Andrew Moy

August 7, 2022

IT FDN 110 A Su 22

Assignment 5 (Module 5)

CDInventory.py program with additional modes (Loading, deleting of list entries)

Introduction

For this week’s assignment, we did not cover anything new in class directly but through the course material, we learned about a new data structure known as a dictionary. We were tasked to return to the CDInventory program we already had created last week and to make minor modifications to it using dictionaries. In addition, we were also asked to make some modifications to our program (and therefore the menu) and add some additional options such as deleting entries from our music inventory and loading the inventory directly from a saved external text file.

Because we were using the previous assignment file as a base, most of the groundwork was already laid out and the amount of completely fresh code work required to shape out functions was much lower than last weeks.

Creating the Program

Because we already had the basic framework for the entire program already laid out for us, we only had to modify our inner data lists into a dictionary data structure and write code for two new menu functions.

We already had a list of tables variable established at the start of the program, so we simply had to declare a second variable, a dictionary of rows (abbreviated as dicRow). The difference was that instead of the square brackets to signify a list, we used curly brackets.

Because of our change to a dictionary instead of a list, we had to also make some changes to the add function we already had established in our menu. When we had a list to represent the inner data (rows), the program was set up as such:

lstRow = [intID, strTitle, strArtist]

lstTbl.append(lstRow)

Now that we have a dictionary, we could set our dicRow with the same three variables, but also added callable keys to represent the ID, Title, and Artist as strings. We also changed the append function to refer to dicRow. Now that that was tidied up, we could then move on to the main core of the assignment; add functionality of deleting an entry and loading existing data.

I focused on the loading function first, as the task asked from it appeared to be a combination of the reading and displaying data functions we went over. We had to open the text file our file path referred to, and then go through all the rows contained within it. We established a new variable, a temporary list of rows inside of a loop that read our text file rows, and then had it split the values in each row into a list by the comma separators:

1,Song Title,Artist Name in the text file would become [‘1’, ‘Song Title’, ‘Artist Name’]

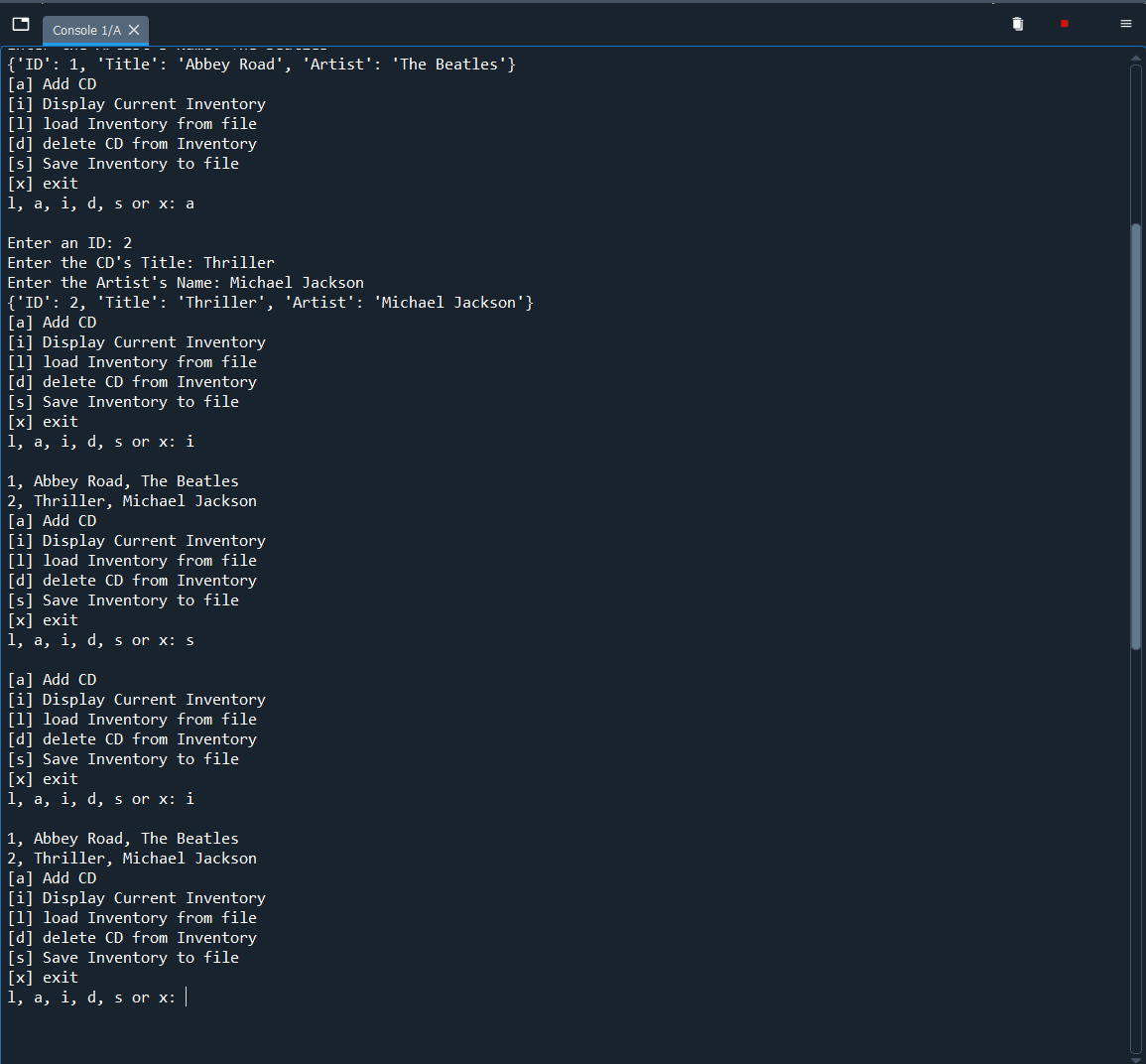
Then, we took our previously unutilized dictionary of rows (dicRow) and stored our previous list into it with keys to identify each value. Finaly, we could then append the dictionary row by row into our main list function while adding a check function to print out the results.

For our deletion function, I initially assumed that we were supposed to write something to delete entries in our text file, and that was exceedingly difficult to think of a starting path into tackling that problem given what we were given as learning material did not address anything related to that. After I went back over the instructions, I switched to focusing on making something that would delete an entry from our current inventory. To do so, I started with creating a variable to read an input asking for the entry to be deleted. I omitted asking for the ID number because I assumed that if we had a duplicate ID/index number in the list, going by that would end up deleting multiple entries and be far less specific than trying to delete multiple entries by artist name.

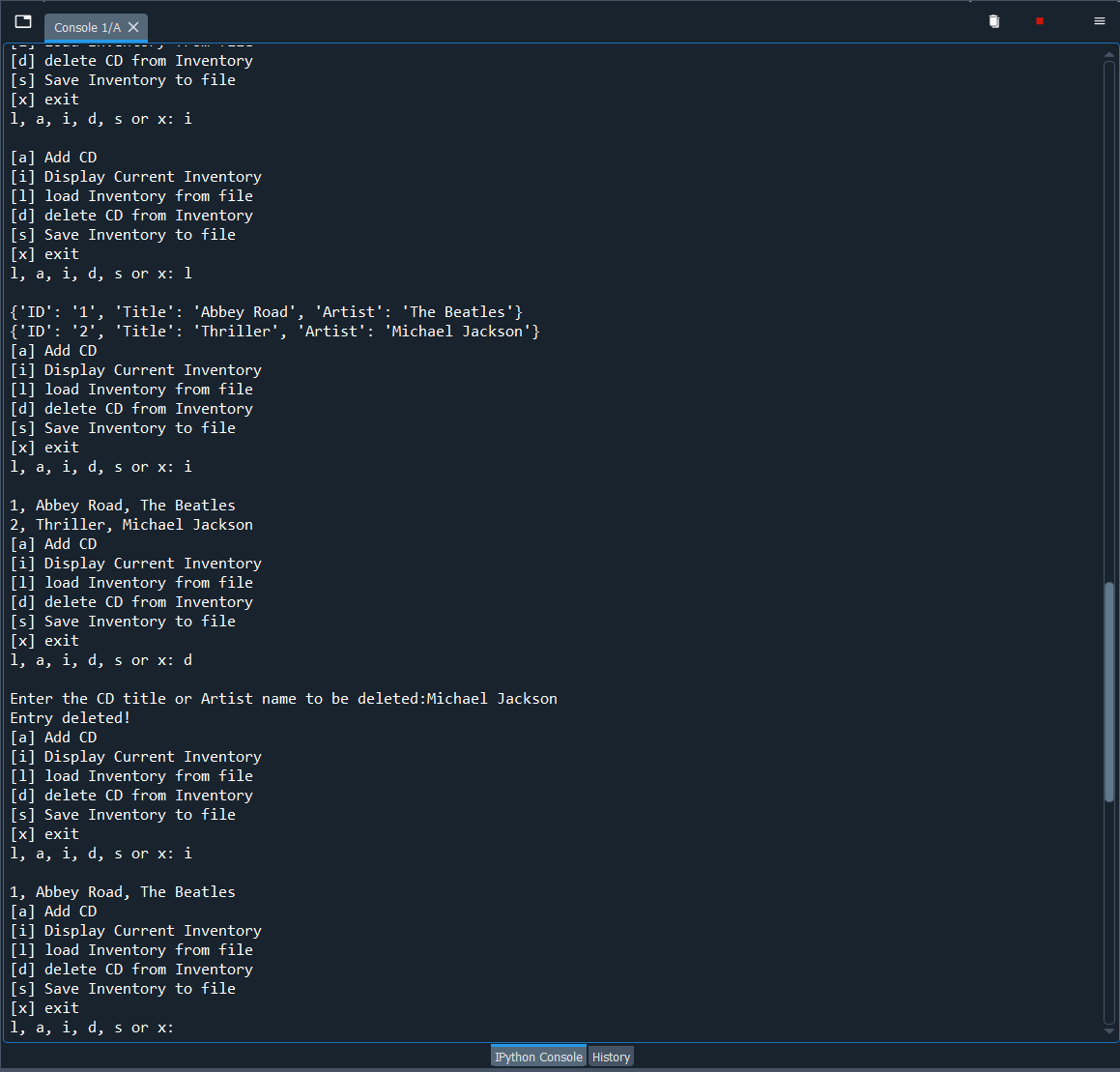
I created a for loop to read the contents of the current inventory (loaded or not), but I used an iterable number that would cover the length of the list, row by row. Next, my if logic statement would then use the get function to read the Title and Artist keys and see if they matched the input exactly, and then had it delete said row. Initially, I used an if statement with an or to separate my two boolean expressions, but for some reason my function would always delete a row despite me entering a non-matching input. After some external research, I realized that if lstTbl[i].get('Title') was returning true every single time my loop went through because it was asking if there was a non-empty list instead of the expected attempt to match with my input. I had to change my if function specifically to attempt to check my input delChoice twice. I also then added a print function to identify that it was working as intended.

Summary

Once I had finished the program, I loaded it for testing directly in Spyder’s console. I first tried adding new entries to the initially blank list, then checking the inventory list. I then saved them and then checked the text file to ensure that they were saved.



Next, I ended the program and then reloaded it with a blank inventory state to test loading my previously saved text file entries. Once I verified they were loaded successfully, I tried to delete entries using one of the two options I gave myself, song title or artist name.



I then loaded up the program in terminal and tried repeating my previous testing and reconfirmed it was working.

