Josephine Gnanaraj
02\15\2021
Foundations Of Programming: Python
Assignment_05

Introduction

This week's module has been a continuation of the last module where we explored writing data to files and Multidimensional data constructs. We will be working with files\lists and using dictionaries instead of lists for the rows.

LAB 05-A: Working with Files and Lists

Lists can contain other lists as elements. A typical use of such nested (or multidimensional) lists is to represent tables of values consisting of information arranged in rows and columns.

Python supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. Here, we create two-dimensional lists to read and write data from list to file.

Completed the starter code to read and write data from list to file and Compiled to verify output:

```
#------------------------------------------------------------------#
# Title: Lab05_A_starter.py
# Desc: Lab05-A starter script for Assignment 05
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
#jgnanaraj, 2021-feb-14, Modified File
#------------------------------------------------------------------#


# Declare variabls


strChoice = '' # User input
lstTbl = []  # list of lists to hold data
lstAdd = {}
lstRow = []  # list of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object
cd_inventory = []

# Get user Input
print('Write or Read file data.')

while True:

    print('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
    print('[d] display data\n[exit] to quit')

    strChoice = input('a, w, r, d, or exit: ').lower()  # convert choice to lower case at time of input
    print('\n')

    if strChoice == 'exit':
        break

    if strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'

        # Add data to list in memory
          print("\n Add CD Details:")
          print('\n')

          title = input("CD Title:")
          name = input("Artist Name:")

          new_cd = [title, name]
          cd_inventory.append(new_cd)

```

Figure 1 - Screen capture of Lab05-A

```
    elif strChoice == 'w':
        # List to File
        dataFile = open('CDInventory.txt', 'a')

        # TODO add code here to write from in-memory list to file
        for cd in cd_inventory:
         dataFile.write("{}, {}\n".format(cd[0], cd[1]))
        dataFile.close()
        pass


    elif strChoice == 'r':

        # TODO read the file line by line into in-memory list.
        dataFile = open('CDInventory.txt', 'r')
        pass


    elif strChoice == 'd':

        # Display data
        print("\n Display CD Details added")
        print('\n')
        print('Artist', 'Title')

        for cd in cd_inventory:
         print("{}, {}".format(cd[0], cd[1]))
        pass


    else:
        print('Please choose either a, w, r or exit!')
```

Figure 2 - Screen capture of Lab05-A

```
Console 7/A


[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: r



[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: d


 Display CD Details added


Artist Title
Tilte 01, Name 01
Title 02, Name 02

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: exit


In [2]:
```

Figure 3 - Screen capture of compiling Lab05-A

untitled6.py* untitled5.py* CDInventory.txt

```
1  Tilte 01, Name 01
2  Title 02, Name 02
3
```

Figure 4 - Screen capture of CDInventory.txt Lab05-A

LAB 05-B: Working with List of Dictionaries

Dictionary is like any element in a list. Therefore, you can access each dictionary of the list using index. A Python list is an ordered sequence of objects, whereas dictionaries are unordered. They are similar to sequence types but replace the index and storing in sequence by storing key: value pairs. This means instead of the index, the programmer defines a key under which the values are being stored and accessed. Dictionaries use the braces {} operator. The full syntax is {key: value}. Key:Value pairs are separated by commas.

Dictionary keys are a lot like named columns in a spread sheet or database, and the dictionary is a row of data. Like creating a list of lists to form a 2D list to resemble a 2D table, dictionaries can be inserted into lists to create a list of dictionaries 2D table.

```
73      elif strChoice == 'd':
74          entry_Del = int(input('Enter the ID of the entry you want to delete: '))
75
76          for row in tblDict:
77              if entry_Del == row['CD ID']:
78                  tblDict.remove(rowDict)
79          pass
80
81      elif strChoice == 's':
82          # 4. Save the data to a text file CDInventory.txt if the user chooses so
83          objFile=open(strFileName,'w')
84          for row in tblDict:
85              strRow=''
86              for item in row.values():
87                  strRow=strRow+str(item)+','
88              strRow=strRow[:-1]+'\n'
89              objFile.write(strRow)
90          objFile.close()
91          print('CD inventory saved to file.\n')
92          pass
```

Figure 6 - Screen capture of Lab05-B showing deletion of entry\save the data to text file

Modified the code from Lab 05-A to use dictionaries instead of lists for the rows. Compiled to verify output:

```
#------------------------------------------------------------------------------------------#

# Title: CDInventory.py

# Desc: Starter Script for Assignment 05

# Change Log: (Who, When, What)

# DBiesinger, 2030-Jan-01, Created File

# jgnanaraj, 2021-Feb-14, Modified File and added requirements

#------------------------------------------------------------------------------------------#

# Declare variables


strChoice = '' # User input
tblDict = []  # list of dictionaries to hold data
rowDict = {} # list of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

# Get user Input
print('\nThe Magic CD Inventory\n')
while True:
    # 1. Display menu allowing the user to choose:
    print('\n[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':
        # Added the functionality of loading existing data
        tblDict=[]
        # read the file line by line into in-memory dictionary
        # keep adding by appending the dictionary to a list
        objFile=open(strFileName,'r')
        for row in objFile:
            lstRow=row.strip().split(',')
            rowDict={'CD ID':int(lstRow[0]), 'CD Title':lstRow[1],'Artist':lstRow[2]}
            tblDict.append(rowDict)
        objFile.close()
        pass

    if 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]
        rowDict={'CD ID':intID,'CD Title':strTitle,'Artist':strArtist}
        tblDict.append(rowDict)
        pass
```

Figure 5 - Screen capture of Lab05-B

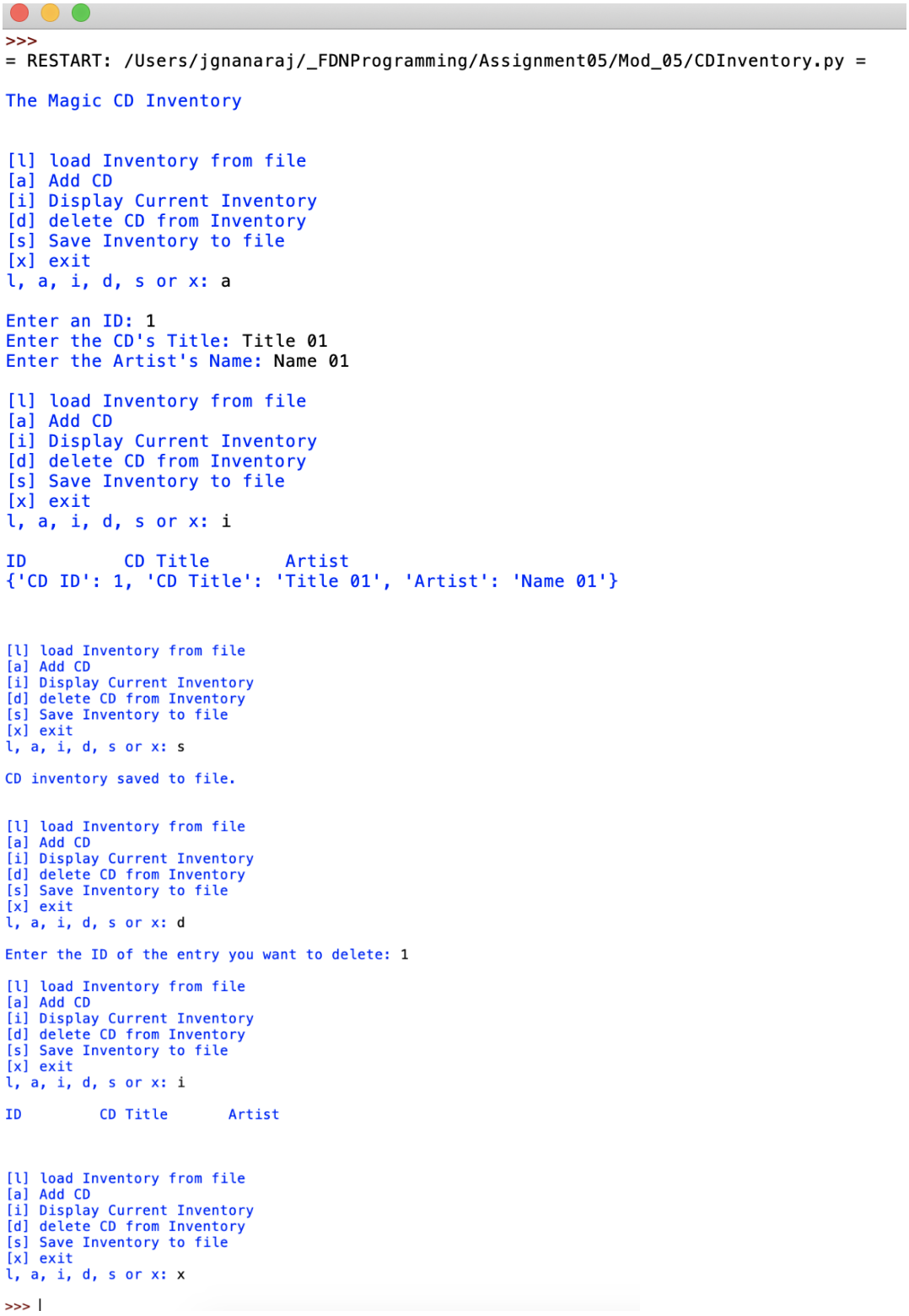

Figure 7 - Screen capture of script compiled in Terminal window for Lab05-B

Summary

In this assignment\module, we continued with lists, covered Dictionaries and file read / write access. We created a script template and looked at ways to improve our scripts. We got a brief introduction to what git and GitHub are and created a first repository. We used Spyder as our IDE.

References:

https://medium.com/better-programming/10-ways-to-convert-lists-to-dictionaries-in-python-d2c728d2aeb8
https://www.geeksforgeeks.org/file-handling-python/
https://www.w3schools.com/python/python_file_handling.asp
https://stackoverflow.com/questions/30522982/list-with-many-dictionaries-vs-dictionary-with-few-lists
FDN_Py_Module_05.pdf