Name: Charles Lin
Date: February 12, 2023
Course: IT FDN 110 A

Assignment 05 Lists and Dictionaries

Introduction

I learned about collections of data in lists and dictionaries. We take the data from a table or file into memory as lists or dictionaries, and process and manipulate in-memory on the lists or dictionaries, then write the data back to the table or file if we want to keep the data.

Lists

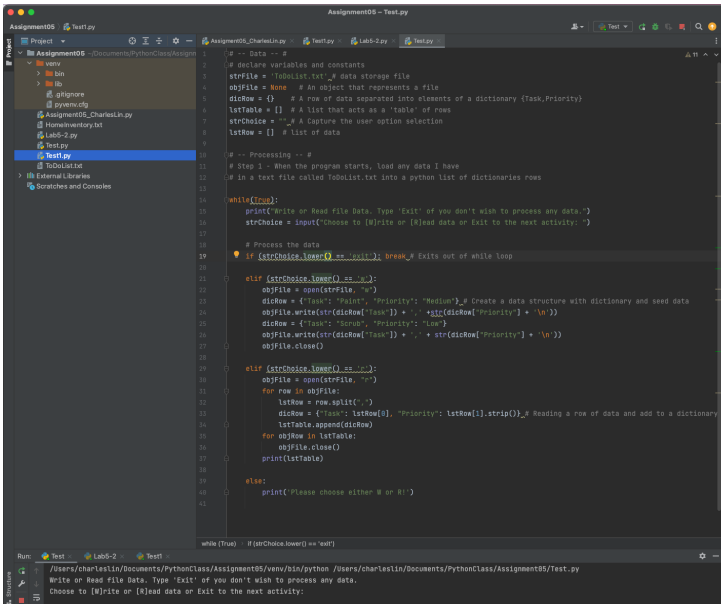
Lists hold a collection of objects, and the data is identified through index of 0, 1, 2, etc. We can think of it as column a, b, c, etc on a spreadsheet. We can read data from a file into a list, and we can write data to a file from a list. We use brackets [ ] to indicate a list.

Dictionaries

Dictionaries are very similar as sequences seen in List, but in dictionaries, we use key (character) subscripts instead of index subscripts (numbers). We use the braces { } to indicate a dictionary.

Assignment with Starting Template

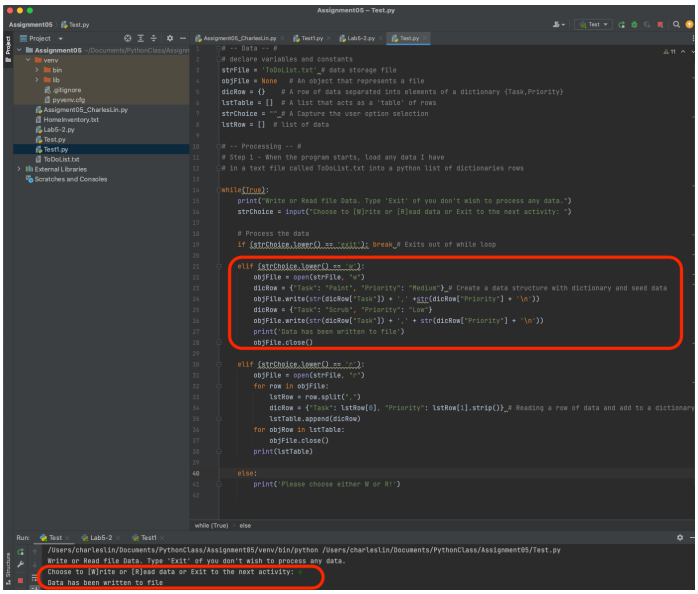
While having a template and pseudo code introduces the foundation, the instructions sometimes appear confusing to me. For example, Step 1's processing mentioned Lab 5-2 that comes with its own starter code, and on this template there is also a short instruction. I had to clarify the scenario of when the text file is not present on Canvas' Discussion Board. I would include a text file ToDoList.txt in the assignment, to be placed in the working folder. I also made the assumption that it takes 'Exit' from Step 1 in order to get to the Menu of Options in the subsequent step.



Listing 1 Processing Data

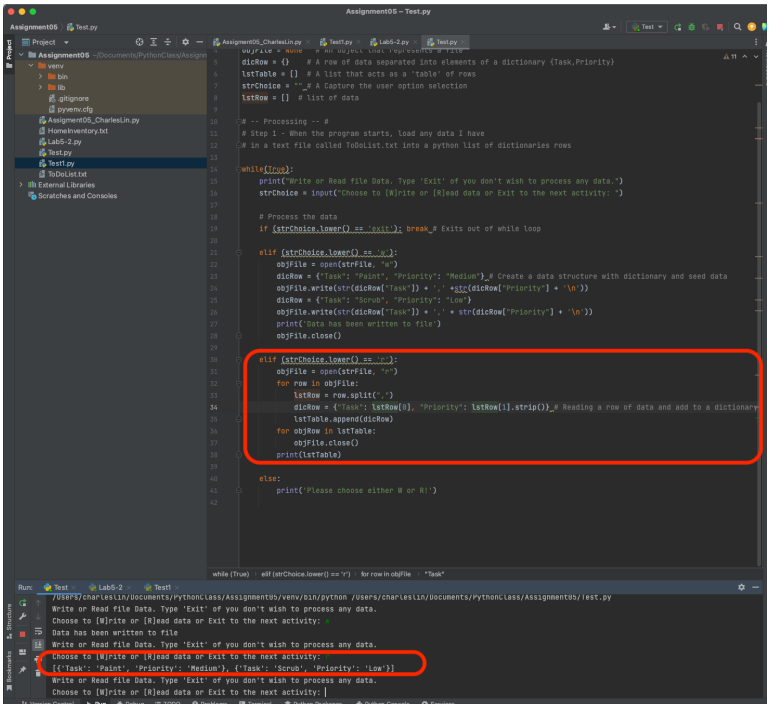
I observed Lab 5-2's starter code and altered it to a dictionary instead of a list. I ran this portion of the code on a scratch pad to see how it behaves. I tested the scenarios with write, read, and exit. ToDoList.txt needs to be present in the working folder, otherwise the step would error out.

- If I select "write", the 2 rows stated in the code itself would be populated on ToDoList.txt.



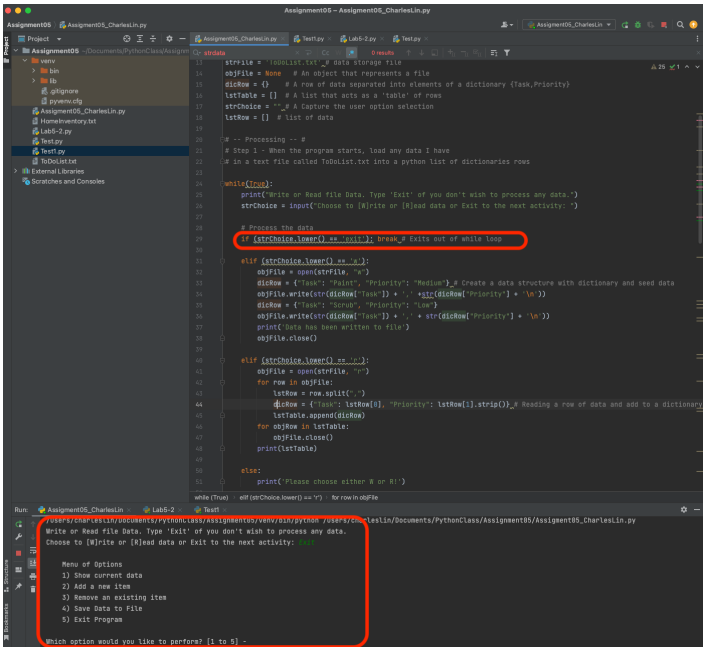
Listing 2 Write Scenario

- If I select 'read', the content from existing file will show in a dictionary string



Listing 3 Read Scenario

- If I select ‘Exit’, it would go to the next activity that shows the Menu of Options. If I select ‘Exit’ prior to ‘read’ or ‘write’, an empty lstTable would be presented in the next step.

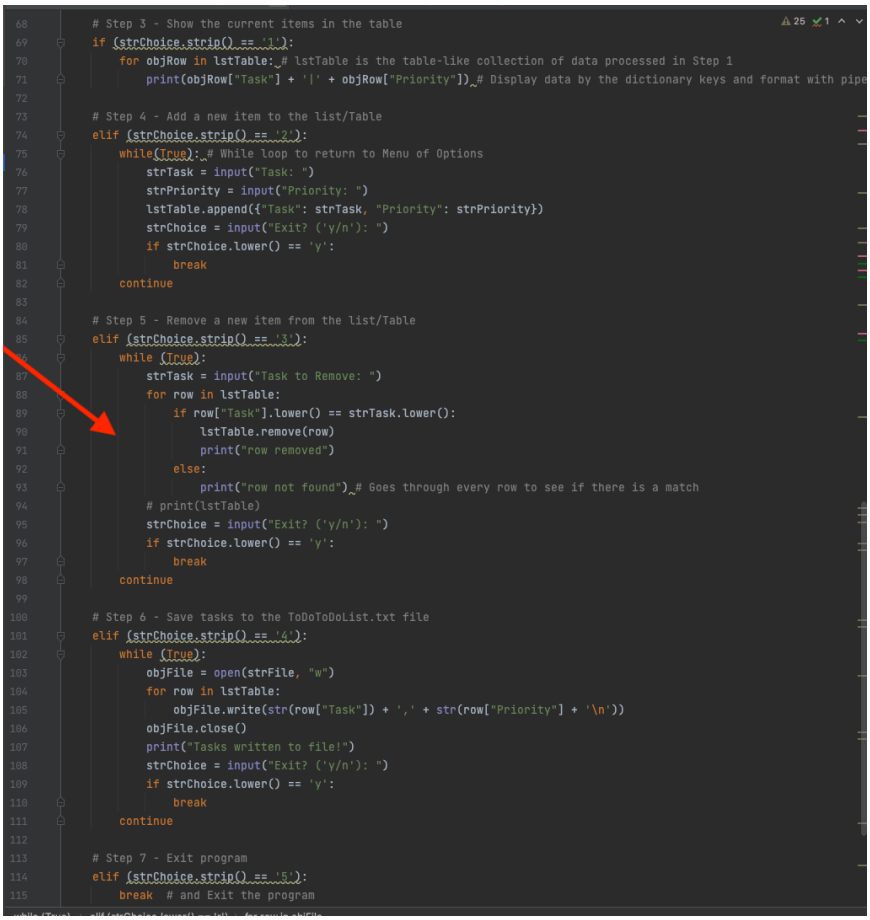


Listing 4 Exit Scenario

I followed the template and referenced Weekly Live QA Session’s example scripts. Initially for Step 3, 4, and 5, I opened ToDoList.txt each time and create a dictionary, then it dawned on me

that I should create a list/Table from Step 1, then work with that list/Table in the subsequent step, then save what has been added or removed back to the file.

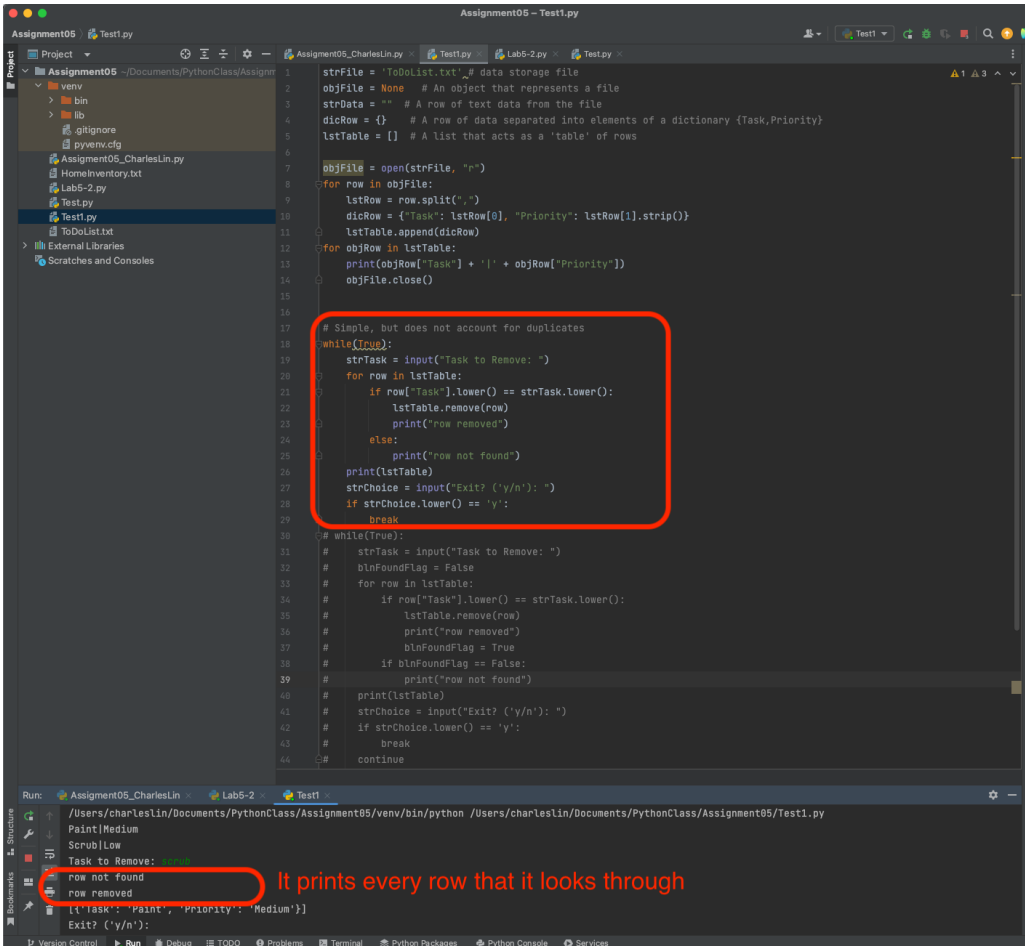
Earlier on I wasn't consistently incorporating while(True) loop to allow the user to continue adding or removing tasks. I modified Steps 3, 4, and 5 to accommodate that. My break and continue commands also weren't consistently bringing user to the Menu of Option at the right step. I also got indentation errors. After running into all sorts of issues, I began running small piece at a time, sometimes by commenting out the rest, and sometimes by taking a portion of the code to scratch pad. I appreciate the faint indentation line PyCharm presents to help me organize the code.



```
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        for objRow in lstTable:_# lstTable is the table-like collection of data processed in Step 1
            print(objRow["Task"] + '|' + objRow["Priority"])_# Display data by the dictionary keys and format with pipe

    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        while(True):_# While loop to return to Menu of Options
            strTask = input("Task: ")
            strPriority = input("Priority: ")
            lstTable.append({"Task": strTask, "Priority": strPriority})
            strChoice = input("Exit? ('y/n'): ")
            if strChoice.lower() == 'y':
                break
        continue

    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        while (True):
            strTask = input("Task to Remove: ")
            for row in lstTable:
                if row["Task"].lower() == strTask.lower():
                    lstTable.remove(row)
                    print("row removed")
                else:
                    print("row not found")_# Goes through every row to see if there is a match
            # print(lstTable)
            strChoice = input("Exit? ('y/n'): ")
            if strChoice.lower() == 'y':
                break
        continue

    # Step 6 - Save tasks to the ToDoToDoList.txt file
    elif (strChoice.strip() == '4'):
        while (True):
            objFile = open(strFile, "w")
            for row in lstTable:
                objFile.write(str(row["Task"]) + ',' + str(row["Priority"] + '\n'))
            objFile.close()
            print("Tasks written to file!")
            strChoice = input("Exit? ('y/n'): ")
            if strChoice.lower() == 'y':
                break
        continue

    # Step 7 - Exit program
    elif (strChoice.strip() == '5'):
        break  # and Exit the program
```

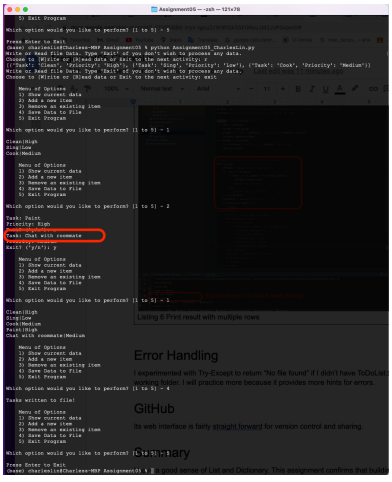
Listing 5 Indentation lines

I struggled with Step 5, removing of a task. I would have liked to have the message that tells me 1 line of "row removed" or "row not found", but I only managed to print every row it goes through until it found a match to remove. At the QA Session that was what the debug mode showed. I would like to figure out how to show just 1 row of either finding a match or not. I experimented with the Boolean Flag, but it didn't give me what I wanted. The simple one that breaks after it found the first occurrence made most sense to me, so I applied that.



Listing 6 Print result with multiple rows

I ran it on Terminal to verify it behaves the same as PyCharm. I also tested a task with spaces in between to make sure it works ok.



Listing 7 Run it on Terminal

Error Handling

I experimented with Try-Except to return “No file found” if I didn’t have ToDoList.txt in the working folder. I will practice more because it provides more hints for errors.

GitHub

Its web interface is fairly straight forward for version control and sharing.

Summary

I have a good sense of List and Dictionary. This assignment confirms that building codes in increments and good organization helps with my thought process, and later on peer review. I’ve become more comfortable with “for” and “while” loops, and seeing the outcome of each stage with placing “print” statement at the right place. While existing template and starter codes are helpful, sometimes they are not laid out in a way I intuitively understand. It’s worthwhile to truly understand what it’s doing, and maybe adjust it to what makes sense to me, so I easily know what the code does.