Name: Rabiya Abdul

Date: 08-03-2020

Course: IT FDN 110 A

Assignment #: Assignment 5

Github link: https://github.com/ruby0606/IntrotoProg-Python

Assignment 5: Dictionaries

Introduction

In this document, I wrote down the steps I took to modify the shared python script to read an existing text file, request user to choose a menu option, add a new row to the dictionary contents, remove existing row, display the stored information, and save it to the file. I watched the videos from Module 5, went through the text for Module 4 and read the fifth chapter from the book "Python Programming for Absolute Beginners". I encountered some challenges in reading the existing file and in displaying text when user tries to remove a task that does not exist.

Step 1: Load Data and display menu

I created a text file called "ToDoList.txt" with some tasks and priorities listed and separated by a comma. This code to display menu was already present in the shared python script. I did not make any changes to the menu.

Step 2: Display the current items in table

I used the for loop to display all task keys and priority keys in the python dictionary.

Step 3: Add a new item to the table

When user selects this action, I requested the user to add a new task and the relevant priority to the table and stored it as string inputs “Task” and “Priority”. I appended this information to the existing tasks and priorities.

Step 4: Remove an item from the table

When user selects this action, I requested the user to enter the name of the task they would like to remove. If a match is found, I removed the selected task and relevant priority from the table. If a match is not found, I displayed a relevant message to the user. This part was challenging as the message appeared multiple times for each match not found. I finally chose to enter a break and added an else statement to print the relevant “no match” message.

Step 5: Save to File

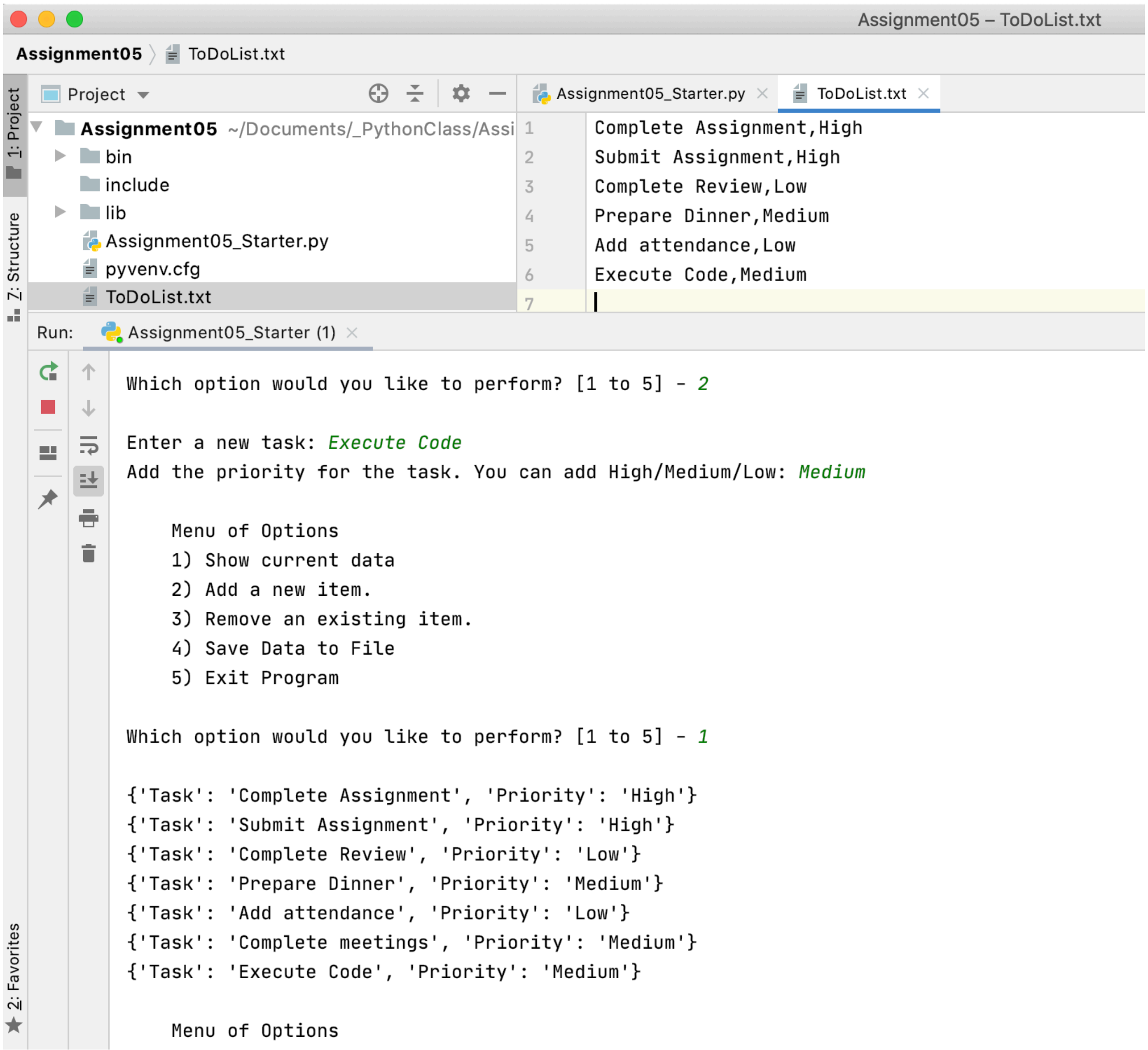
When user selects this action, I wrote these changes to the existing file and displayed a message to the user.

Step 6: Exit Program

When user selects this action, the current contents are displayed along with a message to the user about exiting the program.

Step 7: Results in pyCharm

Results in PyCharm showing the text file that is updated, data input from user to add a new row, input from user to remove an existing row, saving results to file displaying the final results in the text file.



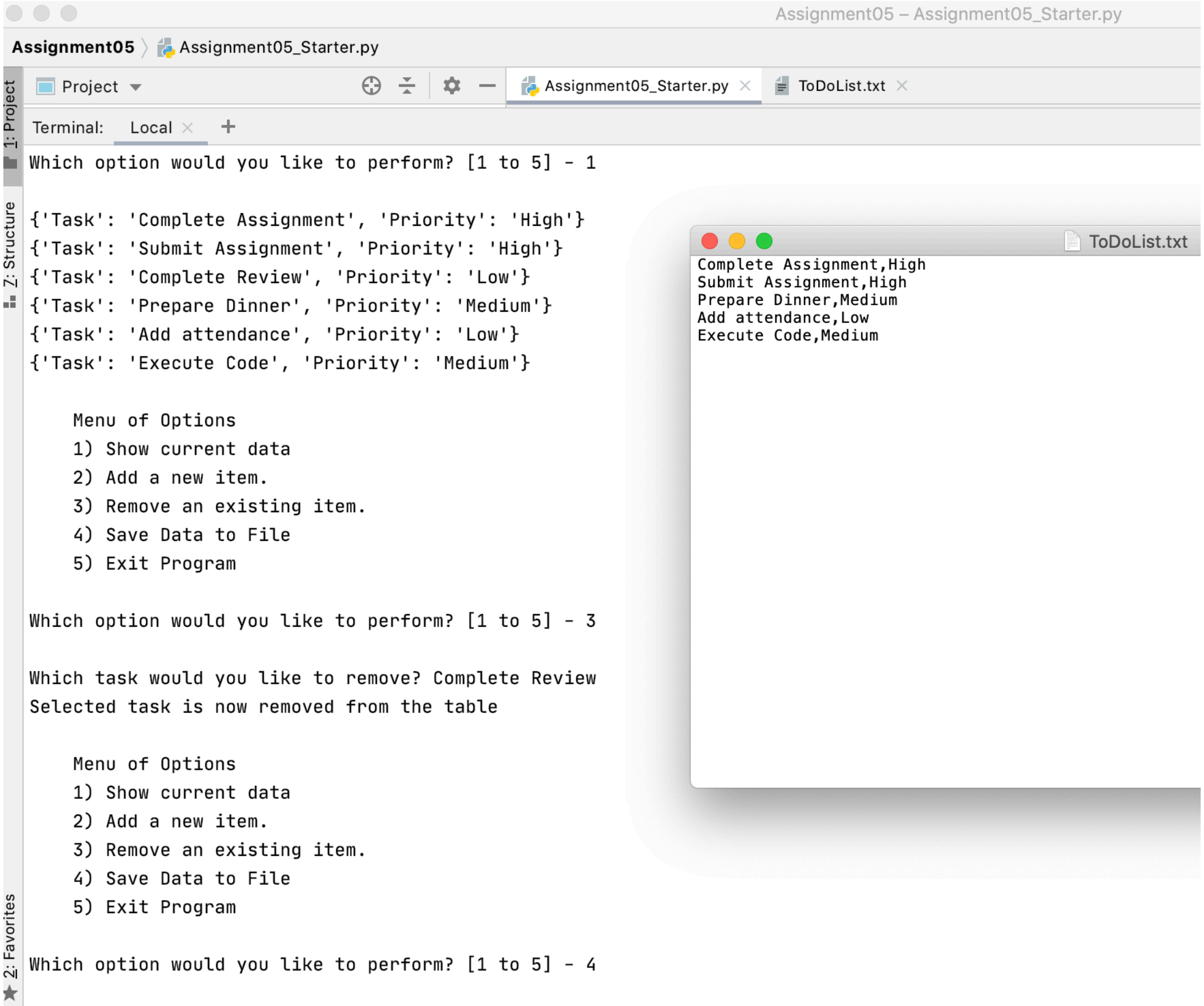
```
Complete Assignment,High
Submit Assignment,High
Complete Review,Low
Prepare Dinner,Medium
Add attendance,Low
Execute Code,Medium

```

```
Which option would you like to perform? [1 to 5] - 2

Enter a new task: Execute Code
Add the priority for the task. You can add High/Medium/Low: Medium

    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

{'Task': 'Complete Assignment', 'Priority': 'High'}
{'Task': 'Submit Assignment', 'Priority': 'High'}
{'Task': 'Complete Review', 'Priority': 'Low'}
{'Task': 'Prepare Dinner', 'Priority': 'Medium'}
{'Task': 'Add attendance', 'Priority': 'Low'}
{'Task': 'Complete meetings', 'Priority': 'Medium'}
{'Task': 'Execute Code', 'Priority': 'Medium'}

    Menu of Options
```

Fig 1: Screenshot of the result in Pycharm along with the results written to the text file.

Step 8: Results in Terminal

Results in Terminal displaying the user’s entry to remove an existing item and the corresponding results saved to the text file.



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

{'Task': 'Complete Assignment', 'Priority': 'High'}
{'Task': 'Submit Assignment', 'Priority': 'High'}
{'Task': 'Complete Review', 'Priority': 'Low'}
{'Task': 'Prepare Dinner', 'Priority': 'Medium'}
{'Task': 'Add attendance', 'Priority': 'Low'}
{'Task': 'Execute Code', 'Priority': 'Medium'}

    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

Which task would you like to remove? Complete Review
Selected task is now removed from the 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] - 4
```

ToDoList.txt

```
Complete Assignment,High
Submit Assignment,High
Prepare Dinner,Medium
Add attendance,Low
Execute Code,Medium
```

Fig 2: Screenshot of the result in Terminal along with the results written to the text file.

Step 9: Final Script that was executed

```
# ---------------------------------------------------------------------- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# Rabiya Abdul,08-10-2020,Added code to complete assignment 5
# ---------------------------------------------------------------------- #
# -- 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
strTask = "" # Adding a new task
strPriority = "" # The priority of the task
strRmvTask = "" # Removing a task

# -- 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)
objFile=open("ToDoList.txt", 'r')
for row in objFile:
    strData=row.split(",")
    dicRow={"Task": strData[0], "Priority": strData[1].strip()}
    lstTable.append(dicRow)
    print(lstTable, "List with Dictionary Objects")
objFile.close()
# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    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
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] -
"))
    print()  # adding a new line for looks
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        for dicRow in lstTable:
            print(dicRow)
        continue
    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        strTask= str(input("Enter a new task: "))
        strPriority= str(input("Add the priority for the task. You can add
High/Medium/Low: "))
        dicRow= {"Task": strTask, "Priority": strPriority}
        lstTable.append(dicRow)
        print("New task and priority added")
        continue
    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        strRmvTask= str(input("Which task would you like to remove? "))
        for dicRow in lstTable:
            if dicRow["Task"] == strRmvTask:
                lstTable.remove(dicRow)
                print("Selected task is now removed from the table")
                break
        else:
            print("Did not find a matching task.")
    # Step 6 - Save tasks to the ToDoList.txt file
    elif (strChoice.strip() == '4'):
        objFile=open("ToDoList.txt", 'w')
        for dicRow in lstTable:
            objFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")
        objFile.close()
        print("Saved to the file")
        continue
    # Step 7 - Exit program
    elif (strChoice.strip() == '5'):
        print(dicRow)
        print("Exiting the Program")
        break  # and Exit the program
```

Step 10: Github link

As per the instructions, I created a Github account and checked in my code.

Summary

I reviewed Professor Randal’s videos and executed each step several times. PyCharm was very helpful in ensuring each conditional statement’s block of code was clearly separated. Breaking down each step and executing it separately helped me complete the assignment. It was challenging to ensure all actions could be completed successfully. I also posted my work to Github.