Sam Ewald

Foundation of Computer Programming: Python

Assignment 5

8/10/23

Chore Priority Dictionary

Summary

In this week's module, we learned about using python dictionaries to read and save user data. We also used GitHub to version and save our data to the cloud.

Reading Data into a Python Dictionary

I began by creating a for loop to read some comma separated data into a python dictionary. The loop first reads the data into a list, and then assigns the list items to the python definitions I created by substring value:

```
# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = "" # A Capture the user option selection


# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)
# TODO: Add Code Here
objFile = open(objFile, "r")
# row = objFile.readline()
for row in objFile:
    lstRow = row.split(",")
    print(lstRow)
    dicRow = {"item": lstRow[0], "value": lstRow[1]}
    print(dicRow)

objFile.close()
```

Show/write data

To show the data, I left in an extra example of what printing the row in its raw format versus calling the individual row items properly would look like. Both of these are wrapped in a for loop that runs through each row in the table.

```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    # TODO: Add Code Here
    print(lstTable)
    for dicRow in lstTable:
        print(dicRow)
        print(dicRow["toDoItem"] + ", " + dicRow["priority"])
    continue
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    # TODO: Add Code Here
    strItem = input("Enter the name of a chore: ")
    strPrior = input("What priority is this chore? ")
    dicRow = {"toDoItem": strItem, "priority": strPrior}
    lstTable.append(dicRow)
    continue
```

Remove row data

Removing row data was more difficult—the example I found in the book proved inadequate for our purposes. The example Randal gave us used the values() function to create a link to the contents of toDoItem and priority variables, and then compared the user input of remChore toDoItem. It seems like it would be possible to create an elif statement to remove any “High” or “Low” priority items.

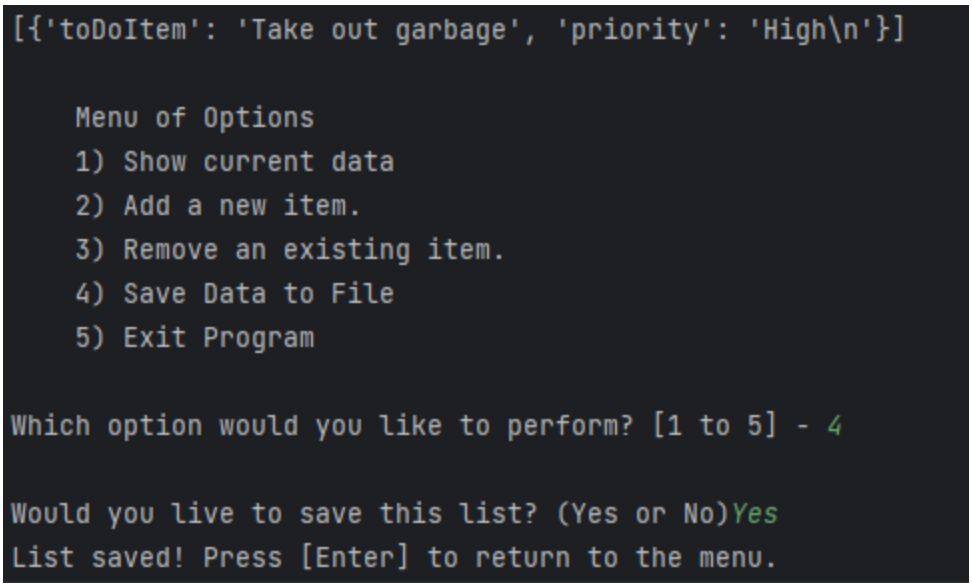
```
    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        # TODO: Add Code Here
        remChore = input("Which chore do you want to remove from the list? ")
        binItemRemoved = False
        for dicRow in lstTable:
            toDoItem, priority = dict(dicRow).values()
            if toDoItem == remChore:
                lstTable.remove(dicRow)
                binItemRemoved = True
                print("\nChore complete!")
            else:
                print("\nThat's isn't on your list.")
        continue
```

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

Which chore do you want to remove from the list? Mow Lawn

Chore complete!

    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] -
```

Saving data

Technically, the save function writes an entirely set of rows from the dictionary instead of interpreting the existing files or adding any new rows. It writes the values of the dictionary definitions, but not the definitions themselves:

```
# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    # TODO: Add Code Here
    if "yes" == str(input("Would you live to save this list? (Yes or No)")).strip().lower():
        objFile = open(objFileName, "w")
        for dicRow in lstTable:
            objFile.write(dicRow["toDoItem"] + "," + dicRow["priority"] + "\n")
        objFile.close()
        input("List saved! Press [Enter] to return to the menu.")
    else:
        input("Data was not saved. Press [Enter] to return to the menu.")
    continue
```



Conclusion

Finally, I created a repository and pushed my assignment to GitHub, which I can't take a picture of because I need to include this file in the push!