str = ''
course_name: str = ''
csv_data: str = ''
file = None
menu_choice: str = ''
student_data: list = []
students: list = []
program_continue: bool = True
```

Note: This is text that can be manipulated not an image and applies to all similar blocks below

Menu Option 1

I decided to put my new knowledge of lists to use immediately so I jumped to menu option 1 and added the following code:

```
students_list: list = [student_first_name, student_last_name, course_name]
student_data.append(students_list)
```

I was simply trying to see if I made my list correctly and then if I could add my student list to my empty student list variable. To my frustration, my list worked correctly but adding it to student data did not work (I later realized I originally spelled “append” wrong) so I commented the code out and jumped to another item in the meantime.

Data Initializing

The next task I tackled was trying to get my program to read and assign my enrollment data to separate lists and then list of lists. This and step 3 were the most difficult steps for me and took some time. To help with my understanding, I used the debugger and stopped line by line to try and understand what was being stored in each variable and when. I also commented each line to keep track of what was going on. I had copied the first few lines that opened the file and read it while assigning that to a variable. Next, I created another variable I called “lines” because it broke up my long string that was in the csv data and separated them as a singular list by the new line carriage /n. With some collaboration and troubleshooting help from my software engineering python whisperer friend, I found I needed to split the data into lists again (to create multifunctional lists) and also add a for loop to remove the empty string my code created the way it was set up originally. The following was the code for this step:

```
# Behind the scenes will create a new file named enrollments (or open if exists)
file = open(FILE_NAME, 'r')

# creates new variable that reads the enrollments file
csv_data = file.read()

# closes the file
file.close()

# This variable splits each line into separate lists by the new lines /n
lines = csv_data.split('\n')

# Saying for an element in lines variable (which is a line in this case) add student data to it
# in the form of a list instead of a string (as long as string is not equal to empty string)
for line in lines:
   if line is not '':
       student_data.append(line.split(','))
```

Menu Option 3, 2, and 4

Part three opens the enrollments file again and w writes over previous data. I set up a for loop that stated for each element in the student data, I created another string which was then re-written over the data file and the file was closed. I worked this side of code while iterating menu 1 and taking it line by line in the debugger to help me keep track of the data. Menu 2 item just needed a print out of what was being stored by the user so I copied and pasted it from the menu3. Item 4 did not change. Here is the final code for menu 2 and 3:

```
# Present the current data
elif menu_choice == '2':
    for student in student_data:
        temp_line: str = f'{student[0]},{student[1]},{student[2]}'
        print(temp_line)

# Save the new data to a file
elif menu_choice == '3':
    file = open(FILE_NAME, 'w')
    for student in student_data:
        temp_line: str = f'{student[0]},{student[1]},{student[2]}\n'

        file.write(temp_line)

    file.close()
```

Summary

In summary, this was a highly useful assignment as it allowed me to collect data via lists rather than just strings as before while also utilizing the split command to work between both strings and lists. This allows for more complex coding operations and higher functionality of collecting data. Though frustrating, I am sure I will get better at these skills with time.