Jonathan Den Haan

11/11/2022

Foundations of Programming: Python

Assignment 05

**Module 5**

**Introduction:**

In this module we continue our work in understanding lists, including built in functions to work with lists, and writing and reading lists from/to a file. Next we begin learning about dictionaries, which are another type of sequence data with key:value pairs that can only be called by referencing the key. We are also introduced to standard formats for coding and how to separate our code functionality for easier reading/troubleshooting. Finally we take a look at GitHub which is an online repository of code that is useful for updating code with other users.

**Labs:**

**Lab05 – A**

This script is essentially trying to do what we did in Assignment 04, that is, to be able to read, write to and view data from the CDInventory text file. The added functionality in this script will be that it can read and display data from file as well as write to file.

Text

Description automatically generated

Text

Description automatically generated

This script starts by giving the user a bunch of choices for adding to the data, writing or saving the data to file, reading data into memory and displaying the data. The write portion of the script is not really new, it asks for an ID, CD title and Artist name and then adds that to the lstRow variable which is then appended to the lstTbl variable. This is appended so that any data already in lstTbl is not erased. I ask the script to print lstRow and lstTbl just to confirm for myself what’s happening.

The save portion of the script is likewise similar to assignment 4, but I really worked hard to make sure the formatting was correct. The script uses a nested for loop to take every item in every column in every row and write it into the variable strRow as a string, and then write that into the text file. Running this multiple times in the same instance of the script will add all items in lstTbl to the file, which poses a problem since some of the data may already be entered, which is annoying since you get repeats in the file. I think the only way I can think to fix this would be to add to the write/save script so that it looks in the file and if an entry matches what it’s about to write it skips that entry.

The read portion of the script first opens the text file, then looks at each row and splits the row at the commas and adds it to the lstRow variable, then adds that item in lstRow and appends it to whatever was already in lstTbl using the += operator. I wanted to account for the fact that the user may already have been entering data while running the script, so I don’t delete whatever was already entered.

The display feature just prints the lstTbl variable so the user can see what’s in memory.

Lab05-B

This script works by first defining some new variables compared to the list method. It defines dicRow{} as an empty variable that is of type: dictionary. If the user chooses to add data, the script directly builds key and value pairs for dicRow using the input function. Keys are created in the script, and their corresponding value is whatever is entered by the user. Finally the dicRow data is appended to the lstTbl variable of type:list where it is stored while the script is running.

If the user chooses to read data from file, the script opens the file, strips out the commas and splits at the commas, then adds the values to the lstRow variable. Then it uses the items in that variable to assign values to the dicRow dictionary and appends them to the lstTbl variable.

I could not figure out how to write values from the dictionary to file. This seems like it should just be the inverse of reading the file, but the syntax evaded me.

Step 4.

The first thing notice is that there is no confirmation that anything is happening when the choices made are either r, or exit. I would probably want to have some message displayed using the print function so that the user knows that their inputs have been received and acted upon by the script. I would also want the display choice to produce better formatting, right now it looks like it’s been configured to just print the dictionary which looks a little clunky.

**Assignment05:**

Running in Spyder:

**Text

Description automatically generated**

**Text

Description automatically generated**

Running in Anaconda Prompt:

Text

Description automatically generated

This script was a fairly simple update of the script in lab B, except I actually had to deal with the problem where I couldn’t write data into file from the dictionary format. I first set about solving this. I first tried setting a bunch of variables equal to the value in each dictionary key:value pair. Then I tried writing those into another variable as a list. But I kept getting an error where the script didn’t like the format of that list and wouldn’t write it into the file. So I had to find the .join(map()) function to convert the dictionary values into strings that python could then write to file. I’m sure there is a more simple way of doing it.

I then also had to add functionality for deleting a dictionary row. I did this by adding a choce ‘e’ and using an elif to prompt the script to

A)print out what items are currently in memory

b) ask the user which item the want to delete

c)check to see if that ID is in memory, and then remove each key:value pair in that row.

Overall, I’m able to get this script to work if only 1 item is in memory. It will perform all functions needed. However, if I add another set of key:value pairs, it will overwrite the first one. Since I have no idea how this is happening I’m at a loss as to how to solve it. It seems to be happening when I tell the script to append the dicRow to the lstTbl, but why that would overwrite rather than append(!) is beyond me! This leads to some cascading errors that I’m pretty sure could be fixed if this was resolved.

**Summary:**

In this module we learned more about lists and then focuses in on dictionaries as another way to work with and store data. We saw how dictionaries can be used like lists where you can call values from the dictionary using the key, similar to how you would access values from a list using an index. We also learned how to work collaboratively with our scripts on GitHub, and started learning the best ways to organize our code.