Tasean Cunningham

November 13th, 2020

Foundations of Programming: Python

Assignment #5: Lists and Dictionaries

Lists and Dictionaries

Introduction

This week, we built upon our knowledge of what we learned about lists last week and were also introduced to dictionaries. Unlike a list where things are organized into sequences, dictionaries are stored in pairs. A dictionary contains a key and a value. We continue to build upon the knowledge from the previous weeks as well. This assignment will demonstrate my knowledge of the material we have learned up to this point in the class.

Creating the Program

This week as a part of our lesson, we were given a piece of partially created script that we were then to update various sections throughout. It is good to begin code with a strong header so that anyone reviewing your code or if a user needs to update their code has a trail to go off. The script that we were adding encompasses a menu of five choices for a user that will enable them to view, add, remove, and save data to a file and then exit. The script starts with an if statement and then I added a print function to check if there is data, if not prompting the user and then a for loop to capture any current data in the file and display that back to the user as seen in figure 1.1.

```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    print("The Current Data Is: \n")
    for row in lstTable:
        print(row["Task"] + "," + row["Priority"])
    continue
```

Figure 1.1 Displays the current item portion of the script.

The remaining code uses an elif statement, which was included in the original script for each section. The second portion of the menu allows a user to input a new task and priority. They are prompted with two inputs and then these values are appended to the lsttable as seen in figure 1.2.

```
elif (strChoice.strip() == '2'):
    newTask = input("What new task would you like to add: ")
    newPriority = input("What new priority would you like to add: ")
    dicRow = {"Task": newTask, "Priority": newPriority}
    lstTable.append(dicRow)
    continue
```

Figure 1.2 Displays the adding a new item section to the script.

The next section of code allows a user to remove an existing item from the file. It prompts a user to input what task or priority they would like to remove and if it is not found then they are asked to insert a different task or priority. I am sure there is a more efficient way to combine these two parts of the code onto a single line, but I was unable to figure it out. I added two separate if statements for the task and priority as displayed in figure 1.3.

```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    strDelete = input("What task or priority would you like to be deleted?: ")
    for row in lstTable:
        if strDelete in row["Task"]:
            lstTable.remove(row)
            if strDelete in row["Priority"]:
                lstTable.remove(row)
            if strDelete not in lstTable:
                print("Insert a new task or priority, task not found on table")
    continue
```

Figure 1.3 Displays the removal of a tsk or a priority.

The final option for the user is to save the data that they have written to a file. The user is given either a choice of yes or no to save the file and if they chose yes then the data they entered in step 2 is saved to the file, if they chose no then they are alerted that their data has not been saved as seen in figure 1.4.

```
# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    print("Are you ready to save your data?")
    saveData = input("Enter 'Yes' or 'No': ")
    objFile = open("ToDoList.txt", "w")
    for row in lstTable:
        objFile.write(row["Task"] + "," + row["Priority"] + "\n")
        objFile.close()
        print("Your Data has been saved to the File")
elif saveData == "n":
    print("Your Data Has Not Been Saved")
    continue
```

Figure 1.4 Displays the save tasks portion of the script.

Finally, the script ends with an option to exit the program as displayed in figure 1.5

```
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    userDone = input("Please press 'enter' to exit the program")
    print("Exiting the Program")
    break  # and Exit the program
```

Running in PyCharm

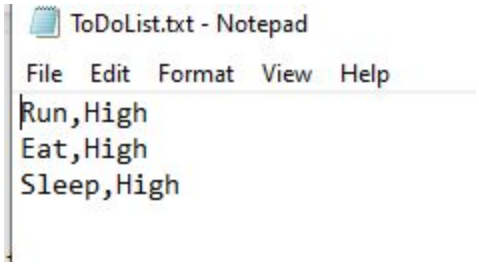
The below figure 1.6 is the program successfully running in Pycharm

```
C:\_PythonClass\Assignment05\venv\Scripts\python.exe C:/_PythonClass/Assignment05/Homework05.py

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

The Current Data Is:

Read,Low
Run,High
Eat,High

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

What new task would you like to add: Sleep
What new priority would you like to add: High

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
```

```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

What task or priority would you like to be deleted?: Read
Insert a new task or priority, task not found on table

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4

Are you ready to save your data?
Enter 'Yes' or 'No': Yes
Your Data has been saved to the File

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] -
```

```
 Which option would you like to perform? [1 to 5] - 5

 Please press 'enter' to exit the program
```

The below figure 1.7 is the output from the text file. As noted, read has been removed from the final file.



Summary

This week we dove deeper into lists and were introduced to working with dictionaries. Dictionaries work similar to a list, but operate in pairs versus operating in a sequence. We were also exposed to working with a script that had been partially created, but was not complete. There were elements that made it easier, but also things that came across as challenging because one is not seeing the full picture of creating their script from start to finish. My homework assignment this week shows my continued knowledge of the material that we continue to learn week in and week out.