Kent Bloodsworth

February 27, 2022

Foundation of Programming (Python)

Assignment05

CDInventory

# Introduction

We continued to work with lists in this assignment and were introduced to dictionaries. We also covered the topics of separation of concerns programming pattern, error handling, functions, and how to use the GitHub repository. After going over the material for this module, the assignment had us modify the CD inventory script to add some functionality and convert the lists to dictionaries. Also, part of completing the work was uploading the documents to a GitHub repository.

# Modifying Script

We were given a starter script with this assignment, which included code from the previous module. It was interesting to see the differences and similarities with this script when compared to my script for module 4 with the same basic functionality. I first replaced the inner data structure with dictionaries when modifying this script. This wasn't too hard and didn't take too much time. Next, I added the functionality of deleting, which took a lot more time as I tried a couple of different methods but decided to use the del method to complete this TODO. The last modification I had to make was loading existing data into memory. I struggled a little more with adding this functionality, but there is an example in the module PDF that I used to get through this. Overall, I think it turned out well, and once I had it done, I uploaded it to GitHub.

# Screenshots Of Working Script

Figure 1 is a screenshot of the script working in the Terminal console. To get the script to save the file and load data from a file, I had to put in the specific path in which I had the file. Figures 2 and 3 are screenshots of the script working in Spyder; using Spyder, I didn't have to give a specific path to the file as it created the file in the same directory that the script was saved in.

# Summary

After going through the modules, reading, and watching the videos, I completed modifying the code. This included adding functionality to load existing data, deleting an entry if the user chooses to, and replacing the inner data structure from the list to dictionaries. Once this was complete, I made an account on GitHub and uploaded the documentation to a repository for this assignment.

# Appendix

Repository link:

<https://github.com/KentBloods/Assignment05>

|  |  |
| --- | --- |
| 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  69  70  71  72  73  74  75  76  77 | *#------------------------------------------#*  *# Title: CDInventory.py*  *# Desc: Starter Script for Assignment 05*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, Created File*  *# KBloodsworth, 2022-Feb-27, Finished TODO's*  *#------------------------------------------#*  *# Declare variabls*  strChoice = '' *# User input*  lstTbl = [] *# list of lists to hold data*  dicRow = {} *# list 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':  *# Add the functionality of loading existing data*  lstTbl.clear()  objFile = open('CDInventory.txt', 'r')  **for** row **in** objFile:  lstRow = row.strip().split(',')  dicRow = {'ID': int(lstRow[0]), 'Title':lstRow[1], 'Name':lstRow[2]}  lstTbl.append(dicRow)  objFile.close()  *# New results*  print('New items in list:')  **for** row **in** lstTbl:  print(\*row.values(), sep = ', ')  **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*  strID = input('Enter an ID: ')  strTitle = input('Enter the CD**\'**s Title: ')  strArtist = input('Enter the Artist**\'**s Name: ')  intID = int(strID)  dicRow = {'ID':strID, 'Title':strTitle, 'Name':strArtist}  lstTbl.append(dicRow)  **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.values(), sep = ', ')  **elif** strChoice == 'd':  *# Add functionality of deleting an entry*  print('Select from the list which entry to delete')  **for** row **in** lstTbl:  print(\*row.values(), sep = ', ')  print('**\n**Note that the index starts at 0')  entry = int(input('Enter the index for the entry you want to delete: '))  **del** lstTbl[entry]  print('**\n**New table is now:')  **for** row **in** lstTbl:  print(\*row.values(), sep = ', ')  **elif** strChoice == 's':  *# 4. Save the data to a text file CDInventory.txt if the user chooses so*  objFile = open('CDInventory.txt', 'a')  **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!') |