Santi Syjuco

02/27/2022

Foundation of Programming (Python)

Assignment\_05 – Knowledge Document

# Introduction

The task for this assignment was to modify the CDInventory\_Starter.py script with the following key changes: (1) use dicts rather than lists for the inner rows (2) Add functionality to the starter script, which included the all (and frustrating) delete function (3) Testing out use of GithHub by posting our work for others to comment on as well as commenting on someone else’s work.

# Steps Followed

## Step 1: Open CDInventory\_starter.py and reviewed the pseudocode.

I opened the CDinventory\_starter.py file, reviewed the pseudocode and determined the key changes I wanted to make to the code. These were:

Converting the User inputs into a dictionary rather than a list and append this to the lstTbl

Display knowledge of the ‘r’ function

Demonstrate how to properly delete items from a text file

## Step 2: Used LAB05 – B as reference

I used my LAB05 – B as reference since majority of the requirements were already covered in this LAB.

## Step 3: Added Code to change the user input to a dictionary rather than rows.

I asked the user for 3 successive inputs as separate variables. From there, I took these 3 inputs and converted them into a dictionary rather than a list.

For good measure, I added a confirmation message to let the user know what they added to the inventory.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **elif** strChoice == 'a': *# no elif necessary, as this code is only reached if strChoice is not 'exit'*  *# 2. Add data to the table (2d-list) each time the user wants to add data. NEED TO CHANGE THIS TO BE DICTIONARY*  strID = int(input('Enter an ID: '))  strArtist = input('Enter the CDs Artist: ')  strTitle = input('Enter the CDs Title: ')  dicRow = {'id': strID, 'Artist': strArtist, 'CD Title': strTitle} *#this defined the dictionary key: values*  lstTbl.append(dicRow)  print('Added', dicRow, 'to the list.') |

## Step 4: Modify ‘s’ (save) function code

I wanted to check whether the input was saved properly to the file, so proceeded to look at the ‘s’ function first.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | *# 4. Save the data to a text file CDInventory.txt if the user chooses so*  objFile = open(strFileName, 'a') *# using option 'a' to add data to file instead of 'w' to add multiple files*  **for** row **in** lstTbl:  strRow = ''  **for** item **in** row.values(): *# this is the line that allowed the data to be written to the text file properly*  strRow += str(item) + ','  strRow = strRow[:-1] + '**\n**'  objFile.write(strRow)  print('Added the following to the file CDInventory.txt') *# this line and next 2 lines shows the user what was added to the text file*  **for** row **in** lstTbl:  print(\*row.values(), sep = ',') *#needed to print row.values() to get the values to print and not the keys*  objFile.close() |

## Step 5: Create the ‘l’ or read function

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | objFile = open(strFileName, 'r')  **for** row **in** objFile:  lstRow = row.strip().split(',')  dicRow = {'ID': lstRow[0], 'Artist': lstRow[1], 'CD Title': lstRow[2]}  lstTbl.append(dicRow) *# simple way to read out the data from the text file as is. it seems to ignore data type. print(objFile.read())*  print("read the following data from CDInventory.txt:") *# added information to let the user know what was read from the file*  **for** row **in** lstTbl:  print(\*row.values(), sep = ',')  objFile.close() |

I wanted to be able to show the user the file that was just read and learned that unless I specified the row in lstTbl to read values (line 8 of the code), it would read keys rather than the values.

## Step 6: Setting Up Delete function

I was very much inspired by the ability to reference lines in a text file through indices, which was shown in the video we were asked to watch: in the Assignment.[[1]](#footnote-1) My high level approach to setting up the delete function as to:

1. Setup a list so I could have the user reference indices to delete the items from the list.
2. Ask the user to input the index representing the CD of their choice from the file
3. Delete the index from the list
4. Write the new list (without the removed item) to file.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **elif** strChoice == 'd':  *# TODO Add functionality of deleting an entry. High level approach: turned the rows of text in CDInventory to items in a list to facilitate deletion by index.*  objFile = open(strFileName, 'r')  lines = objFile.readlines() *# reads each line item in the file strFileName and converts each row into an item in a list called 'lines'*  print(lines) *#shows the user the CDs in the text file to help them decide the CD to remove by inputting its index number on the list*  kill = int(input('Select Index of the CD you want to remove from file. Remember first item is 0: '))  lines.pop(kill) *# removes the CD by referencing its index*  print(lines)  objFile.close()  objFile = open(strFileName, 'w')  **for** row **in** lines: *#writes the remaining items in the list 'lines' to the file without the removed CD*  strRow =''  strRow += str(row) + ','  strRow = strRow[:-1] + '**\n**'  objFile.write(strRow)  objFile.close()  print('The following CDs remain in the file CDInventory.txt')  **for** row **in** lines:  print(row) |

## Step 7: Screen capture working code on Spyder

Text

Description automatically generated

Figure 1 - Home Menu

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 2 Selection [a] Add CD

Text

Description automatically generated

Figure 3 - Save Inventory to file [s] option

Graphical user interface, text, application, email

Description automatically generated

Figure 4 - View of text file

Text

Description automatically generated

Figure 5 Display Current Inventory [i]

Text

Description automatically generated

Figure 6 Load Inventory from File [l]

Text

Description automatically generated

Figure 7 - Deleting an item from file

Graphical user interface, text, application, email

Description automatically generated

Figure 8 New file after deletion of CD 1

## Step 8: Capture Code Working on Terminal

Text

Description automatically generated

Figure 9 - Home menu

Text

Description automatically generated

Figure 10 Add CD

Shape

Description automatically generated with low confidence

Figure 11 Display current inventory

Text

Description automatically generated

Figure 12 SAve file to CDInventory.txt

Graphical user interface, text, application, email

Description automatically generated

Figure 13 See file contents of CDInventory.txt

Text

Description automatically generated

Figure 14 Deleted CD1

Graphical user interface, text, application, email

Description automatically generated

Figure 15 CDInventory.txt no longer has CD1

Text

Description automatically generated

Figure 16 Exit Option

Text

Description automatically generated

Figure 17 Load Inventory from file [l] option

## Step 9: Program Happy Path

Following these steps will result to no errors and demonstrates full functionality of the program

1. Run the program
2. Use option [a]
3. Use option [i]
4. Use option [s]
5. Close the program
6. Run the program
7. Use option [l]
8. Use option [d]
9. Use option [x]

## Step 10: Noted code limitations

1. Using the delete function leaves a ‘\n ‘ in the file. If the user tries to then read the text from the file, [l] option fails because of the gap. Need to manually go into the file to delete this ‘\n’ if ever.
2. User will need to delete CDs with the same IDs one at a time.
3. If I had more time, I would have liked to explore the solution we referenced from LAB04 – C, which had the concept of pulling an integer within a tuple and then returning a result.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 18 LAB 4 C

I got stuck trying to determine how to reference the proper dictionary that would be created using [a] option. My understanding would be repeated CD entries will result to a lstTbl such as that : lstTbl =[dicRow, dicRow, dicRow]. I didn’t know if my understanding was correct, but if so, how do we reference which dicRow to get?

## Step 11: Uploaded python script and assignment to Github

[syjuco/Assignment\_05: Assignment\_05 of UW Foundations of Programming (Python) (github.com)](https://github.com/syjuco/Assignment_05)

# Summary

This assignment, took considerably more time (15+ hours) mostly because of trial and error to find a solution that worked for the delete function. It was challenging to reference the methods of the dictionary that would result to:

1. Referencing the dictionary within a list
2. Performing a deletion

1. https://www.youtube.com/watch?v=m0o0CkYsDzI&feature=youtu.be [↑](#footnote-ref-1)