Dmitriy Alexandrov

November 5th, 2019

IT FDN 100 A

Assignment05

Coding a To Do Task List Script

Introduction

In this paper I am going to go through the steps that I took in adding code to a starter script to create a fully functional To Do Task List script, which opens up a ToDoList.txt file (or creates it, if the files doesn't already exist) and is able to read, write and update tasks to that file using input that the user provides.

Steps Taken in Adding Code to the To Do Task List Script

I started out the process by launching PyCharm and creating a new project called Assignment05 in my _PythonClass folder on my C: drive using a virtual Python interpreter. After the project was created, I imported the starting code for the assignment called "Assignment05_Starter.py".

With the file in PyCharm, the first thing I added to the document was a change log entry in the header of the code to identify who was working on the code after the initial creation of it.

```
# ------------------------------------------------------------------------ #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoList.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
# DAlexandrov,11.5.19,Added code to complete assignment 5
# ------------------------------------------------------------------------ #
```

After taking a look at the declared variables that were already in the code, I identified what I already had to work with. The only change that I made to the already declared variables was changing the strMenu variable to lstMenu and initialize it with an empty list value. This was because I intended to use this variable for tracking what options the user has chosen over the course of using the script.

```
lstTable = []  # A dictionary that acts as a 'table' of rows
lstMenu = []   # A menu of user options
strChoice = ""# A Capture the user option selection
```

Moving onto the Processing section of the script and working through Step 1, I thought of the correct way to read the ToDoList.txt file if it exists or create it if it doesn't. I did this with a while loop and a try, except statement. The while loop is put into place so that if the file doesn't exist and an error is thrown, the user doesn't have to manually restart the script when the file is created. The try, except block tries to read the file and if it can't tell the user that the file doesn't exist and is being created. After the file is created by the script, the loop has a break statement to continue forward.

```
while(True):
    try:
        objF = open(objFile, "r")# Tries to read the file
        break
    except FileNotFoundError:# If the file isn't found, throws an exception and creates the file and restarts the program.
        print()
        print("Cannot find file, creating the " + objFile + " and restarting...")
        objF = open(objFile, "x")
        continue
```

After the file is created, then the script reads the file with a for loop and each row in the file is taken into the strData value and split by a "," and then entered into a dictionary (dicRow) where the first value in strData is attributed to the "Task" key and the 2nd value in strData is attributed to the "Priority" key. Lastly the new dictionary row is added to a list called lstTable.

```
for row in objF:
    strData = row.split(",")
    dicRow = {"Task": strData[0], "Priority": strData[1].strip()}
    lstTable.append(dicRow)
```

After the file has been read into memory by the script, I moved onto Step 3 which was to display the current data in memory back to the user. This was first done by initially checking if the lstTable is empty (aka an empty file was read) and letting the user know that there was no data to display. Otherwise for every row in the list (lstTable) the script printed out the "Task" key and it's value separated by a "," back to the user. Also the lstMenu variable had the strChoice appended to it for exiting purposes which will be discussed below.

```
if (strChoice.strip() == '1'):
    lstMenu.append(strChoice)# Keep track of what choices the user has made
    if not lstTable:# checks if the lstTable is empty
        print("No data stored in " + objFile + ".")
    for row in lstTable:
        print(row["Task"] + "," + row["Priority"])
    continue
```

Moving onto Step 4, which was giving the user to add an item to the list. Created two new variables (strUserTask and strUserPriority) which both asked the user for a Task they would like to add and a priority that they would like to give that task. With the user input, created a new dictionary entry (dicRow) assigning the strUserTask input to the "Task" key and the strUserPriority input to the "Priority" key. And appending the newly created dictionary row to the list (lstTable).

```
elif (strChoice.strip() == '2'):
    lstMenu.append(strChoice)# Keep track of what choices the user has made
    strUserTask = input("What task would you like to add: ")
    strUserPriority = input("What priority would you give to this task: ")
    dicRow = {"Task": strUserTask.strip(), "Priority": strUserPriority.strip()}
    lstTable.append(dicRow)
    continue
```

Next was Step 5, which was to add the functionality to allow the user to remove an item from the list. For this, I initially did a check to see if there are any items in the list if there were none then told the user that there were no tasks stored and they should create one first before deleting it. If there were tasks in the list, then the whole list is printed out to the user and then their input (in the form of strRemoveTask variable) was asked from the list above which item would they like to remove and also stating that their response was case sensitive. Using the user input, with a for loop I looked through the entire list of tasks and if the task that the user asked to be removed was in the list then the script would tell the user that the task was found and deleted. The deletion was done by deleting the dictionary entry at the row that it was found in the list. If the item was not found, then the user got a response from the script saying that the task wasn't found. This portion took the longest for me, because I wanted to give the user the ability to delete any item from the list not just the latest entry that they put in.

```
elif (strChoice.strip() == '3'):
    lstMenu.append(strChoice)# Keep track of what choices the user has made
    if not lstTable:# checks if the lstTable is empty
        print("No data stored in " + objFile + ". Nothing to delete, please add task(s) first.")
        continue

    print("Current Task(s):")# display current tasks
    for row in lstTable:
        print(row["Task"])

    strRemoveTask = input("Which task would you like to remove from the list above? (Case Sensitive) ") #ask the user which task they would like to delete
    for row in lstTable:
        if strRemoveTask in row["Task"]: #search for the user selected task in the list of tasks
            print()
            print("Found the Task, " + "\"" + strRemoveTask + "\"" + "; deleting from task list.")
            lstTable.remove({"Task": row["Task"], "Priority": row["Priority"]}) #if task is found in the last, delete that row from the list
            break
    else:
        print()
        print("Task not found, please try again.") #task not found
    continue
```

Next for Step 6, which gave the user the ability to save their current list to a file. I started with opening the file with the "write" functionality and for every row in the list (lstTable) write it out to the ToDoList.txt file with a "new line" carriage return at the end so that every entry was on a new line in the text file. And the user was told that their data was saved.

```
elif (strChoice.strip() == '4'):
    lstMenu.append(strChoice) # Keep track of what choices the user has made
    objF = open(objFile, "w")
    for row in lstTable:
        objF.write(row["Task"] + "," + row["Priority"] + "\n")
    objF.close()
    print()
    print("Data Saved!")
    continue
```

Last portion of the code asked for creating an exit function (Step 7). Rather than just quitting the program, I wanted to user to be notified if they modified their list without saving it first and given a chance to save the data to a file before closing. I did this with an if statement and the lstMenu that has been getting appended to every time that the user chose an option from the main menu. If the user chose the add/delete task option (option 2 or 3) and if they have not chosen, the option to save (option 4) then the script would ask them if they want to save their work first. Depending on if they choose “y” or “n” the script would either just exit out or go back to the main menu giving them a chance to save.

```
elif (strChoice.strip() == '5'):
    if (("2" in lstMenu or "3" in lstMenu) and "4" not in lstMenu):
    #if the user has added or deleted tasks from the list and they have not saved the data, ask them if they want to save
        strExit = input("You added or deleted task(s) without saving, are you sure you want to quit? (y/n) ")
        if (strExit.lower().strip() == 'y'):
            print()
            print("Exiting...")
            break # Exit the program
        elif (strExit.lower().strip() == 'n'):
            continue
    else:
        print()
        print("Exiting...")
        break  # Exit the program
```

Summary

Overall, the script worked as intended and reflected what was being asked for in the assignment. I added extra functionality to the script outside of the original assignment to help add some more functionality to the program as stated in the steps above.

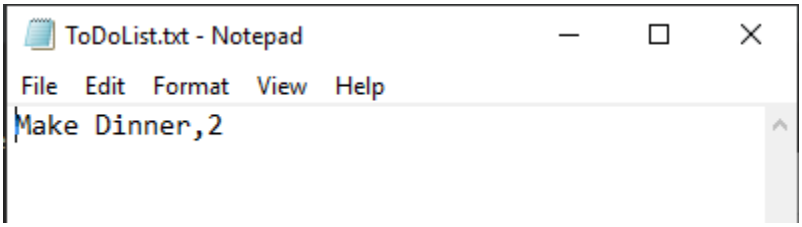
The results from running the script from PyCharm were:

```
C:\_PythonClass\Assignment05\venv\Scripts\python.exe C:/_PythonClass/Assignment05/Assigment05_Starter.py

Cannot find file, creating the ToDoList.txt and restarting...

    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

No data stored in ToDoList.txt.

    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

What task would you like to add: Make Dinner
What priority would you give to this task: 2

    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

Make Dinner,2
```

```
    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

What task would you like to add: Take Test
What priority would you give to this task: 1

    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

Current Task(s):
Make Dinner
Take Test
Which task would you like to remove from the list above? (Case Sensitive) Take Test

Found the Task, "Take Test"; deleting from task list.

    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

Make Dinner,2
```

```
    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

You added or deleted task(s) without saving, are you sure you want to quit? (y/n) n

    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


Data 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


Exiting...

Process finished with exit code 0
```

The output in the ToDoList.txt file was:



And running the script from the OS shell looked like this:

```
C:\Users\dimaa>cd c:\_PythonClass\Assignment05\

c:\_PythonClass\Assignment05>Python Assigment05_Starter.py

    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

Make Dinner,2

    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

What task would you like to add: Take Test
What priority would you give to this task: 1

    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

Current Task(s):
Make Dinner
Take Test
Which task would you like to remove from the list above? (Case Sensitive) Take Test

Found the Task, "Take Test"; deleting from task list.
```

```
    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

You added or deleted task(s) without saving, are you sure you want to quit? (y/n) n

    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


Data 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


Exiting...
```

The output in the ToDoList.txt file was: