Nicco Fortes

11/13/22

Foundations of Programming

Module 5

Module 5

Intro

This week we are expanding on the different types of data we work with by introducing dictionaries. We will expand on the CD Inventory program with our new knowledge of dictionaries. And lastly we will be taking a look at Github and setting up an account.

Module 5

In this module we finished up some items regarding lists and were introduced to dictionaries. With the "Key:Value" structure of dictionaries I can see how this data type could be very useful. After working through the last assignment and understanding the structure of that code, I found it somewhat easier to digest this weeks topics because I have already spent a good amount of time pondering how I could write that code. So as I worked through this module I found myself going back to some thoughts/roadblocks I had last week and how this new data type would fit into the code.

Assignment 5

In this assignment I ran into a number of issues while writing my code. Below I will give a brief description of these issues under

Importing data from Text File

For this section I first used the print function to see what form the data is when it is imported to the script. From there I knew I could put it into a list of list, but I would need to convert to a list of dictionaries. To do this I used a "for" loop in which for each list within the list I would create a new dictionary by assigning keys to the corresponding index in the list. From there the new dictionary is appended to the main list of dictionaries.

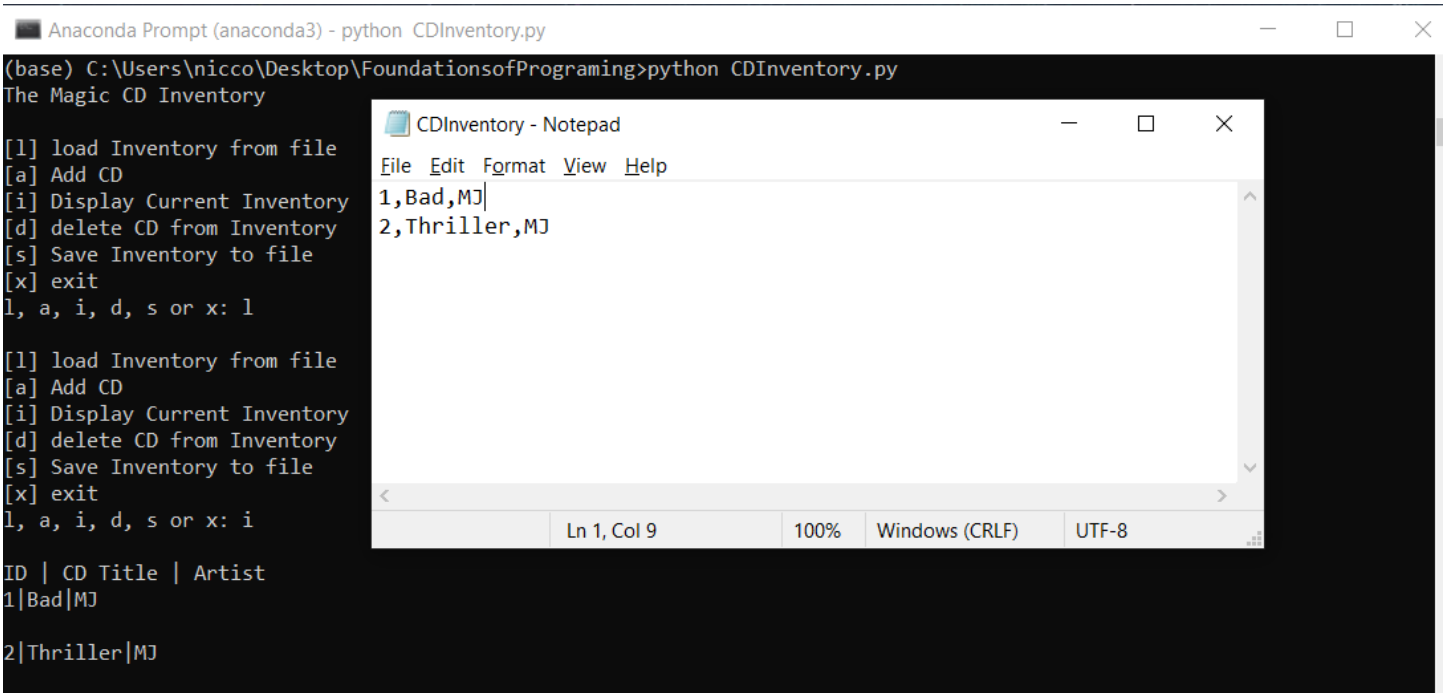

Figure 1: Importing data from tet file in prompt.

Deleting Items

In order to delete entries I utilized a "for" loop to look through my list of dictionaries and pull the value corresponding to the "ID". This part of the code is limited as it assumes that each ID number that corresponds to a CD is unique. I played around with accounting for these cases but ended up setting with asking the user to keep this in mind when entering in data. This could be improved.

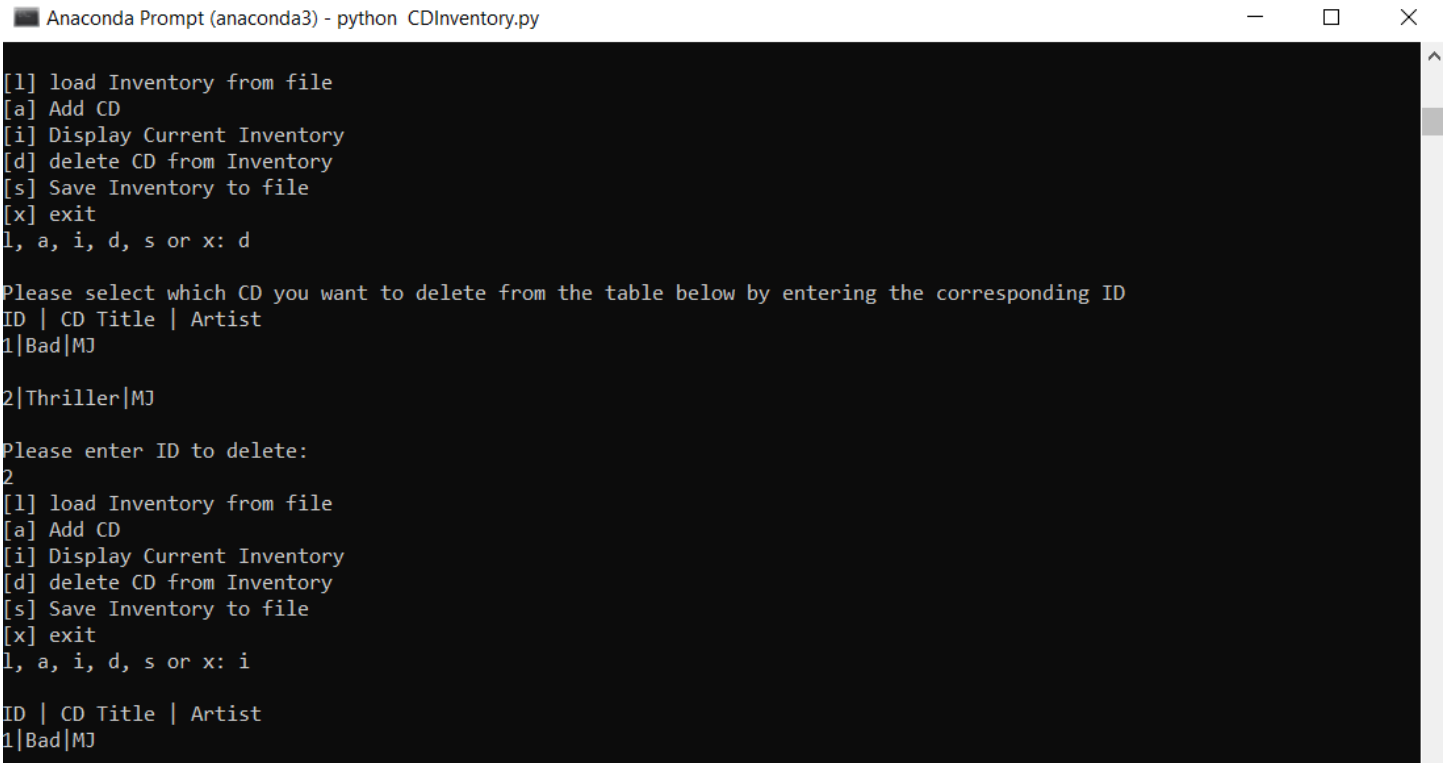

Figure 2: Deleting entry from prompt

Gathering input data from user

My first approach was to write the input directly into the dictionary row index, ie dicRow['ID']= input('Enter an ID:'). I opted for this method because I though it would make my code more readable. However this ended up with the list of dictionaries over writing all entries to the last input of the user's. In other works it would update all entries in the list to be equal to the last input. I realized this was because the list was referencing dicRow and updating the indexies also updated the indexes in the list. To get around this I gathered input data into variables and overwrite dicRow then append it to the list of dictionaries.

```
    elif strChoice == 'a':
        #Add data to the table (2d-list) each time the user wants to add data

        #Code needs ID numbers to be unique to work. Code trusts user to input unique numbers
        #does not currently have functionallity to check against current list.
        strID = input('Enter an ID Number (must be unique): ')
        strTitle = input('Enter the CD\'s Title: ')
        strArtist = input('Enter the Artist\'s Name: ')
        intID = int(strID)
        dicRow = {'ID':intID,'Title':strTitle,'Artist':strArtist}
        Tbl1.append(dicRow)
```

Figure 3: Code to gather user input in Spyder

Writing Data into Text

I also struggled to write the memory in the script into the text file, which I knew off the bat was a data type issue but took some time for me to work through the problem. In order to make a code that made sense to me I took the list of dictionaries and converted it to a list of lists, taking out the keys. Then I was able to take the same code/process from last week and write the data into the text document.

```
elif strChoice == 's':
    # Save the data to a text file CDInventory.txt if the user chooses so
    objFile = open(strFileName, 'a')
    for row in Tbl1:
        list1= list(row.values())
        Tbl2.append(list1)
        for row in Tbl2:
            strRow = ''
            for item in row:
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
        objFile.write(strRow)
    objFile.close()

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

Figure 4: Code to export data to text file in Spyder

Summary

My main take away from this week is the importance of data types. I spent most of my time overthinking the coding problem before taking a step back and realizing that most of my script is just converting back and forth from one data type to another defending on what functionality I am working towards. Using Spyder and having the ability to see the data types easily after running the script became the way I knew I was staying on the correct course.

GitHub Link

https://github.com/NiccoFortes/Assignment_05

Appendix/Search

loading list from text file python

Converting list to dictionary

Converting Dictionary to list