Kenji Petrucci

8/9/2022

IT FDN 110 B Su 22: Foundations of Programming: Python

Assignment 5

**Assignment 5 Write Up**

**Introduction**

Assignment 5 is to modify starter script to read data from a file and load that data into a dictionary object. Using this data we will build a table and then, using a printed menu and user inputs, display, add, remove, or save new data.

**Topic 1**

Similar to the previous assignment we begin by declaring variables and constants. Once our declarations are made, we need to create the target file if it doesn’t already exist.

Like before we then read in the data and store it, only this time we use a dictionary object where we define the “keys” and “values.”

lstrow = row.split(",")  
rows = {"task":lstrow[0], "priority":lstrow[1].strip()}

The code above shows how we split the data we just read in at the “,” delineation. We then define “rows” as a dictionary object populated with key = “task” and the associated value = index[0] of the data (first column). This is how we read data, separate it, and assign it to keys.

Displaying the data to the screen uses the same methodology as previous exercises; a for-loop steps through each row of data and prints it to the screen.

However, adding new data is done slightly differently. Like before we define variables to store user inputs but now we’ve used the previously defined keys to store new values. Using the .append() function allows us to add each new row of data to the table.

The most interesting (and difficult – for me) part was allowing the deletion of specific rows of data. Initially, I added a “row ID” “column” to the data set; the intent was to have the user input numbers as they input data. However, it became clear that this was not an ideal way to number the data. There would be no protection against duplicate entries, nor would there be a requirement to enter a number at all. I elected to use an interval counter similar to what we covered in one of our earlier modules. This method eliminates the unknowns of allowing user input and guarantees integer increments as specified. From there it was a simple matter of displaying the data with the row numbers, defining a variable for the user input specifying the data to delete, and deleting the row. Out of habit I started the increment counter at 0 which required me to subtract 1 from the user input; without that subtraction a user input of 1 would remove item 2, and so forth. After reviewing this, I’m curious to know if I started the counter at 1 if this additional math would have been necessary. For the sake of time I have chosen not to alter the code for submission but will continue to experiment with this.

Finally, to write the data to file it is necessary to convert the data (dictionary object) to a string; attempting to write a dictionary object to file results in an error.

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedThis assignment calls for documentation of the script running in **both** PyCharm and the OS command window. Figure 1 below shows screen shots of the program running in PycCharm. Program executes from left to right --------------------------------->

*Figure 1, ToDo.py running in PyCharm*

Figure 2, below, shows a screen shot of the data captured in the .txt file.

Graphical user interface, text, application, chat or text message

Description automatically generated

*Figure 2, ToDoList.txt*

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedFigure 3, below, shows the script running in the Windows command window. Program executes from left to right. ------------------------------------------------------------------------------------->

*Figure 3, ToDo.py running in the Windows command window*

**Summary**

In this example we guided users through data collection, manipulation, and display using a printed menu. We used dictionary objects and lists to build and manipulate tabulated data and save it to a file. We used increment counters to eliminate user input risk and identify specific data sets for manipulation. PyCharm and .txt files are attached.