Adam Gibbs

5/17/2020

IT FDN 100 A Sp 20: Foundations of Programming: Python

Assignment 05

Create a Python Task Table/Dictionary Script with Menu

**Introduction**

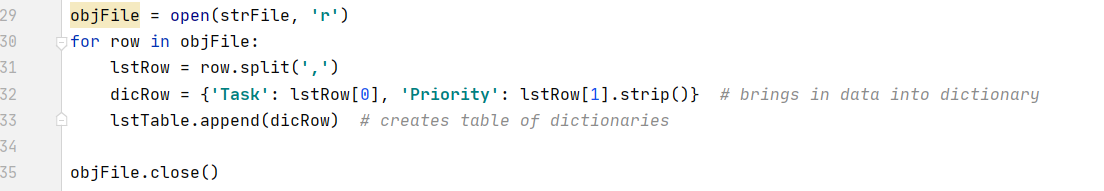
This paper will discuss how to create a task table script file in Python that allows a user to choose options from a menu on what they want to do with their task table. The user can enter data, view data, edit data, or save/exit. If the user wishes, their data will be saved to a .txt file. This script also imports data from the previous task .txt file so it can be edited and updated.

**How the script works**

Using the Python script editing tool PY CHARM, this assignment involved using the print and input functions with string variables, the append and write function to store keystrokes in a .txt file, and a ‘while’ loop and ‘if’, ‘elif’, ‘else’ functions. I started the script by identifying variables that will be used in future steps. After naming the variables, the following steps were completed.

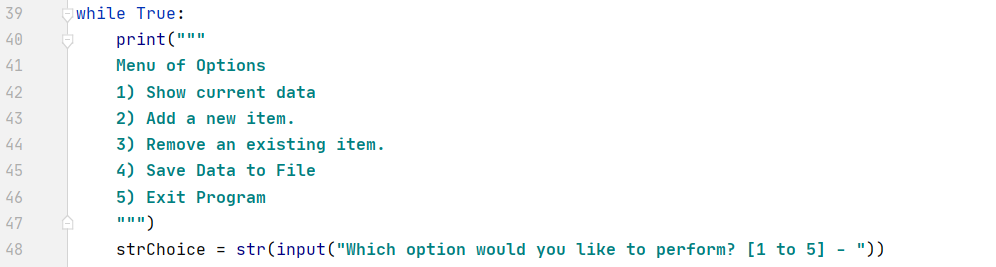
Step 1: Load data from existing .txt file into a table of dictionary rows in memory.

For this step I began by opening the ‘ToDoList.txt’ and telling the script to read the file using function ‘r’. I then used the ‘for’ function to call out the rows in the .txt file, split them with commas and placed each row into a dictionary with keys ‘Task’ and ‘Priority’. These dictionaries then append a table name ‘lstTable’ and the .txt file is closed (see Figure 1. for code). Now a table made of dictionary rows from the .txt is stored in memory and can be used during the rest of the script.

Figure 1.

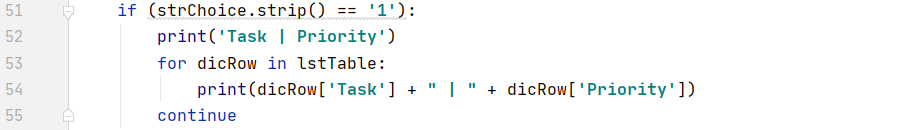
Step 2: Display a menu of choices to the user.

This step is completed using the ‘print()’ function. I nested the print function within a while loop to complete additional steps that will be discussed later. As other function within this while loop are completed the loop will restart with this menu. This step is completed with a user input (see Figure 2.) I also placed an ‘else’ statement at the end of the ‘while’ loop which prompts the user is they enter a character that is not a menu choice.

Figure 2.

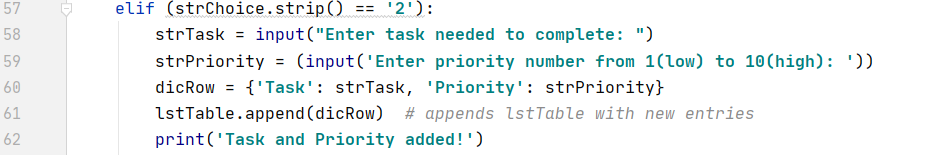
Step 3: Show current items in table

If the user input in step 2 equals ‘1’ this step beings. This is possible using an ‘if’ statement withing the ‘While’ loop. The printout will start with a header ‘Task | Priority’ followed by the values withing the ‘lstTable’. I used a ‘for’ function to print out the dictionary values for keys ‘Task’ and ‘Priority’ within the table (see Figure 3.). Once complete, the loop restarts with the menu.

Figure 3.

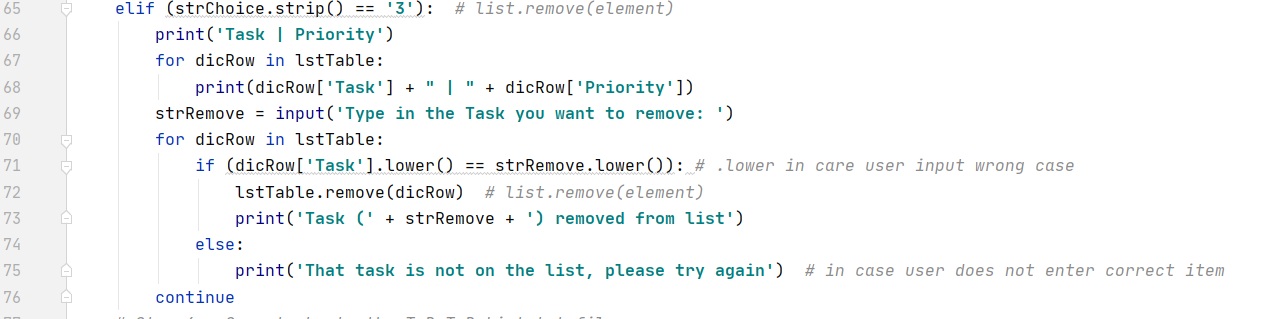
Step 4: Add a new item to the table.

I used an ‘elif’ function within the while loop to start the script if the user input equals ‘2’. This section asks the user for two inputs, a Task and its Priority. Those input are then assigned to keys within a dictionary and the new dictionary is added to the ‘lstTable’. The script then prints a confirmation to the user that their entry has been added (see Figure 4.).

Figure 4.

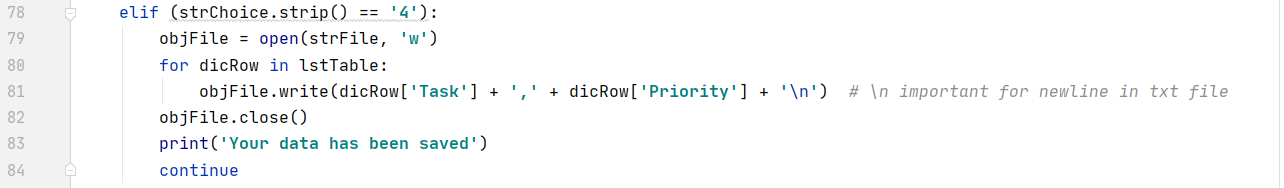
Step 5: Remove an item from the table.

I used an ‘elif’ function within the while loop to start the script if the user input equals ‘3’. This step allows the user to delete an item from the list. I start this section by printing out the table like I dis in step 3. This lets the user see what is in the list before they decide what to remove. The user this then asked for an input. This input, which is the name of value under the key ‘Tasks’, is set to equal the variable ‘strRemove’. This is followed by a ‘for’ function which looks for the dictionary row within the table that has a ‘Tasks’ value equal to the user input. If the script finds the correct dictionary row it is then removed using the ‘remove’ method. If the user input is not equal to any value within the ‘Tasks’ key, the user is told the try again and the while loop restarts. This is completed using an ‘else’ function (see Figure 5.).

Figure 5.

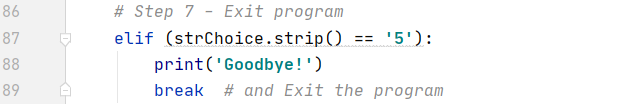
Step 6: Save tasks to the .txt file

I used an ‘elif’ function within the while loop to start the script if the user input equals ‘4’. This saves the new appended table to the .txt file. I start by opening the .txt file and then tell the script to write over the file using ‘w’. I want to write over the previous file because the newly created list is the complete list. If I appended the .txt file it would simply add the new list to the old list which is not correct for this script. I the us the ‘for’ function to call out each dictionary row with the ‘lstTable’ and then write each row to the .txt file. I used ‘\n’ at the end of each row to ensure a character turn separates each row. The file is then closed, and a confirmation statement is then printed to the user (see Figure 6.).

Figure 6.

Step 7: Exit Program

I used an ‘elif’ function within the while loop to start the script if the user input equals ‘5’. If the user enter 5, the statement ‘Goodbye’ is printed and the programs ends with a break of the While loop (see Figure 7.).

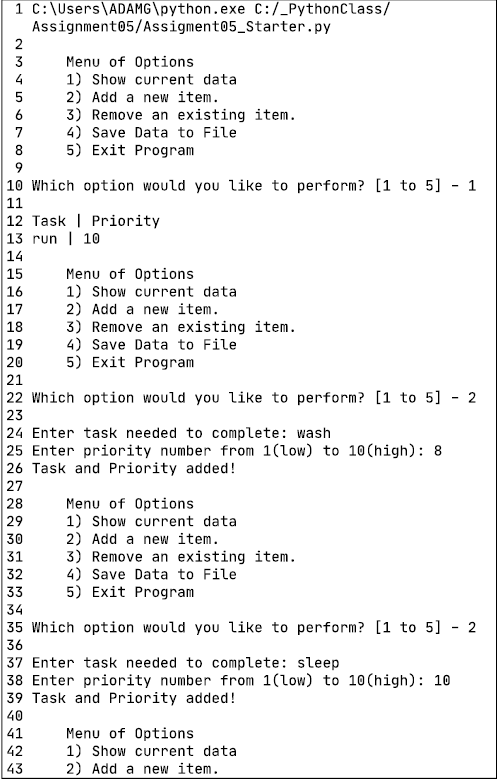
Figure 7.

**Lessons learned**

Step 5, removing item, was the most difficult to wrap my mind around. I first tried to simply look for a value within the table. This did not work, and research revealed that this can not be done. You must look for a value within a key. I was not fully comprehending that the table I created is a table of dictionaries and that I needed to find the dictionary that had a certain value for key ‘Tasks’. It took me diagraming my table out on paper before I could comprehend this.

**Script running**

Figure 8. below shows the script running in Pycharm. For this run I first show what is already on the table. I then add task ‘wash’ with priority ‘8’ and task ‘sleep’ with priority ‘10’. Then I remove task ‘run ‘, show the new list, save the file and exit the program. Figure 9. Shows the contents of the created .txt file.



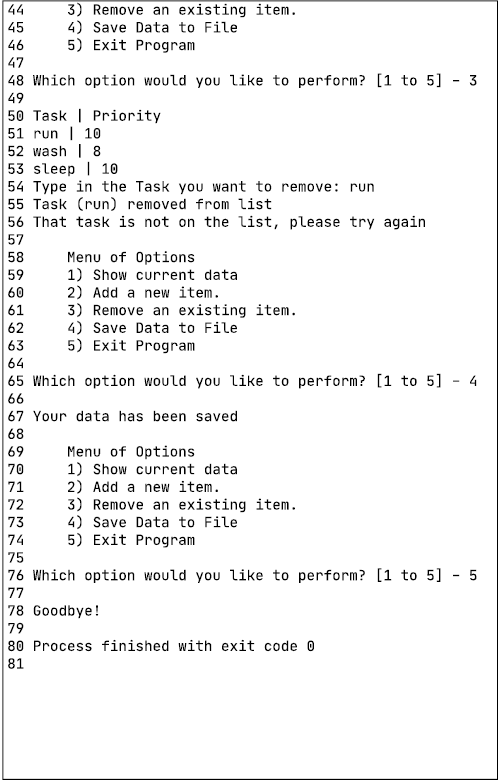


Figure 8.

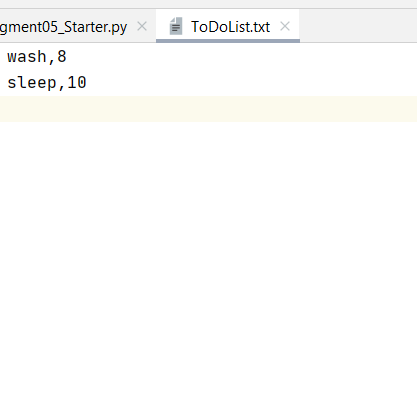
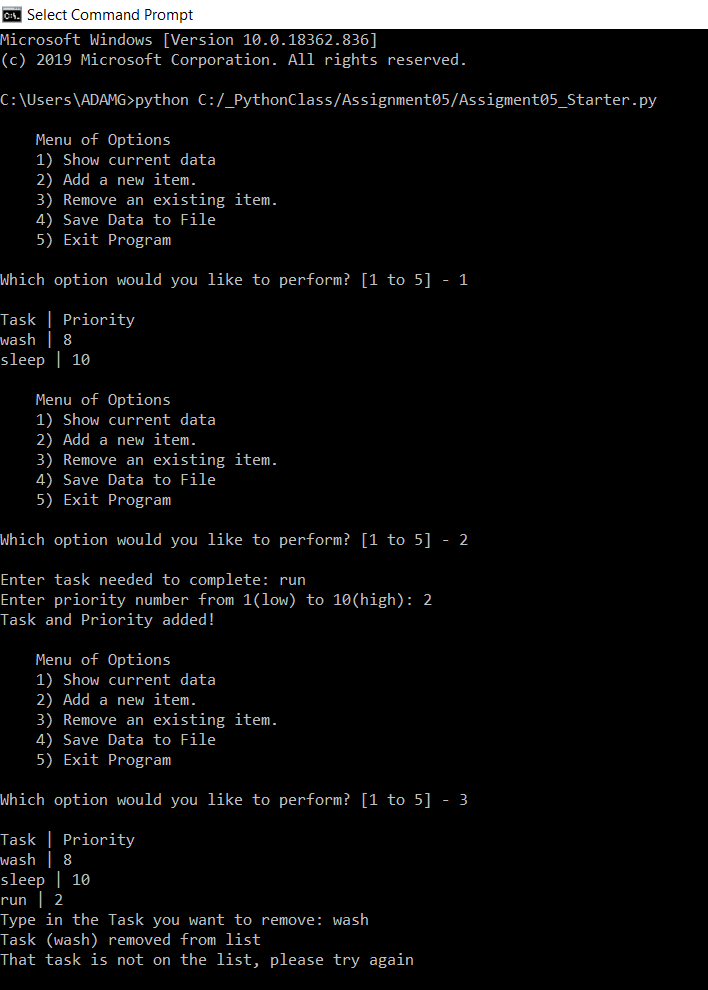


Figure 9.

Figure 10. below shows the script running in CMD. For this run I first show what is already on the table. I then add task ‘run’ with priority ‘2’. Then I remove task ‘wash ‘, show the new list, save the file and exit the program. Figure 11. Shows the contents of the created .txt file.



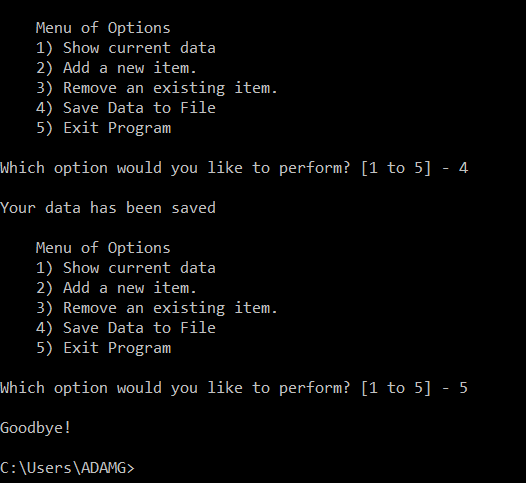
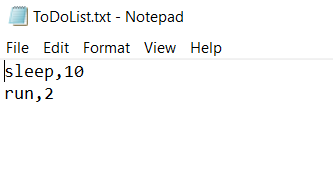


Figure 10.

Figure 11.

**Summary**

Through this assignment I created a task table file in Python that allows a user to choose options from a menu on what they want to do with their task table. The user can enter data, view data, edit entries or save/exit. The created table is a table of dictionaries.