Sheena Harms

Nov. 12, 2022

IT FDN 110 B

Assignment 05

Dictionaries for Cataloging

# Introduction

This week I re-built the CD Inventory code to include dictionaries and a few other commands, including delete. I will outline how my code operates and some of the bigger hurdles I faced in building it. Furthermore, I will talk about the code’s shortcomings and potential areas for improvement.

# From Lists to Dictionaries

I began by using the neatly organized shell provided by the professor. Pseudocode laid out the necessary steps in conjunction with the user’s menu.

Initially I worked through each strChoice to ensure it functioned. The code begins with an if statement that breaks if the user chooses to exit. The next if statement allows the user to input the artist and album names. As always, I reiterated and improved this code over the course of a few days. In one of the later iterations I changed the lstRow list to a dictionary, which takes the artist and title as key value pairs. Those pairs are then appended to the list lstTbl.

The other strChoices are elif statements. The w input writes the lstTbl rows to the CDInventory.txt file. The r input reads rows from the text file and appends them to lstTbl. I added x later, which is designed to give the user the ability to delete a record. Currently, the user may only input an artist’s name to delete the associated key value pair from the lstTbl. If someone wanted to delete multiple records associated with one artist, they would need to repeatedly delete them. The delete choice works but is the most limiting of the menu choices and would be potentially frustrating due to the lack of control and transparency it gives a user. Finally, for the display (d) command I wrote a simple print statement which printed the contents of the lstTbl to the user.

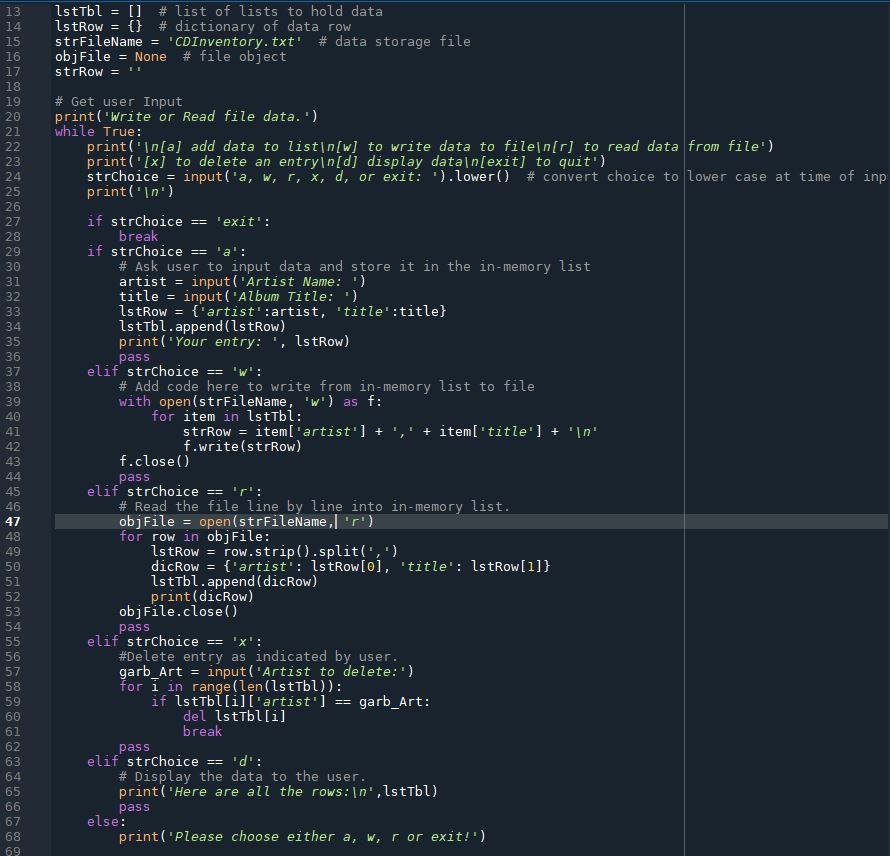


Figure : CD Inventory code, in Spyder.

# Areas for Improvement

In future, it would be helpful to have the write (w) command overwrite the text file, to prevent duplication of records from using the write and read commands repeatedly.

It would also be helpful to allow deletion by artist and title, rather than just artist. As it is currently set up you may only delete one artist entry at a time.

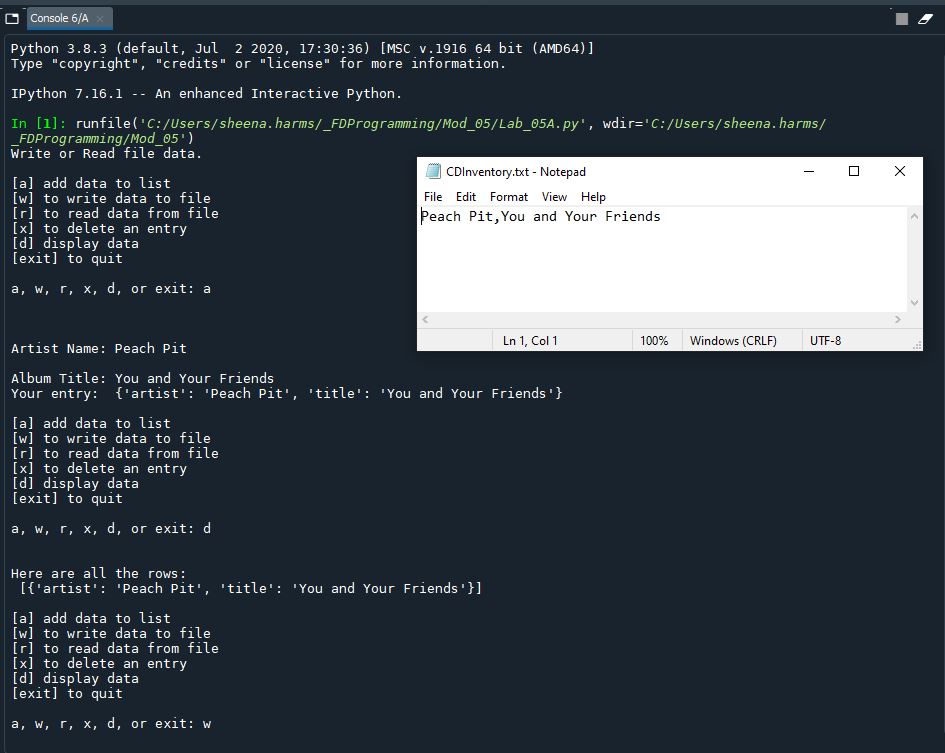


Figure 2 – CD Inventory code running in Spyder.

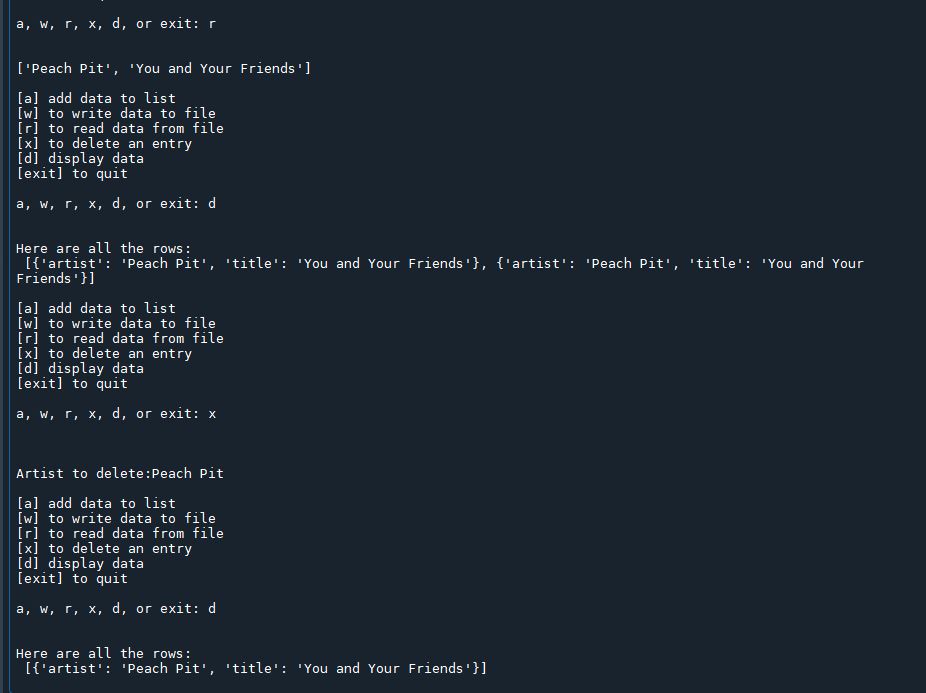


Figure 3 – CD Inventory code running in Spyder, cont’d..

# Summary

Assignment 5 drives home the concepts of where data is being stored when you’ve written to memory rather than saved data to a text file. This assignment also teases out the fine differences between dictionaries and lists in Python.

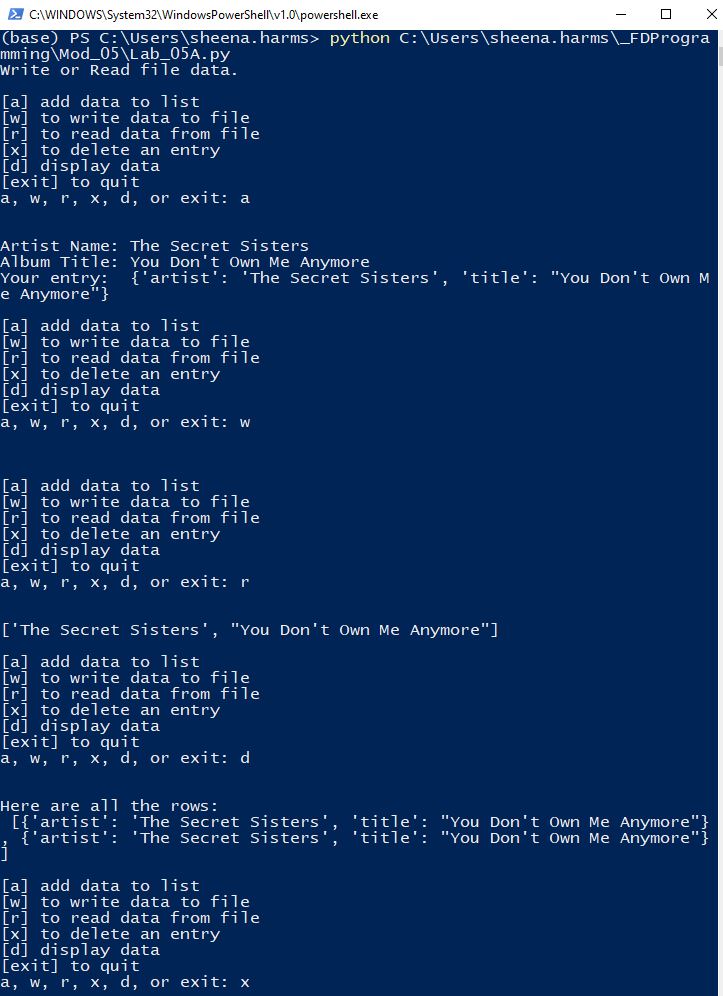


Figure 4: CD Inventory script in terminal.

# Appendix

## Listing Assignment\_05.py

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | *#------------------------------------------#*  *# Title: Assignment\_05.py*  *# Desc: Lab05-A starter script edited*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, Created File*  *# SHarms, 2022-Nov-10, Edited File*  *# SHarms, 2022-Nov-12, Edited File*  *#------------------------------------------#*  *# Declare variables*  *strChoice = '' # User input*  *lstTbl = [] # list of lists to hold data*  *lstRow = {} # dictionary of data row*  *strFileName = 'CDInventory.txt' # data storage file*  *objFile = None # file object*  *strRow = ''*  *# 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('[x] to delete an entry\n[d] display data\n[exit] to quit')*  *strChoice = input('a, w, r, x, d, or exit: ').lower() # convert choice to lower case at time of input*  *print('\n')*  *if strChoice == 'exit':*  *break*  *if strChoice == 'a':*  *# Ask user to input data and store it in the in-memory list*  *artist = input('Artist Name: ')*  *title = input('Album Title: ')*  *lstRow = {'artist':artist, 'title':title}*  *lstTbl.append(lstRow)*  *print('Your entry: ', lstRow)*  *pass*  *elif strChoice == 'w':*  *# Add code here to write from in-memory list to file*  *with open(strFileName, 'w') as f:*  *for item in lstTbl:*  *strRow = item['artist'] + ',' + item['title'] + '\n'*  *f.write(strRow)*  *f.close()*  *pass*  *elif strChoice == 'r':*  *# Read the file line by line into in-memory list.*  *objFile = open(strFileName, 'r')*  *for row in objFile:*  *lstRow = row.strip().split(',')*  *dicRow = {'artist': lstRow[0], 'title': lstRow[1]}*  *lstTbl.append(dicRow)*  *print(dicRow)*  *objFile.close()*  *pass*  *elif strChoice == 'x':*  *#Delete entry as indicated by user.*  *garb\_Art = input('Artist to delete:')*  *for i in range(len(lstTbl)):*  *if lstTbl[i]['artist'] == garb\_Art:*  *del lstTbl[i]*  *break*  *pass*  *elif strChoice == 'd':*  *# Display the data to the user.*  *print('Here are all the rows:\n',lstTbl)*  *pass*  *else:*  *print('Please choose either a, w, r or exit!')* |