

Benjamin Reinhart

February 14, 2021

IT FDN 110 A

Assignment 05

Modifying CD Inventory Python Script

Introduction

In this assignment I will cover how to modify and add to a script in Python that manages a CD Inventory. I will modify the script to use a list of dictionaries instead of a list of lists, add an option to load in existing data, and add an option to delete an entry. I will use the information learned in Module 05 dictionaries and lists in order to do this.

Pseudocode

We were provided the `CDInventory_Starter.py` file in order get us started on this program. This was a version of the CD Inventory that had all the functionality we used in Assignment 04, but also included the Pseudocode for what we needed to modify and add in Assignment 05. Again, the Pseudocode here helps us break down the ask into more manageable pieces and calls out the action items for us. In Figure 1 below I have highlighted the key action items we have for this assignment.

```
CDInventory_Starter.py - Notepad
File Edit Format View Help
# Desc: Starter Script for Assignment 05
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
#-----#

# Declare variables

strChoice = '' # User input
lstTbl = [] # list of lists to hold data
# TODO replace list of lists with list of dicts
lstRow = [] # list of data row
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()

    if strChoice == 'x':
        # 5. Exit the program if the user chooses so
        break
    if strChoice == 'l':
        # TODO Add the functionality of loading existing data
        pass
    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
        strID = input('Enter an ID: ')
        strTitle = input('Enter the CD's Title: ')
        strArtist = input('Enter the Artist's Name: ')
        intID = int(strID)
        lstRow = [intID, strTitle, strArtist]
        lstTbl.append(lstRow)
    elif strChoice == 'i':
        # 3. Display the current data to the user each time the user wants to display the data
        print('ID, CD Title, Artist')
        for row in lstTbl:
            print(*row, sep = ', ')
    elif strChoice == 'd':
        # TODO Add functionality of deleting an entry
        pass
    elif strChoice == 's':
```

Figure 1: Pseudocode

Modifying the Script

We want to update the script to use a list of dictionaries instead of a list of lists, so this is where I started on this assignment. Dictionaries are a new data type for us as they are mapping types instead of sequence types. This means they use key: value pairs to store data. They are initiated by using curly brackets, {}¹. In order to modify the starter script, I first updated the initialization of the lstRow object to dictRow using the curly brackets. Then I went to update the current functionality.

To update the 'add' functionality, I put the user input into a dictionary with the key: value pairs. I identified the keys as 'id', 'title', and 'artist' as simple descriptors of what is held in the value as you can see in Figure 2.

¹ Dirk Biesinger, Foundations of Programming (Python), Module 05 Pages 7-8

```

33     ....elif strChoice == 'a':..# no elif necessary, as this code is only reached if strChoice is not 'exit'
34     .....# 2. Add data to the table (2d-list) each time the user wants to add data
35     .....strID = input('Enter an ID:')
36     .....strTitle = input('Enter the CD's Title:')
37     .....strArtist = input('Enter the Artist's Name:')
38     .....intID = int(strID)
39     .....dictRow = {'id': intID, 'title': strTitle, 'artist': strArtist}
40     .....lstTbl.append(dictRow)

```

Figure 2: Add Using a Dictionary

In order to update the Display Inventory and Save to File functionality, I used the values() function that dictionaries have in Python. This function grabs all the values of a certain dictionary and allows us to display or write the CD data that is held in those values². You can see an example of this in Figure 3 below.

```

41     ....elif strChoice == 'i':
42     .....# 3. Display the current data to the user each time the user wants to display the data
43     .....print('ID\t/ CD Title\t/ Artist\t/')
44     .....for row in lstTbl:
45     .....    print(*row.values(), sep=',')

```

Figure 3: values() Dictionary Function

Also, wanted to determine if we should be adding to a current text file or overwriting the text file. In order to do this, I am assuming that if we load existing data, we want to overwrite the current text file with any changes we make. If we do not load existing data, we will just add any added data to our current file. To implement this, I created a Boolean value loadedData and set it to False. If we use our load existing data functionality, this gets flipped to True. We then check this Boolean when we save our data and it changes how we open the file. If we loaded existing data, we write over the existing data in the file, if not, we just add the data to the end of the file. You can see this implemented in Figure 4 below. This way we do not duplicate data that we loaded in when we save to file.

```

68     ....elif strChoice == 's':
69     .....# 4. Save the data to a text file CDInventory.txt if the user chooses so
70     .....if loadedData:
71     .....    objFile = open(strFileName, 'w')..# overwrite data if we loaded existing data
72     .....else:
73     .....    objFile = open(strFileName, 'a')..# add to file if we did not load existing data
74     .....for row in lstTbl:
75     .....    strRow = ''
76     .....    for item in row.values():
77     .....        strRow += str(item) + ','
78     .....    strRow = strRow[:-1] + '\n'
79     .....    objFile.write(strRow)
80     .....    objFile.close()
81     .....    else:

```

Figure 4: Save Functionality

With that, the starter script has now been updated to use a list of dictionaries and we can move forward with adding more functionality to the program.

Adding Functionality

² Biesinger, Foundations of Programming (Python), Module 05 Page 8

We want to add functionality to be able to load existing data and delete a data entry. Let's start with loading in data.

To load in existing data, I used the method we used in LAB05_B³. We assume the file will comma separated and be in the order of ID,Title,Artist. This allows us to open the file and loop over each row. We split the row by the comma to create a list of the values, then save those values to our keys of a new dictionary, and add this new dictionary to our table.

```
30     ...if strChoice == 'l':
31         .....# TODO Add the functionality of loading existing data
32         .....objFile = open(strFileName, 'r')
33         .....for row in objFile:
34             .....lstRow = row.strip().split(',')
35             .....dicRow = {'id': int(lstRow[0]), 'title': lstRow[1], 'artist': lstRow[2]}
36             .....lstTbl.append(dicRow)
37         .....objFile.close()
```

Figure 5: Load Functionality

Next, we want to delete an entry from our table. To do this, we ask the user to input the ID they would like to delete, then loop through our table and check if that ID is in any of our dictionaries. If it is, we use the remove() function for lists and break out of our loop. If it is not, we notify the user that their inputted ID is not in their inventory.

```
52     ...elif strChoice == 'd':
53         .....# TODO Add functionality of deleting an entry
54         .....deleteID = int(input('What ID would you like to delete?').strip())
55         .....idIn = False
56         .....for row in lstTbl:
57             .....if row['id'] == deleteID:
58                 .....lstTbl.remove(row)
59                 .....idIn = True
60                 .....break
61         .....if not idIn:
62             .....print('This ID is not in your inventory!')
63         .....print()
```

Figure 6: Delete Functionality

Running the Script - Spyder

To test the modifications I made, I first ran the program in Spyder. I wanted to test all functionality, so I loaded in a current file with a couple CDs then displayed, added, deleted, saved and exited. In Figures 7 and 8 below you can see the code running smoothly in Spyder.

³ Biesinger, Foundations of Programming (Python), Module 05 Page 22

```
Console I/O

In [13]: runfile('C:/Users/breinhard/ITFDN110A/Mod_05/Assignment05/CDInventory.py', wdir='C:/Users/breinhard/ITFDN110A/Mod_05/Assignment05')
The Magic CD Inventory

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: 1

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: a

Enter an ID: 3

Enter the CD's Title: Live Forever

Enter the Artist's Name: Barteess Strange
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: i

ID | CD Title | Artist
1 | The Big Wheel | Runrig
2 | Bad | Michael Jackson
3 | Live Forever | Barteess Strange
```

Figure 7: Load, Add, and Display Functionality in Spyder

```
Console I/O

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: d

What ID would you like to delete? 2

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: s

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

1, a, i, d, s or x: x

In [14]: |
```

Figure 8: Delete, Save, and Exit Functionality in Spyder

To ensure the file is running correctly, we also need to check the text is updated accordingly. Here is the text file after the above is run in Spyder where we read in two lines of data, added a third, then deleted a line before saving. It has two lines with one of the original lines having been deleted.



Figure 9: Text File After Spyder Run

Running the Script - Terminal

We also ran the program through the Terminal to finish testing. First, we added a couple of CDs then displayed what we had. Then tried to delete an ID that was not currently in our data then actually deleted an ID that was in our table. Finally, we saved to the file, this time appending to our current file instead of writing over the existing table because we did not load in the current data. Then we exited the program. You can see this successful run in Figures 10 and 11 below.

```

Anaconda Prompt (Anaconda3)
(base) C:\Users\breinhardt>cd C:\Users\breinhardt\ITFDM110A\Mod_05\Assignment05
(base) C:\Users\breinhardt\ITFDM110A\Mod_05\Assignment05>python CDInventory.py
The Magic CD Inventory

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a

Enter an ID: 4
Enter the CD's Title: Everything so Far
Enter the Artist's Name: Pinegrove
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a

Enter an ID: 5
Enter the CD's Title: Sprained Ankle
Enter the Artist's Name: Julien Baker
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: i

ID      | CD Title      | Artist
4       | Everything so Far | Pinegrove
5       | Sprained Ankle  | Julien Baker

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: d

What ID would you like to delete? 1
This ID is not in your inventory!

```

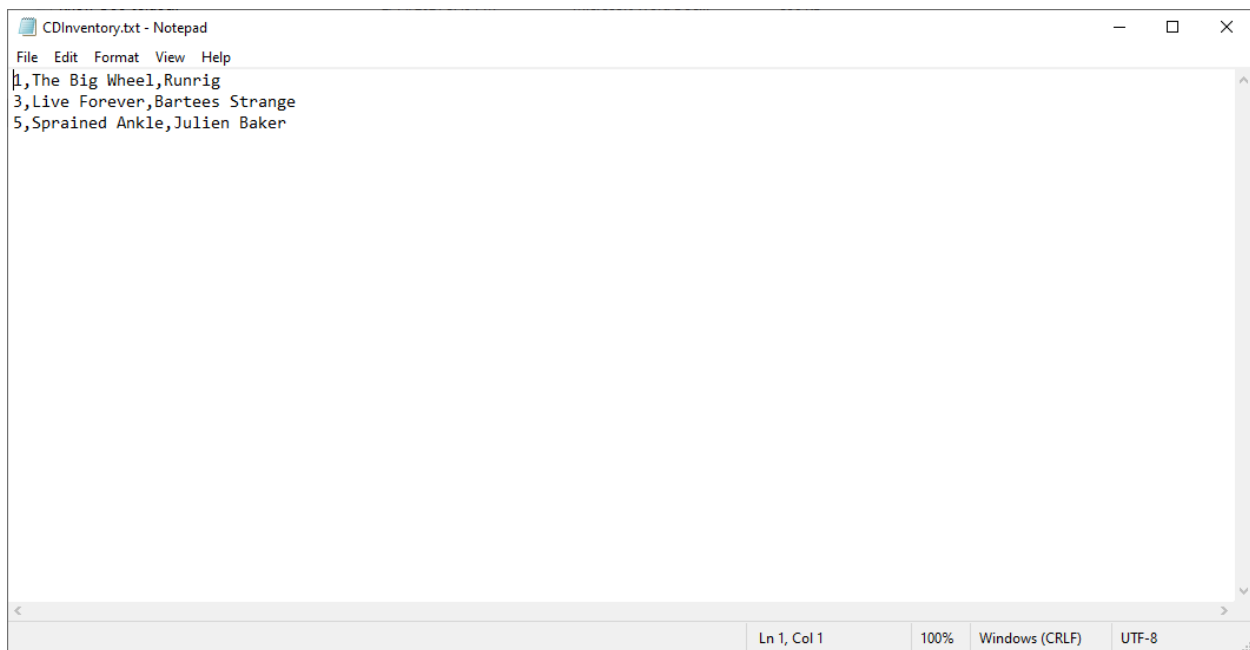
Figure 10: Add, Display, and Delete Functionality in Terminal



```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: d
What ID would you like to delete? 4
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: s
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: x
(base) C:\Users\breinhart\ITFDN110A\Mod_05\Assignment05>
```

Figure 11: Delete, Save, and Exit Functionality in Terminal

After this test we would expect the file from the Spyder test to have one more row added with ID '5'. In Figure 12 below, you can see the file output as expected.



```
CDInventory.txt - Notepad
File Edit Format View Help
1,The Big Wheel,Runrig
3,Live Forever,Bartees Strange
5,Sprained Ankle,Julien Baker
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Figure 12: Final Terminal File Output

Summary

Given the information provided throughout Module 05, I was able to modify this starter script that helps a user manage a CD inventory by adding updating the code to use a list of dictionaries and add functionality to load existing data and delete an entry. I also uploaded the script and this document on GitHub. Here is the link to the repository:

https://github.com/reinhartben/Assignment_05