Cheryl Montejo

August 7, 2022

Foundations of Programming, Python

Assignment 5

CD Inventory (again): With Dictionaries!

# Introduction

For this assignment, I will explain the steps I took to update an existing Python script for a CD inventory system to use a list of dictionaries, allow for the loading of data, and allow the user to delete rows in the data as well.

# Learning Through the Module

For this assignment, I again spent most of my time focusing on the lab in the module. I skimmed through Chapter 5 in the book, but I didn’t feel like it added anything extra to what I learned from the lab or from looking up information online.

# Drafting the Script

For this assignment, we were asked to modify an existing script that was a solution to Assignment 4. I ended up rearranging the script and adding spaces between each section to make it easier for me to read. The script also included “to do” sections to address the new required features of deleting and entry and loading data from a text file. As I worked through the script, I edited the psuedoscript to describe each section more accurately.

# 2. Add data to the table (2d-list) each time the user wants to add data

**if** strChoice **==** 'a'**:**

strID **=** **input(**'Enter an ID: '**)**

strTitle **=** **input(**'Enter the CD\'s Title: '**)**

strArtist **=** **input(**'Enter the Artist\'s Name: '**)**

intID **=** **int(**strID**)**

dicRow **=** **{**'ID'**:**intID**,**'Title'**:**strTitle**,**'Artist'**:**strArtist**}**

lstTbl**.**append**(**dicRow**)**

For the Section 2 where users would enter data, I updated this section to use dictionaries. I used the example script from Module 5 to do this. This required me adding and defining a new variable called dicRow.

# 5. Add functionality of deleting an entry

**elif** strChoice **==** 'd'**:**

delID **=** **int(input(**'Please enter the ID of the row you want to delete: '**))**

**for** i **in** **range(len(**lstTbl**)-**1**):**

**if** lstTbl**[**i**][**'ID'**]==**delID**:**

**del** lstTbl**[**i**]**

**pass**

**print(**'Row has been cleared.\n'**)**

I also used Module 5 to help address Section 4 to load existing data from a text file. I initially had issues with this because when I was exporting data to the text file, it was including the keys and all the punctuation marks. I appreciate the suggestion from class to be intentional about the formatting when exporting the data to a text file.

# 3. Display the current data to the user each time the user wants to display the data

**elif** strChoice **==** 'i'**:**

**print(**'ID | CD Title | Artist'**)**

**for** row **in** lstTbl**:**

**print(list(**row**.**values**()))**

**print()**

For Section 3 to print the inventory that is currently stored locally, I chose to specifically print just the values without the keys as I felt the “**print(**'ID | CD Title | Artist'**)”** line should act as a header for that section, and it looks cleaner with just the values listed. To be able to print the values without the dict\_values object, I defined the row as a list based on a suggestion from StackOverflow (StackOverFlow, <https://stackoverflow.com/questions/70604810/how-to-print-dictionary-values-without-dict>) (external site)[[1]](#footnote-1).

# 5. Add functionality of deleting an entry

**elif** strChoice **==** 'd'**:**

delID **=** **int(input(**'Please enter the ID of the row you want to delete: '**))**

**for** i **in** **range(len(**lstTbl**)-**1**):**

**if** lstTbl**[**i**][**'ID'**]==**delID**:**

**del** lstTbl**[**i**]**

**pass**

**print(**'Row has been cleared.\n'**)**

For Section 5 where I added the functionality to delete an entry, I also relied on an external website to help out with script (Geeks for Geeks, <https://www.geeksforgeeks.org/python-removing-dictionary-from-list-of-dictionaries/>) (external website)[[2]](#footnote-2). I did struggle with some errors around the index being outside of the range when I was choosing to delete items that weren’t last on the list. For example, when I had 3 rows in my list, row 3 would delete as expected but rows 1 and 2 would cause an error. I remedied this by the addition of the "-1" in **for** i **in** **range(len(**lstTbl**)-**1**):** to keep the range and the index consistent inside of the for loop.

# 6. Save the data to a text file CDInventory.txt if the user chooses so

**elif** strChoice **==** 's'**:**

objFile **=** **open(**strFileName**,** 'w'**)**

**for** row **in** lstTbl**:**

**for** value **in** row**.**values**():**

value **=** **str(**value**)**

objFile**.**write**(**value **+** ','**)**

objFile**.**write**(**'\n'**)**

objFile**.**close**()**

I really struggled during Section 6 for saving the data to a text file. As mentioned when I discussed Section 4 where we import data from a text file, I was having issues loading the data into the script because of the formatting when I was exporting it. I tried following the example from Module 5, but it doesn’t quite work the same because of the switch from using a list of lists to a list of dictionaries. Specifically, having the key be an integer made it difficult to export the data the same way. With more help from the internet, I was able to create nested for loops to print the values separated by commas to the text file (Geeks for Geeks, <https://www.geeksforgeeks.org/write-a-dictionary-to-a-file-in-python/> and <https://www.geeksforgeeks.org/iterate-over-a-dictionary-in-python/>) (external sites)[[3]](#footnote-3). I’d like to think there is a simpler and cleaner way to print the values with commas separating each value, but this is as close as I could get it. Fortunately, the formatting still works when you import the data back in from the text file.

# Running the Script in Spyder and Anaconda Prompt

For running the script in Spyder, I launched Spyder and opened CDInventory.py (File -> Open…->CDInventory.py). Through the menu, I tested out the options. I started by loading multiple items into the inventory before displaying the inventory and then saving it to a text file. I then deleted an item from the inventory before saving the inventory to a text file again and exiting the script. I ran the script again and loaded the inventory from the text file to make sure that feature also worked.

Text

Description automatically generated

Figure - Screenshot showing CDInventory.py in Spyder being used to add items to the inventory

A screenshot of a computer

Description automatically generated with medium confidence

Figure - Screenshot showing CDInventory.py in Spyder being used to display the inventory, save the data to a text file, and then to delete an item

Graphical user interface, text, application

Description automatically generated with medium confidence

Figure - Screenshot showing the data that was saved to a text file named CDInventory.txt

A screenshot of a computer

Description automatically generated with medium confidence

Figure - Screenshot showing CDInventory.py in Spyder being used to save an updated inventory, exiting, and then reloading to import the inventory from the text file

Application

Description automatically generated with low confidence

Figure - Screenshot showing the data in the text file after deleting an item and then saving to the text file

For running the script in Anaconda Prompt, I opened the Terminal console and navigated to the correct folder using the cd command. I then used the python command with the file name, CDInventory.py, to run the script. Using the menu, I started this time by loading the inventory from the text file that had been created from the script in Spyder. I then proceeded to add an item, display the inventory, and then save it to the text file.

Text

Description automatically generated

Figure - Screenshot showing CDInventory.py in the terminal being used to load the existing inventory and add an item

Text

Description automatically generated with medium confidence

Figure - Screenshot showing the data in the text file after adding an item and saving to the text file

After saving the inventory to the text file, I then proceeded to delete a different item from the inventory, displayed the inventory, and saved the text file again before exiting the script.

Text

Description automatically generated

Figure - Screenshot showing CDInventory.py in the terminal being used to delete an item and save the data to a text file

Application

Description automatically generated with low confidence

Figure - Screenshot showing the data in the text file after deleting an item and saving

# Summary

In this assignment, we explored using a list of dictionaries and how that impacted the existing script we had from assignment 4. We also updated the functionality of the script to be able to delete items from the inventory and load an existing inventory from a text file. I know we also introduced functions as a part of this module, so I am looking forward to applying that knowledge and using functions in future assignments.

# Appendix

## Listing AddressBook.py (pasted from Notepad++)

#------------------------------------------#

# Title: CDInventory.py

# Desc: Assignment 05 - CD Inventory that allows users to load data, add data, display inventory, delete entries, and save data to txt file

# Change Log: (Who, When, What)

# DBiesinger, 2030-Jan-01, Created File

# CMontejo, 2022-Aug-01, Updated tuples to dictionaries, added functionality for deleting

# Cmontejo, 2022-Aug-04, Added functionality for saving list to txt file

# CMontejo, 2022-Aug-07, Troubleshooting and updating delete functionality, updated script description

#------------------------------------------#

# Declare variables

strChoice **=** '' # User input

lstTbl **=** **[]** # list of lists to hold data

lstRow **=** **[]** # row in list

dicRow **=** **{}** # dictionary

strFileName **=** 'CDInventory.txt' # data storage file

objFile **=** **None** # file object

# Get user Input

**print(**'The Magic CD Inventory\n'**)**

**while** **True:**

# 1. Display menu allowing the user to choose:

**print(**'[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory'**)**

**print(**'[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit'**)**

strChoice **=** **input(**'l, a, i, d, s or x: '**).**lower**()** # convert choice to lower case at time of input

**print()**

# 2. Add data to the table (2d-list) each time the user wants to add data

**if** strChoice **==** 'a'**:**

strID **=** **input(**'Enter an ID: '**)**

strTitle **=** **input(**'Enter the CD\'s Title: '**)**

strArtist **=** **input(**'Enter the Artist\'s Name: '**)**

intID **=** **int(**strID**)**

dicRow **=** **{**'ID'**:**intID**,**'Title'**:**strTitle**,**'Artist'**:**strArtist**}**

lstTbl**.**append**(**dicRow**)**

# 3. Display the current data to the user each time the user wants to display the data

**elif** strChoice **==** 'i'**:**

**print(**'ID | CD Title | Artist'**)**

**for** row **in** lstTbl**:**

**print(list(**row**.**values**()))**

**print()**

# 4. Add the functionality of loading existing data

**elif** strChoice **==** 'l'**:**

objFile **=** **open(**strFileName**,** 'r'**)**

**for** row **in** objFile**:**

lstRow **=** row**.**strip**().**split**(**','**)**

dicRow **=** **{**'ID'**:int(**lstRow**[**0**]),**'Title'**:**lstRow**[**1**],**'Artist'**:**lstRow**[**2**]}**

lstTbl**.**append**(**dicRow**)**

objFile**.**close**()**

# 5. Add functionality of deleting an entry

**elif** strChoice **==** 'd'**:**

delID **=** **int(input(**'Please enter the ID of the row you want to delete: '**))**

**for** i **in** **range(len(**lstTbl**)-**1**):**

**if** lstTbl**[**i**][**'ID'**]==**delID**:**

**del** lstTbl**[**i**]**

**pass**

**print(**'Row has been cleared.\n'**)**

# 6. Save the data to a text file CDInventory.txt if the user chooses so

**elif** strChoice **==** 's'**:**

objFile **=** **open(**strFileName**,** 'w'**)**

**for** row **in** lstTbl**:**

**for** value **in** row**.**values**():**

value **=** **str(**value**)**

objFile**.**write**(**value **+** ','**)**

objFile**.**write**(**'\n'**)**

objFile**.**close**()**

# 7. Exit the program if the user chooses so

**elif** strChoice **==** 'x'**:**

**break**

# 8. Prints if the user inputs an invalid option

**else:**

**print(**'Please choose either l, a, i, d, s or x!'**)**

1. Retrieved 2022-Aug-01 [↑](#footnote-ref-1)
2. Retrieved 2022-Aug-01 [↑](#footnote-ref-2)
3. Retrieved 2022-Aug-04 [↑](#footnote-ref-3)