Zach DeVries
Feb 15, 2023
IT FDN 110 A
Assignment 05

Lists and Dictionaries

Introduction:

As part of this week's module, I worked with lists and dictionaries to finish off a to Do List program provided by the professor.

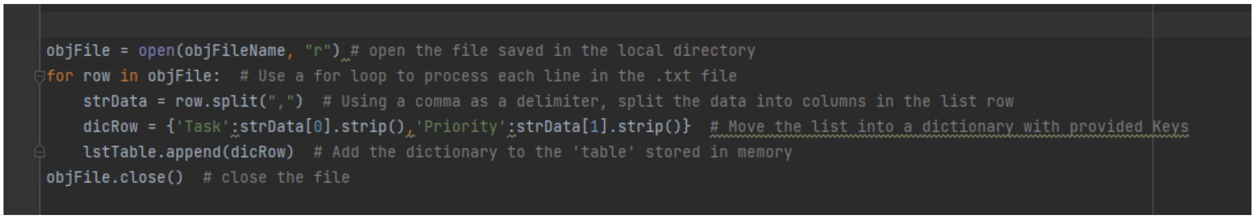
Program:

In this week's lessons, the project roadmap was already laid out in the starter file I was provided. I looked over the header information, read through the file and analyzed the 6 steps that were stubbed out. After reviewing the doc initially, I then took note of the variables that had been declared in the header and set off to code step 1.

Step 1

Step 1 was to open a file and import the data into a table. I started by using information learned in previous lessons and added code to open, read, then close the file.Then because I knew the format of the file and the delimiter being used, I began extracting the data by looking at each row of data and pulling out the data and storing it in a dictionary with pre-defined keys identified in the program prompt. Once the dictionary row was completed, I appended it to a list which would act as my way to store and manage the collection of data that had been moved from the disk to my local memory.

Figure 1: Retrieve Data From File Stored on Hard Drive



```
objFile = open(objFileName, "r") # open the file saved in the local directory
for row in objFile:  # Use a for loop to process each line in the .txt file
    strData = row.split(",")  # Using a comma as a delimiter, split the data into columns in the list row
    dicRow = {'Task':strData[0].strip(),'Priority':strData[1].strip()}  # Move the list into a dictionary with provided Keys
    lstTable.append(dicRow)  # Add the dictionary to the 'table' stored in memory
objFile.close()  # close the file
```

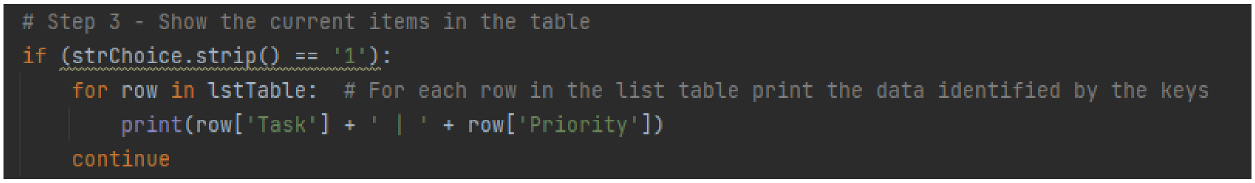
Step 2

Step 2, which displays a series of choices to the user, was programmed ahead of time. I reviewed the code and jumped straight into step 3.

Step 3

In step 3, I was given direction to show the current list of items to the user. I worked out that for each row, I wanted to print both Task and Priority. I achieved this by using a for loop that runs as many iterations as there are rows in the table. Then using concepts from a previous module, I added the print command with some key based indexing to output the data.

Figure 2: Print Current Table Data to user

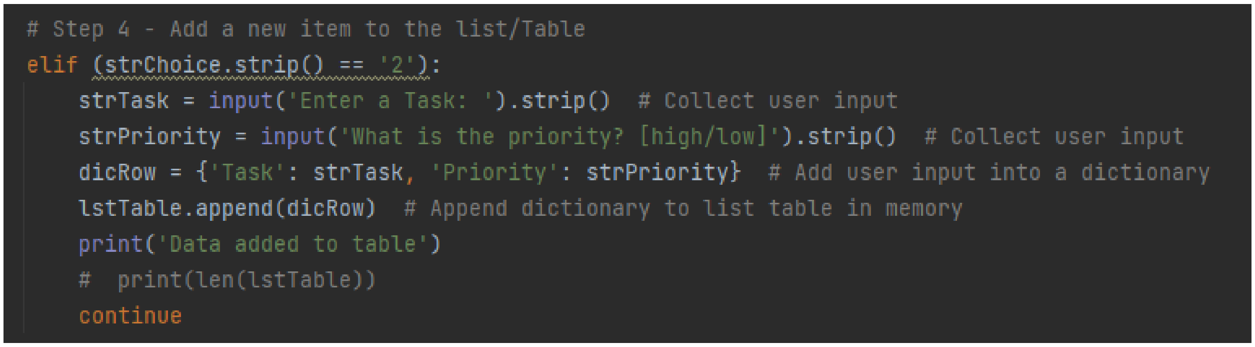


```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    for row in lstTable:  # For each row in the list table print the data identified by the keys
        print(row['Task'] + ' | ' + row['Priority'])
    continue
```

Step 4

In step 4 I worked out how to add a new item to the list. First I used input commands to collect the data the user would like to add. Then I added that data to a dictionary row. After that, I appended the row to the table, much like I did back in step 1. After the theavy lifting was done, I added a message to let the user know their data was added.

Figure 3: Add data to the table

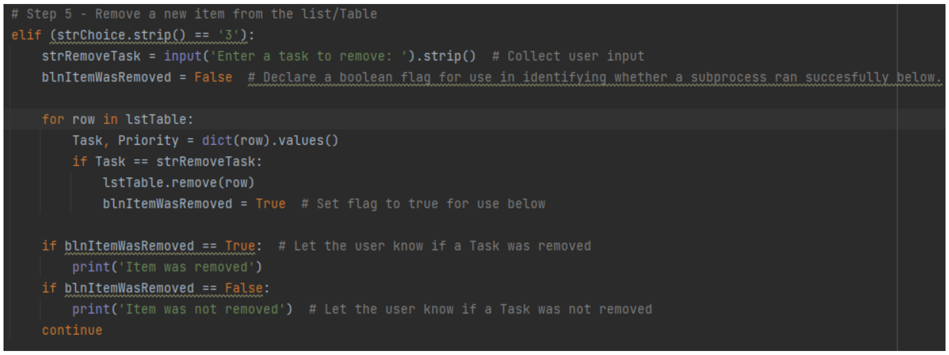


```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    strTask = input('Enter a Task: ').strip()  # Collect user input
    strPriority = input('What is the priority? [high/low]').strip()  # Collect user input
    dicRow = {'Task': strTask, 'Priority': strPriority}  # Add user input into a dictionary
    lstTable.append(dicRow)  # Append dictionary to list table in memory
    print('Data added to table')
    #  print(len(lstTable))
    continue
```

Step 5

Step 5 was the most difficult. In this step I had to develop a way to remove data. I started prompting the user to provide the Task they wished to remove. Given more time, I might go back and use more formatting constraints in user input variables such as .lower() or .strip() to ensure the intent of the user is being is not being lost through semantics. I then set upon a method for removing data. To evaluate each row in the table on an individual basis, I started with a for loop.I then parsed out the data from the two columns into variables for the row in question, then compared the Task to the task identified for removal. If the two matched I removed the whole row using the nifty python remove command. Next to aler the user as to whether the item had been removed or not, I employed a boolean flag that would be set True, if the equality condition stated above was met. With an IF statement outside the for loop, I then created outputs based on boolean flag state.

Figure 4: Remove Data From Table



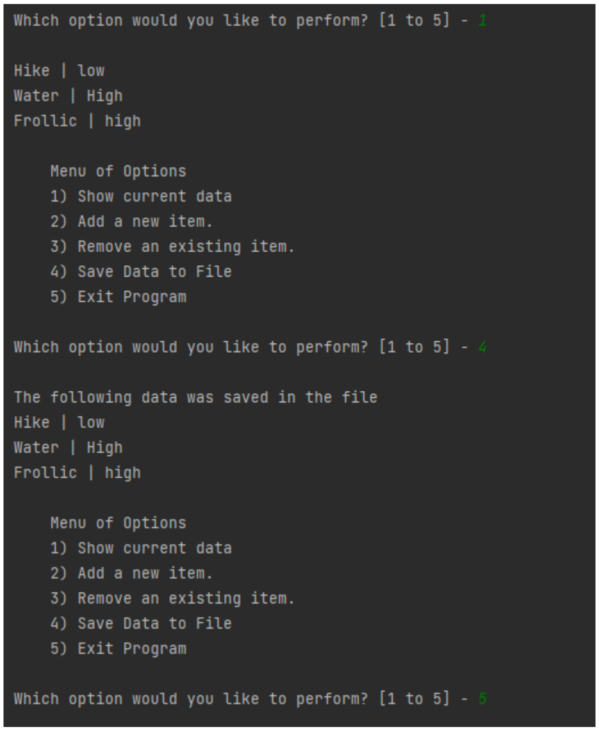
```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    strRemoveTask = input('Enter a task to remove: ').strip()  # Collect user input
    blnItemWasRemoved = False  # Declare a boolean flag for use in identifying whether a subprocess ran succesfully below.

    for row in lstTable:
        Task, Priority = dict(row).values()
        if Task == strRemoveTask:
            lstTable.remove(row)
            blnItemWasRemoved = True  # Set flag to true for use below

    if blnItemWasRemoved == True:  # Let the user know if a Task was removed
        print('Item was removed')
    if blnItemWasRemoved == False:
        print('Item was not removed')  # Let the user know if a Task was not removed
    continue
```

Step 6
Step 6 was the final task of saving the data back to the file. In this step, I opened the file again and used the familiar for-loop and write commands to write the data back to the file. I then closed up the file and drafted an output message to let the user know their data was saved.

Step 7
Step 7 which is the option to exit the program was complete prior to handoff.

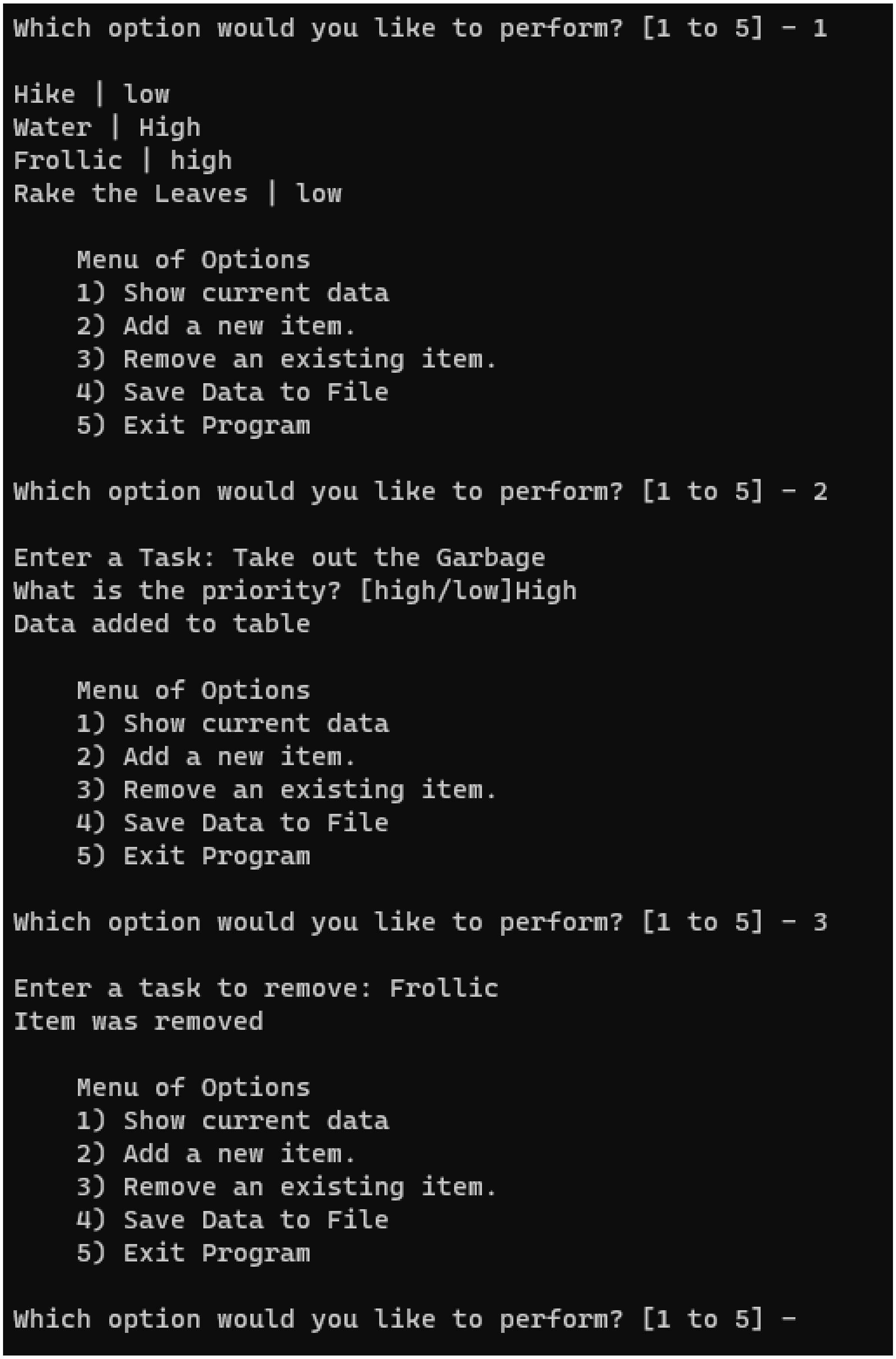
Conclusion:
In this module, I edited an existing pycharm and worked with lists/dictionaries in more detail to pull data from a file, edit the data based on user inputs, and then save the data back to the file. The files are saved on Github here: [zdevries123/IntroToProg-Python (github.com)](https://github.com/zdevries123/IntroToProg-Python)

Figure 5: Program Execution in PyCharm



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

Hike | low
Water | High
Frollic | 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] - 4

The following data was saved in the file
Hike | low
Water | High
Frollic | 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] - 5
```

Figure 6: Program Execution in Console



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

Hike | low
Water | High
Frollic | high
Rake the Leaves | low

    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

Enter a Task: Take out the Garbage
What is the priority? [high/low]High
Data added to 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] - 3

Enter a task to remove: Frollic
Item was removed

    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] -
```