Hyojin Kwak

November 16, 2020

Foundations of Programming

Assignment05

https://github.com/hihojin/IntroToProg-Python/tree/main

**Creating a Starter.py**

**Introduction**

This week, we learned about dictionary and how to change a script with lists to dictionaries. I learned that dictionaries contain keys and values and each key in dictionary is unique and stores its value, therefore we can return its value with the assigned key. Starter.py is a modified script from the last week’s HomeInventory.py assignment, scripting with dictionaries and adding the row of data in dictionaries to the table of lists. Below paragraphs will show a step-by-step information of how I wrote codes to modify the script.

**Creating the Program**

First, I copied and pasted the codes of ‘Starter.py’ script from a file in canvas and pasted to ‘TextEdit.txt’ so that I can paste the codes right to PyCharm by keeping the original code format. I changed the changelog with my name, date, and the task in the script header as guided.

In step 1 where to load any data when the program starts, I used open function with read as a task to bring any data from the ‘ToDoList.txt’ file. Then to go through each row of data in text file I used for-in objFile: and .split function to split with comma between a, and b which are values for each key: Task, and Priority. A and B variables I used are similar with lstrow[0] and lstrow[1] in the lecture example. Then I defined dicrow with task: a and priority: b and these a and b will be variables of user’s input of task and priority. Lastly, I used lsttable.append so this dicrow will be added to the lsttable as a dictionary.

To show the current items in the table, I used for-in again so it can read row by row data from the text file. Since dictionaries will be in the lsttable, I formatted as ‘for i in lsttable: print(i[“Task”] + “, ” + i[“Priority”]). This code will bring each dictionaries’ values of task and priority in the lsttable.

In a section to add new items to the lsttable, I first used input functions for user to put their task and priority then I defined what dicrow is with the dictionary format {“Task”: “strtask”, “Priority”: “strpriority”} and added to the lsttable with .append function. To show the added data in the text file, I opened the text file with append(‘a’) mode. Since key brings its value, I coded as dicrow[“Task”], and dicrow[“Priority”] with the carriage return: ‘\n’ so that the next code will be written one line below of the existing ones. Next, I closed the file with append mode and reopened the file with read mode to show the user about the added data. I used ‘for row in objfile:’ with split(“,”) function between values of task and priority a, and b , and lastly strip() function in the end to cut off unnecessary spaces. I printed this code, closed the file again and moved on to the next section.

To remove the lastly added data, I used pop function with lsttable as list has pop function. I used ‘len function -1’ inside of the pop function to indicate the correct index number. Then I printed the lsttable and it showed the current data with the lastly added dictionary deleted. As for text file, I opened the same text file but with write mode this time so the only new data will be written. Objfile.write(i[“Task”] + “,” + i[“Priority”]) is the code I used with I as a variable indicating for a dictionary. The last code in the step 5 was to close the objfile.

Step 6 and step 7 were the easiest where I just used print function in the step 6 and showed the user that the data has been saved to the text file and input and print function in step 7 to ask user to press enter to exit.

Text

Description automatically generated

*Figure 1. Codes from step 1-4*

Text

Description automatically generated

*Figure 2. Codes from step 4-7*

**Getting Output**

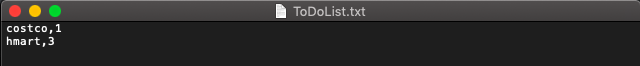
**Text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

*Figure 3. Output in PyCharm*



*Figure 4. Output in ToDoList.txt*

It was very challenging to get the finally running code for this week’s assignment. Still there are things I need to work on to make the script running more smoothly, however I am glad that the script is running fine.

**Summary**

It was little bit challenging to complete this week’s assignment than any other assignments before as the program was little bit longer and that we had to combine all codes that we have learned so far. I believe I still need to work on some codes for step 1 as I figured the codes work when there are existing data in the text file but make an error when there is none. As next week’s assignment is related to this week’s assignment, I will be working on this part.