Ljubica Trpcheva Smileski

November 13, 2022

Foundations of Programming, Python

Assignment 05

Modifying the Python CD Inventory program

Introduction

In this assignment I will explain the steps I used to modify and extend the provided Python code in CDInventory_Starter.py.

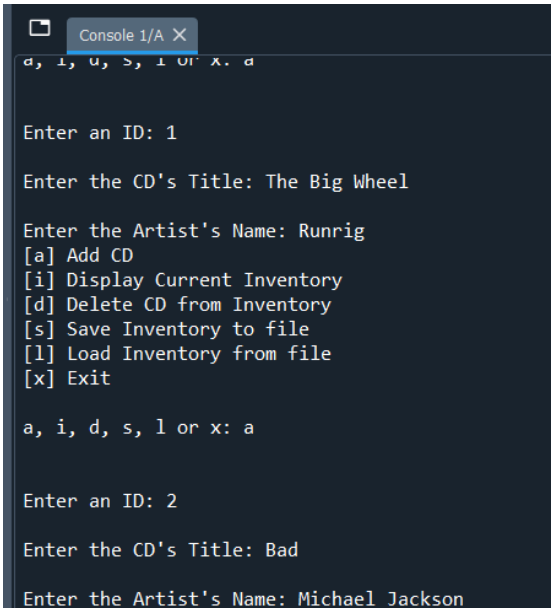
Modifying and extending Python Script and running in Spyder IDE

Working with dictionaries

This week Assignment05 was modifying the given 2D data structure while using dictionaries instead of lists. First, I spent some time reviewing the code, to make sure I understood what the code was doing, then to be able to modify it. The given starter code was very similar to my previous assignment. It was a CD inventory program that contains a menu with these options: adding cd, displaying current inventory, deleting cd from inventory, saving inventory to a .txt file, loading/reading inventory from a .txt file and exit. I run the given code in Spyder to make sure the menu is working properly. I was very careful not to delete any of the previous programmer's notes. Next step that I took was changing the lists with dictionaries, so when a user is adding data (id, title and artist) that data saves in dictionary (dicRow, that is my dictionary row), and each dictionary saves into list (lstTable, that is my list of dictionaries). See Figure1.

```
strChoice = '' # User input
lstTbl = []  # list of lists to hold data
# TODO replace list of lists with list of dicts
lisRow = {}  # dictionary of data row
dicRow = {}  # dictionary of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

# Get user Input
print('\nThe Magic CD Inventory\n')
while True:

#  Display menu allowing the user to choose:
    print('[a] Add CD\n[i] Display Current Inventory\n[d] Delete CD from Inventory')
    print('[s] Save Inventory to file\n[l] Load Inventory from file\n[x] Exit')
    strChoice = input('a, i, d, s, l or x: ').lower()  # convert choice to lower case at time of input
    print()

# 6. Exit the program if the user chooses so
    if strChoice == 'x':
        break

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

Figure 1 – Add data to a table using dictionary

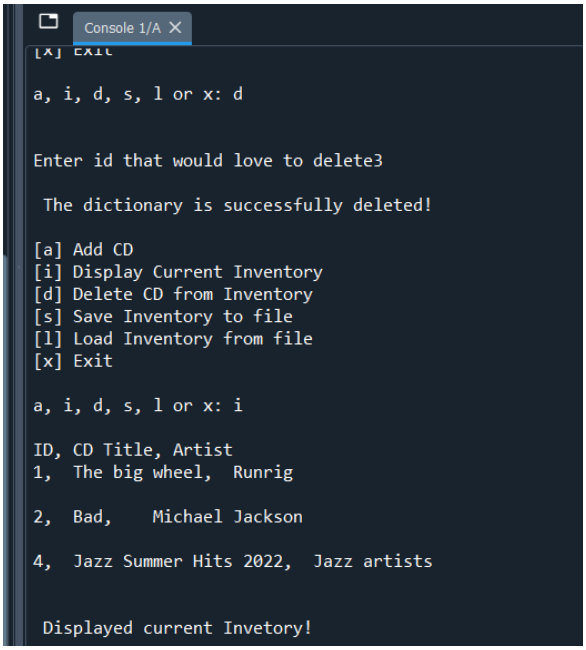


Delete CD from Inventory

For the menu option 'Delete CD from inventory', I knew that I could use For Loop to read the lstTbl , to go through each dictionary in the table, to find the key you would love to delete, and then delete the dictionary (in my case dicRow). Adding input() function was easy, where user can add id for delete. The real struggle was how to remove the dictionary from the list of dictionaries. After a little research I found a method using del in For Loop, this is the article, https://www.geeksforgeeks.org/python-removing-dictionary-from-list-of-dictionaries/

```
# 3. Delete CD from Inventory
    elif strChoice == 'd':
        # TODO Add functionality of deleting an entry
        deleteId = int(input('Enter id that would love to delete'))
        # using del + loop, deleting dictionary in list
        for row in range(len(lstTbl)):
            if lstTbl[row]['id'] == deleteId:
                del lstTbl[row]
                break
        print('\n The dictionary is successfully deleted! \n', lstTbl)
```

Figure 2 – Delete CD from inventory

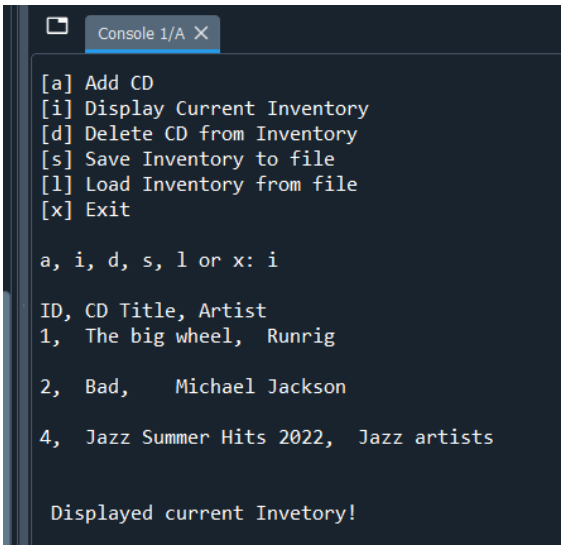


Displaying CD inventory program

Displaying the current data to the user each time the user wants to display data was easy to implement, just with using one For Loop and printing each dictionary key. See Figure 3

```
# 2. 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(str(row['id']) + ',\t' + row['title'] + ',\t' + row['artist'] + '\n')
        print('\n Displayed current Invetory! \n')
```

Figure 3 – Display the current data



Write and read data to/from .txt file

For writing data to a CDInventory.txt I used the same logic that I used for displaying data, just adding the commands for writing to a .txt file: open() function, write() method and close() method.

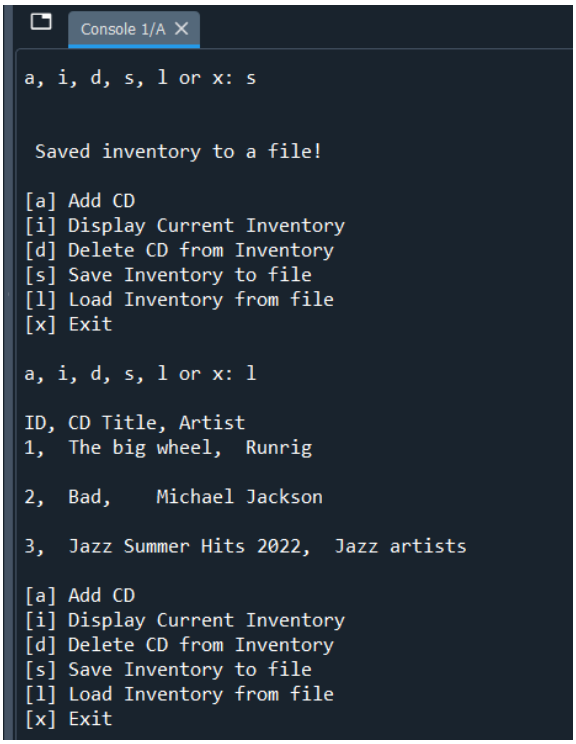
For loading inventory data from a file I used strip() and split() methods. The strip() method in python removes whitespaces from the beginning or end of a string, and the split() method splits a string , than add to dictionary. Then from a dictionary the rows are appending to a table.

See Figure 4.

```
# 4. 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:
            objFile.write(str(row['id']) + ',' + row['title'] + ',' + row['artist'] + '\n')
        objFile.close()
        print('\n Saved inventory to a file! \n')

# 5. Load inventory from a file (read from a file)
    elif strChoice == 'l':
    # TODO Add the functionality of loading existing data
        lstTbl.clear()
        objFile = open(strFileName, 'r')
        for row in objFile:
            lstRow = row.strip().split(',')
            dicRow = {'id': lstRow[0], 'title': lstRow[1], 'artist':lstRow[2]}
            lstTbl.append(dicRow)
            #print(dicRow)
        objFile.close()
        print('ID, CD Title, Artist')
        for row in lstTbl:
                print(str(row['id']) + ',\t' + row['title'] + ',\t' + row['artist'] + '\n')

    else:
        print('Not a valid choice! Please choose either a, i, d, s, l or x!')
```

Figure 4 – Write data to CDInventory.txt and read data from CDInventory.txt

Figure 5 – The data is successfully saved into CDInventory.txt



Summary

This is just one way of modifying and upgrading the CD inventory file. I found it very interesting and helpful using dictionaries, because they allow us to associate a value to a unique key, and then quickly access this value. I am glad that I was able to successfully modify the code while using dictionaries.

Appendix

CDInventory.py using dictionaries

```
#------------------------------------------#
# Title: CDInventory.py
# Desc: Starter Script for Assignment 05
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# LSmileski, 2022-Nov-12, Edited File
#------------------------------------------#

# Declare variabls

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
# TODO replace list of lists with list of dicts
lisRow = {}  # dictionary of data row
dicRow = {}  # dictionary of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

# Get user Input
print('\nThe Magic CD Inventory\n')
while True:

#  Display menu allowing the user to choose:
    print('[a] Add CD\n[i] Display Current Inventory\n[d] Delete CD from Inventory')
    print('[s] Save Inventory to file\n[l] Load Inventory from file\n[x] Exit')
```

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

# 6. Exit the program if the user chooses so
    if strChoice == 'x':
        break

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

# 2. 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(str(row['id']) + ',\t' + row['title'] + ',\t' + row['artist'] + '\n')
        print('\n Displayed current Invetory! \n')

# 3. Delete CD from Inventory
    elif strChoice == 'd':
        # TODO Add functionality of deleting an entry
        deleteId = int(input('Enter id that would love to delete'))
        # using del + loop, deleting dictionary in list
        for row in range(len(lstTbl)):
            if lstTbl[row]['id'] == deleteId:
                del lstTbl[row]
                break
        print('\n The dictionary is successfully deleted! \n')

# 4. 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:
            objFile.write(str(row['id']) + ',' + row['title'] + ',' + row['artist'] + '\n')
        objFile.close()
        print('\n Saved inventory to a file! \n')

# 5. Load inventory from a file (read from a file)
    elif strChoice == 'l':
    # TODO Add the functionality of loading existing data
        lstTbl.clear()
        objFile = open(strFileName, 'r')
        for row in objFile:
            lstRow = row.strip().split(',')
            dicRow = {'id': lstRow[0], 'title': lstRow[1], 'artist':lstRow[2]}
            lstTbl.append(dicRow)
            #print(dicRow)
        objFile.close()
        print('ID, CD Title, Artist')
        for row in lstTbl:
            print(str(row['id']) + ',\t' + row['title'] + ',\t' + row['artist'] + '\n')

    else:
        print('Not a valid choice! Please choose either a, i, d, s, l or x!')
```