Saglara Orellana
May 14, 2022
Foundation of Programming: Python
Assignment 05
https://github.com/saglarao/IntroToProg-Python

Creating To Do List

Introduction

This script starts with loading data from the text file ToDoList as a list of python dictionaries. Then it proceeds to show menu options to the user and depending on user choice, the program displays the data, adds or removes a task, then the user has an option to save all data to the same text file.

Starting the script

Since this script was already created, I started with a test run to make sure that the *while* loop works properly. The first thing I noticed, that my script needs to notify the user, when they enter a different character from the menu options. I added else statement at the end of the script to display a message to the user, that they can only choose 1 to 5.

Then I proceed to start adding code under step 1 of the already written pseudo-code. I need to load the data from the file into the program as a list of dictionary. The text file was already defined as *objFile* in variables list at the beginning of the script, I added file handle *objF* and assigned it value of *None*. I started the code of step 1 with opening the text file to read. Then I need the program to go through each row of *strData* with the *for* loop: turn it into a list, then create a python dictionary *dicRow* with key subscripts *"Task"* and *"Priority"* and add it to the *lstTable* (a list that acts as a table of rows). All the variables were already introduced at the beginning of the script with a definition as a comment. Lastly, the *objF* needs to be closed.

When I tested the newly added code, the program gave me an error, since there was no existing file "ToDoList.txt". I added try-except construct to show a user friendly error message and also for the program to continue to run next statement.

Here is the code for step 1:

```
# -- Processing -- #
# Step 1 - When the program starts, load the data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)
try:
    objF = open(objFile, "r")
    for row in objF:
        strData = row.split(" - ")  # returns a list
        dicRow = {"Task": strData[0], "Priority": strData[1].strip()}
        lstTable.append(dicRow)
    objF.close()
except:
    print('File not found, new file will be created, when you save.’)
```

Displaying, adding and removing items

Next I need to add code to step 3 to show current data. First, I added the message to display “Current To Do List [Task - Priority]:”. Current data is being held in lstTable, I used the *for* loop to go through each element of *lstTable* and print them. Since *lstTable* contains dictionaries, I am using key subscripts “Task” and “Priority”. I chose to use a hyphen as a separator.

```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    print('Current To Do List [Task - Priority]:')
    for row in lstTable:
        print(row["Task"], row["Priority"], sep=" - ")
    continue
```

When the user chooses option 2 of the menu, I need to collect the user input and then add it to the table of data. I created two variables *strTask* and *strPriority*, added it to declare variables section of the script as empty strings, then assigned them *input* values in the *elif* statement, where the user chooses option 2. Once user enters new task and assigns it a priority, I added them as another dictionary to *lstTable*. This block ends with a message that new task was added.

```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    strTask = input("Please enter a task: ")
    strPriority = input("Please give it a priority: ")
    lstTable.append({"Task": strTask, "Priority": strPriority})
    print("New Task has been added!")
    continue
```

When the user wants to remove an item, the program needs to request which task to remove. I declared a new empty string variable *strRemove*, then assigned it an input value in the *elif* statement, where user choice is 3. Then I need to go through *lstTable* and compare value of *strRemove* to the tasks in dictionaries. If the value matches, then the dictionary need to be removed. The message "The task has been removed" displayed after the successful find, if not, then the program shows the message "The task was not found".

```
# Step 5 - Remove an item from the list/Table
elif (strChoice.strip() == '3'):
    strRemove = input("Enter a task to remove: ")
    for row in lstTable:
        if row["Task"].lower() == strRemove.lower():
            lstTable.remove(row)
            print("The task has been removed")
        else:
            print("The task was not found")
    continue
```

Saving Data to a File and Ending the Program

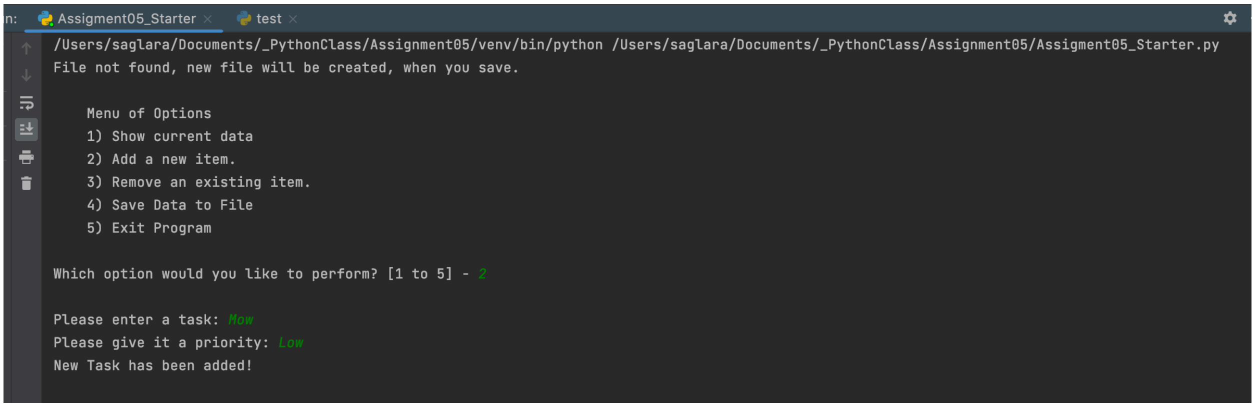
When the user chooses to save the data into a file, I need to open the file to write. I chose to write over append, in case the user already had an existing file with data, it was already loaded into *lstTable* and if I append it, then I will have a repeat information. After the file was opened or created, I unpack my list with the *for* loop and write down dictionary items with hyphen in between and adding a new line at the end. Then file can be closed and this block ends with a message to let the user know, that the data was saved to a file.

I don't need to do much for option 5 of the menu, I just added a message to display, that the program ended.

```
# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    objF = open(objFile, "w")
    for row in lstTable:
        objF.write(str(row["Task"]) + " - " + str(row["Priority"]) + "\n")
    objF.close()
    print("The data was saved!")
    continue
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    print("Program Ended")
    break  # and Exit the program
```

Testing the script

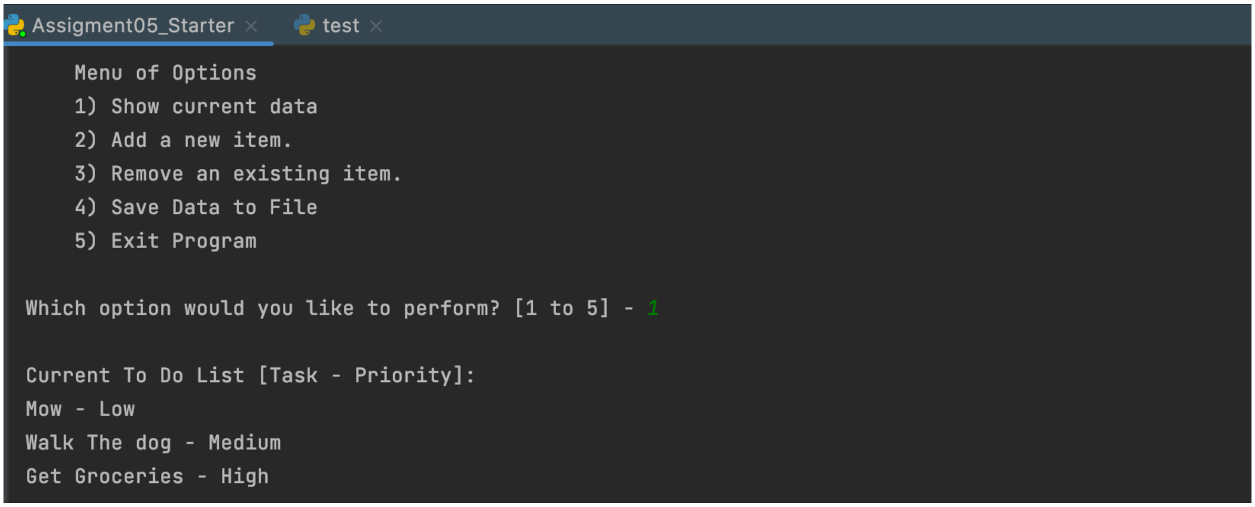
I deleted text file ToDoList, that was created during my test runs. When I start the program, the message is displayed, that it was not able to find the file. Menu options are displayed and I chose option 2 to start adding tasks.



```
/Users/saglara/Documents/_PythonClass/Assignment05/venv/bin/python /Users/saglara/Documents/_PythonClass/Assignment05/Assigment05_Starter.py
File not found, new file will be created, when you save.

    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: Mow
Please give it a priority: Low
New Task has been added!
```

Figure 01. Adding new tasks.

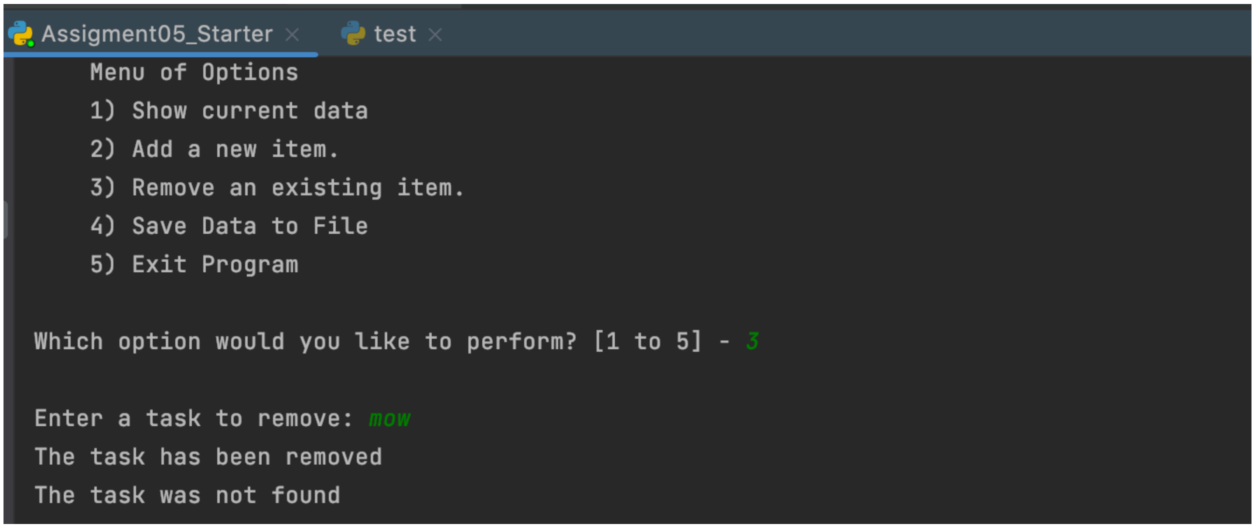
Once I enter my task and its priority, the program lets me know that it was added. I have chosen option 2 again to add more data. Once added, I can request to display it to me by choosing option 1. Figure 02 shows, that, when I entered option 1, the data I have entered previously is shown with a title “Current To Do List [Task - Priority]:”



```
     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 To Do List [Task - Priority]:
Mow - Low
Walk The dog - Medium
Get Groceries - High
```

Figure 02. Showing current data

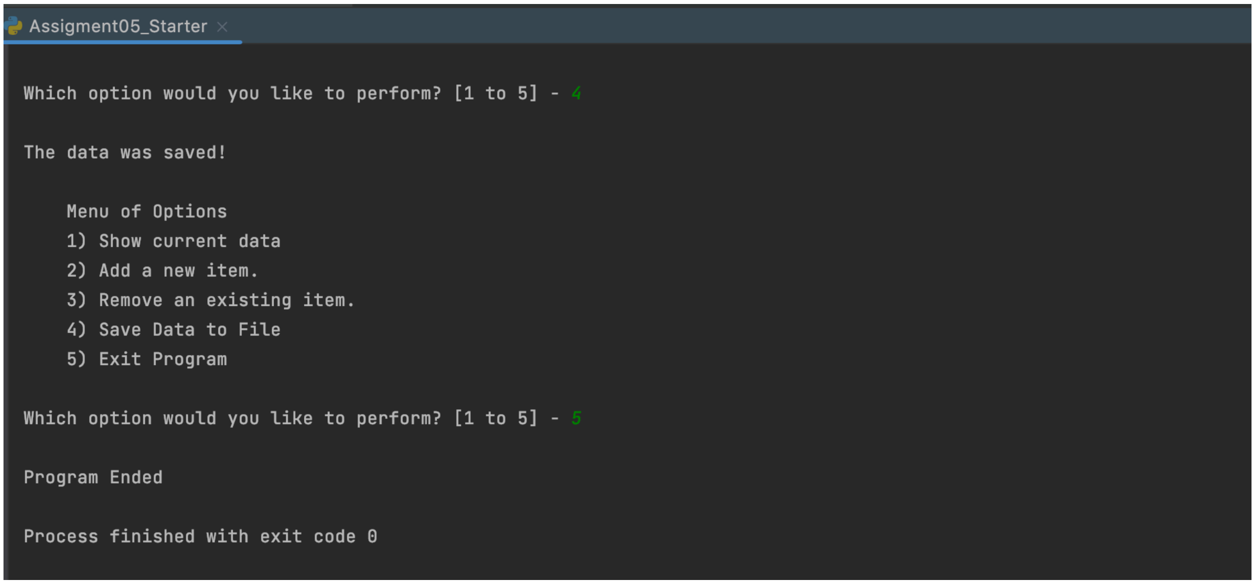
Then I want to remove a task and choose option 3 of the menu. The program requests to enter a task to be removed. Once I enter a task, I see two messages "The task has been removed" and "The task was not found". That happens, because the *for* loop goes through each element of the table and notifies each time it has found a match or not. At the current level of my Python knowledge, I was not able to solve this problem to show only one message.



```
     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: mow
The task has been removed
The task was not found
```

Figure 03. Removing an item

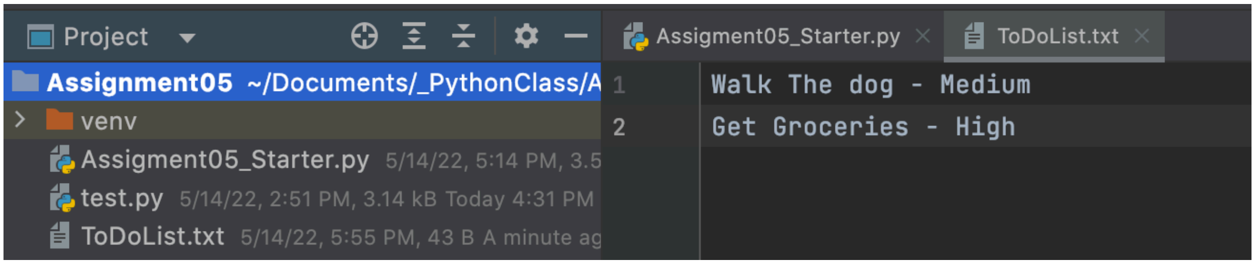
I am ready to save the data into a file and choose option 4, following by option 5 to exit the program.



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

The data was 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

Program Ended

Process finished with exit code 0
```

Figure 04. Saving data and ending the program.

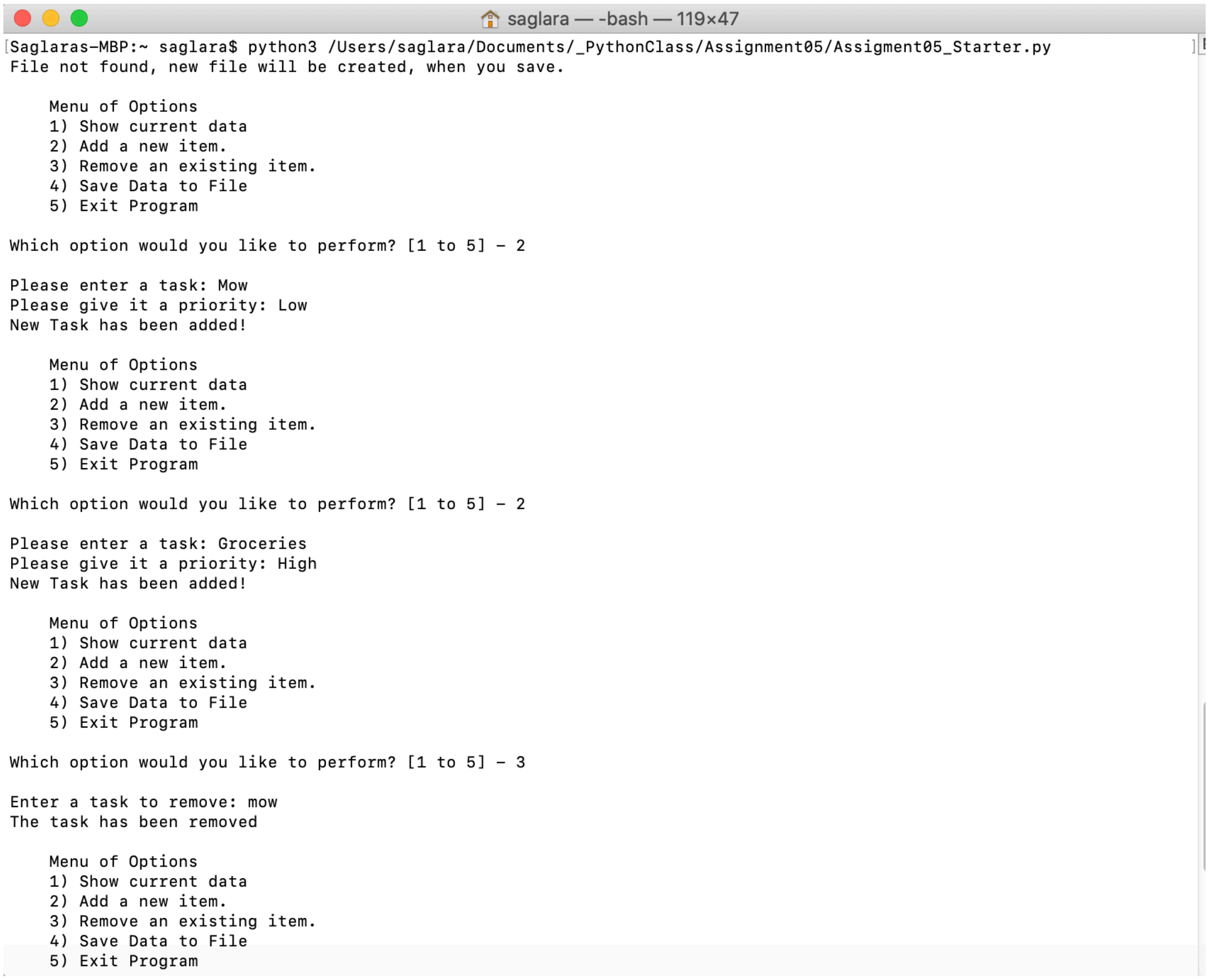
Once the script stops running, new text file ToDoList appears in the folder with my script and contains two tasks “Walk the Dog” and “Get Groceries” with their priorities.


Project
Assigment05_Starter.py
ToDoList.txt
Assignment05 ~/Documents/_PythonClass/A
venv
Assigment05_Starter.py 5/14/22, 5:14 PM, 3.5
test.py 5/14/22, 2:51 PM, 3.14 kB Today 4:31 PM
ToDoList.txt 5/14/22, 5:55 PM, 43 B A minute ag

```
Walk The dog - Medium
Get Groceries - High
```

Figure 05. Text file ToDoList.

When I run the script in command shell, although I do have ToDoList.txt created in the folder with the script, it is looking for a file in the main user folder on my Mac. I would need to add a path to the script for it to look in the assignment folder. I am not doing that for grading purposes, so a new text file with ToDoList is created in the user folder on my computer after I finish the program. Figure 06 shows the screenshot of Terminal window with the script running.



```
[Saglaras-MBP:~ saglara$ python3 /Users/saglara/Documents/_PythonClass/Assignment05/Assigment05_Starter.py
File not found, new file will be created, when you save.

    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: Mow
Please give it a priority: Low
New Task has been added!

    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: Groceries
Please give it a priority: High
New Task has been added!

    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: mow
The task has been 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
```

Figure 06. Running the script in Terminal.

Summary

When I wrote the script, I was able to load text data into a Python as a list (table) of dictionaries or, if no files exists, continue to run the program with the use of try-except construct. The user will be able to add more tasks to the list or to remove them and, when ready, to save all entries into a text file.