< 2021-Aug-8th >

< IT FDN 110 B Su 21 >

< Assignment_05 >

Assignment_05

Introduction

The script developed uses a simple, menu-driven interface allowing the user to load CD inventory data from file, enter CD data, view the inputted inventory data, delete CD data, save the inventory to a data file, and exit out of the program. The program explores the script's ability to use the list and dictionary methods to create and maintain a list / dictionary every time user enters the data.

Procedure

Setting Up the while loop

In the while condition, a Boolean True operator is used to control the loop. Input() is used to capture the data from the user. The variable *strChoice* represents user's choice from the menu. *lstTbl* is a list that will contain the multiple lists generated through user's inputted data.

Storing data – *Hello "Dictionaries"*

Through the use of append(), dictionary method adds key:value pairs to the end of local dictionary variable called *userInputDict.* The dictionaries replace the index with key and sequence to create a 2D table. Variables for storing the user inputted "ID", "title", and "artist" are then assigned to *userInputDict*. See listing 1.

```
41          # 2. Add data to the table (2d-list) each time the user wants to add data
42          userInputDict = {}
43          strID = input('Enter an ID: ')
44          strTitle = input('Enter the CD\'s Title: ')
45          strArtist = input('Enter the Artist\'s Name: ')
46          intID = int(strID)
47          userInputDict = {'ID': intID, 'title': strTitle, 'artist': strArtist}
48          lstTbl.append(userInputDict)
49          pass
```

Listing 1: Appending userInputDict and lstTbl using append()

Loading data – *"For" Loop*

Through a For loop, data from an external .txt file is first loaded into a list called *lstRow* and then into a dictionary called *dicRow1*. This makes sure we have data added to memory to manipulate further. The data from *dIicRow1* is further appended to *lstTbl* to build the inventory 2D list. The local dictionary initializing of *dicRow1* makes sure we there's no reminiscence of previous data row left when moving to following row. See listing 2.

```
28      if strChoice == 'l':
29          # functionality of loading existing data
30          objFile = open(strFileName, 'r')
31          for row in objFile:
32              dicRow1 = {}
33              lstRow = row.strip().split(',')
34              dicRow1 = {'ID': int(lstRow[0]), 'title': lstRow[1], 'artist': lstRow[2]}
35              lstTbl.append(dicRow1)
36              print(lstRow)
37          objFile.close()
38          pass
```

Listing 2: Loading lists from file to memory

Deleting data – Meshing "List" & "Dictionary" together!

By referring to the list's index (inputted by user) the corresponding dictionary can be deleted from the memory. The *userId* is compared with the "ID" in the dictionary *dicRow.* The program checks to see if the user Id is actually in the dictionary with "==" operator. And upon detection, deletes the dictionary item from the list, lstTbl, by finding the correct index. See listing 3.

```
56      elif strChoice == 'd':
57          # functionality of deleting an entry
58          userId = input("what ID do you want to delete? :")
59          for dicRow in lstTbl:
60              if int(userId) == dicRow['ID']:
61                  del lstTbl[lstTbl.index(dicRow)]
62                  print("\ndicRow got deleted", userId)
63              else:
64                  print("\nAction aborted", userId, "doesn't exist in the dictionary.")
65          pass
```

Listing 3: Deleting a dictionary in memory

Summary

In this assignment, I learned how to read and execute changes to the code snippet received from someone else. Great learning experience on how to maintain code's functional integrity while introducing changes in a professional environment. I learned how to pseudocode an algorithm and then refine the code to match Python's syntax. Next, I learned how to create and modify dictionaries.

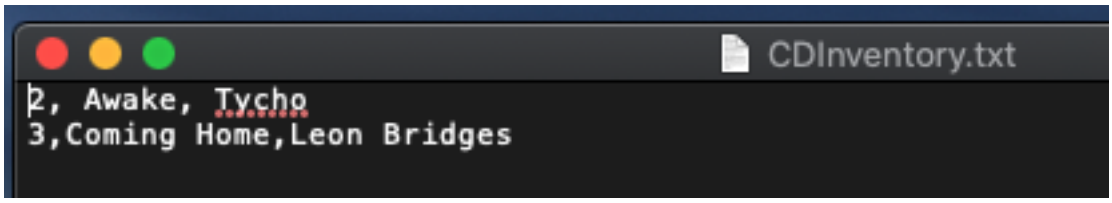
Appendix

```python
#------------------------------------------#
# Title: CDInventory.py
# Desc: Mod. of starter script to use dictionaries (Assignment_05)
# Change Log: (Who, When, What)
# SGupta, 2021-Aug-08, Created File
#------------------------------------------#

# Declare variabls

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # dict 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':
        # functionality of loading existing data
        objFile = open(strFileName, 'r')
        for row in objFile:
            dicRow1 = {}
            lstRow = row.strip().split(',')
            dicRow1 = {'ID': int(lstRow[0]), 'title': lstRow[1], 'artist': lstRow[2]}
            lstTbl.append(dicRow1)
            print(lstRow)
        objFile.close()
        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
        userInputDict = {}
        strID = input('Enter an ID: ')
        strTitle = input('Enter the CD\'s Title: ')
        strArtist = input('Enter the Artist\'s Name: ')
        intID = int(strID)
        userInputDict = {'ID': intID, 'title': strTitle, 'artist': strArtist}
        lstTbl.append(userInputDict)
        pass
    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)
        pass
    elif strChoice == 'd':
        # functionality of deleting an entry
        userId = input("what ID do you want to delete? : ")
        for dicRow in lstTbl:
            if int(userId) == dicRow['ID']:
                del lstTbl[lstTbl.index(dicRow)]
                print("\ndicRow got deleted", userId)
        pass
    elif strChoice == 's':
        # 4. Save the data to a text file CDInventory.txt if the user chooses so
        objFile = open(strFileName, 'w')
        for row in lstTbl:
            strRow = ''
            for item in row.values():
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
            objFile.write(strRow)
        objFile.close()
    else:
        print('Please choose either l, a, i, d, s or x!')

```

Listing 4: CDInventory.py

Listing 5: Output from working script in IDLE - (1/2)

Listing 6: Output from working script in IDLE – (2/2)



Listing 7: Output in CDInventory.txt