

Felix Chen

2/15/23

IT FDN 110A

Assignment05

Creating a menu with lists and dictionaries

Lists

This week we continued where we left off from last week's lecture about lists. Randal went over how to store list data to a file and read list data from a file. It was helpful going over this because it applied to what we had to do this week for our assignment. Also, last week's brief discussion about list functions also helped because I was stuck on my assignment until I remembered that he mentioned lists have built in functions and was able to figure out my problem after rewatching the lecture.

```
#!/usr/bin/env python3
# Declare my variables
strChoice = '' # User input
lstRow = [] # list of data
strFile = 'HomeInventory.txt' # data storage file
objFile = None # file handle

# Get user Input
while(True):
    print("Write or Read file data, then type 'Exit' to quit!")
    strChoice = input("Choose to [W]rite or [R]ead data: ")

    # Process the data
    if (strChoice.lower() == 'exit'): break
    elif (strChoice.lower() == 'w'):
        # List to File
        objFile = open(strFile, "w")
        lstRow = ["Item", "Value"]
        objFile.write(lstRow[0] + "," + lstRow[1] + "\n")
        lstRow = ["Lamp", "$30"]
        objFile.write(lstRow[0] + "," + lstRow[1] + "\n")
        lstRow = ["End Table", "$60"]
        objFile.write(lstRow[0] + "," + lstRow[1] + "\n")
    elif (strChoice.lower() == 'r'):
        # File to List
        objFile = open(strFile, "r")
        for row in objFile:
            lstRow = row.split(",")
            print(lstRow[0] + "|" + lstRow[1].strip())
        objFile.close()
    else:
        print('Please choose either W or R!')
```

Figure 1: Lab 5-1 in PyCharm where we wrote and read data to and from a file using lists

Dictionaries

After we learned about lists, Randall introduced us to dictionaries. Dictionaries share some similar things with lists such as they are both mutable and can be nested. However, we learned that dictionary elements are accessed by their key and list elements are accessed by index or position of it in that list.

```
# Declare my variables
strChoice = '' # User input
dicRow = {} # list of data
strFile = 'HomeInventory.txt' # data storage file
objFile = None # file handle

# Get user Input
while(True):
    print("Write or Read file data, then type 'Exit' to quit!")
    strChoice = input("Choose to [W]rite or [R]ead data: ")

    # Process the data
    if (strChoice.lower() == 'exit'): break
    elif (strChoice.lower() == 'w'):
        # List to File
        objFile = open(strFile, "w")
        dicRow = {"Item": "Lamp", "Value": "$30"}
        objFile.write(dicRow["Item"] + ',' + dicRow["Value"] + '\n')
        dicRow = {"Item": "End Table", "Value": "$60"}
        objFile.write(dicRow["Item"] + ',' + dicRow["Value"] + '\n')
        objFile.close()
    elif (strChoice.lower() == 'r'):
        # File to List
        objFile = open(strFile, "r")
        for row in objFile:
            dicRow = row.split(",") # Returns a list!
            print(dicRow[0] + '|' + dicRow[1].strip())
        objFile.close()
    else:
        print('Please choose either W or R!')
```

Figure 2: Lab 5-2 in PyCharm where we wrote and read data to and from a file using dictionaries

Steps I Took

My first step was to copy over the Assignment05 starter script so I could add to it. The starter script already had variables declared so I didn't have to do that. The first thing I did after copying over the starter script was write code for the processing part of the script. I decided to try out the error handling that we learned about for the processing data part because I realized the user might not have the text file that was specified to read data from. I used a try-except so that if they did have the specific text file we wanted then it would read the data in it and if they didn't have the file it would tell them that one wasn't found but we'll make one for them. For the except part, I found it to be best if I just opened a file in "w" mode and closed it so a new one would be made.

```
try:
    objFile = open("ToDoList.txt", "r")
    for row in objFile:
        lstRow = row.split(",")
        dicRow = {"Task": lstRow[0].strip(), "Priority": lstRow[1].strip()}
        lstTable += [dicRow]
    objFile.close()
    print("Task|Priority")
    for row in lstTable:
        print(row["Task"] + '|' + row["Priority"].strip())
    print("^Current Data listed above^")

except:
    print('''
    ToDoList.exe file currently not found
    One has now been created for you
    ''')
    objFile = open("ToDoList.txt", "w")
    objFile.close()
```

Figure 3: Processing Data part of my script in PyCharm

After the processing data part of the script, I had to add code to the 5 menu options. The code for option 1 (show data) was relatively easy since it is something we've done before. The code for that option was

```
print("Current data is: ")

for row in lstTable:

    print(row["Task"] + '|' + row["Priority"].strip())

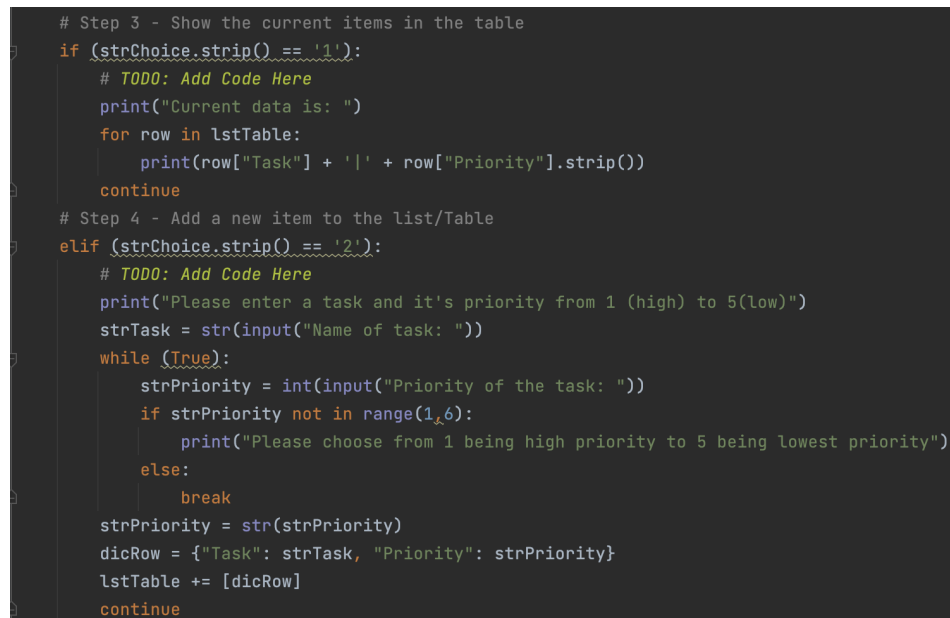
Continue.
```

The 2nd option in the menu was to add an item to the list. This part I used some of the code from last week's assignment and added in some minor changes.

```
if strPriority not in range(1,6):

    print("Please choose from 1 being high priority to 5 being lowest priority")
else:
```

Break. I decided to use not in range so that I could remind users to only choose from 1-5 if they pick something else. I felt like this kept my code short and clean.

A screenshot of a PyCharm code editor with a dark background. The code is written in Python and implements two menu options. Option 1 (labeled '1') prints the current data from a list 'lstTable'. Option 2 (labeled '2') prompts the user to enter a task and its priority, validates the priority against a range of 1 to 5, and then adds the new task as a dictionary to the 'lstTable' list. The code includes comments like '# TODO: Add Code Here' and uses 'continue' and 'break' statements to manage the flow of execution.

```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    # TODO: Add Code Here
    print("Current data is: ")
    for row in lstTable:
        print(row["Task"] + '|' + row["Priority"].strip())
    continue

# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    # TODO: Add Code Here
    print("Please enter a task and it's priority from 1 (high) to 5(low)")
    strTask = str(input("Name of task: "))
    while (True):
        strPriority = int(input("Priority of the task: "))
        if strPriority not in range(1,6):
            print("Please choose from 1 being high priority to 5 being lowest priority")
        else:
            break
    strPriority = str(strPriority)
    dicRow = {"Task": strTask, "Priority": strPriority}
    lstTable += [dicRow]
    continue
```

Figure 4: Code for option 1 and 2 of the menu in PyCharm

The code for the 3rd menu option was a little tricky as I did not know how to remove an item. I had to go back to last week's notes and lecture to find the `list.remove` function and I used that in my code.

```
if row["Task"].lower() == strTask.lower():  
    lstTable.remove(row)  
  
    print("Task has now been removed!")  
  
    Break
```

The 4th and 5th option of the menu were easier than the 3rd since I used what we did in labs 5-1 and 5-2 for writing the data to a file and exiting the program was something we've done a lot already. I also decided to add an else part to the whole loop in case the user didn't choose from options 1-5 and had put something else.

A screenshot of a PyCharm code editor with a dark theme. The code is written in Python and implements three menu options. Option 3 (elif strChoice.strip() == '3') prompts the user for a task name and removes it from a list table. Option 4 (elif strChoice.strip() == '4') saves the current list table to a file named 'ToDoList.txt'. Option 5 (elif strChoice.strip() == '5') prints a thank you message and breaks the loop. A final else clause prompts the user to choose options 1 to 5. The code includes comments for each step and a 'TODO: Add Code Here' placeholder for each option.

```
# Step 5 - Remove a new item from the list/Table  
elif (strChoice.strip() == '3'):  
    # TODO: Add Code Here  
    strTask = str(input("Name of the task you'd like removed: "))  
    for row in lstTable:  
        if row["Task"].lower() == strTask.lower():  
            lstTable.remove(row)  
            print("Task has now been removed!")  
            break  
        else:  
            print("Task not found.")  
    continue  
  
# Step 6 - Save tasks to the ToDoToDoList.txt file  
elif (strChoice.strip() == '4'):  
    # TODO: Add Code Here  
    objFile = open("ToDoList.txt", "w")  
    for row in lstTable:  
        objFile.write(str(row["Task"]) + "," + str(row["Priority"]) + "\n")  
    objFile.close()  
    print("Your data has been saved!")  
    continue  
  
# Step 7 - Exit program  
elif (strChoice.strip() == '5'):  
    # TODO: Add Code Here  
    print("\tThank you for your time!!!")  
    break # and Exit the program  
else:  
    print("Please choose options 1 to 5")
```

Figure 5: Code for menu options 3,4, and 5 in PyCharm

```
ToDoList.exe file currently not found
One has now been created for you

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
Current data is:

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
Please enter a task and it's priority from 1 (high) to 5(low)
Name of task: dishes
Priority of the task: 1

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
Your data has been saved!

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] - 5

Thank you for your time!!!
felixchen@Felixs-MacBook-Pro ~ %
```

Figure: 6: my code working in terminal

```
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
Please enter a task and it's priority from 1 (high) to 5(low)
Name of task: homework
Priority of the task: 1

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

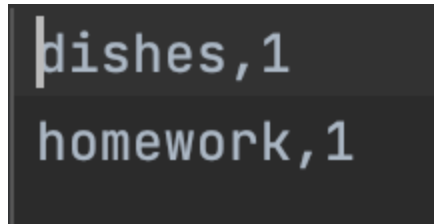
Which option would you like to perform? [1 to 5] - 4
Your data has been saved!

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] - 5

Thank you for your time!!!
```

Figure 7: my code working in PyCharm



```
dishes, 1  
homework, 1
```

Figure 8: text file containing data from my code

Summary

I was confused reading this week's assignment at first but after some clarification from Kelly and rewatching this week's lecture, I was able to tackle this assignment. Rewatching the lecture and going over lists and dictionaries was very helpful in my understanding of both of them and helped me with this week's assignment. Some parts of my code took a bit to troubleshoot like how I had to change from append to write when saving data to file or else it would keep adding the full list including previous data and not just what was added by the user. Randall was right that this assignment was different enough from last week to be challenging. Also, glad he talked about github in the end as that is something I have heard of but never really went further into looking it up and learning about github.