Name: Fabio Sacca

Date: Nov-13, 2022

Course: Foundations of Programming (Python)

Title: Assignment 05

# Assignment 05

## Introduction:

In this document, I will describe the steps I took to update the ‘CD Inventory’ program from a starter script, following these requirements:

1. Add functionality of loading existing data
2. Add functionality of deleting an entry
3. Replace inner data structures with ‘dictionaries’ and ensure the script uses a list of dictionaries as 2D table.

## Topic 1: Creating a Python script

I opened the 'CDInventory\_Starter.py' starter script, and worked through the various sections of the program focusing on one requirement at a time.

### 1.1. Add functionality of loading existing data

To meet this requirement, I added a functionality that will open the strFileName (CDInventory.txt), and extract each row of the file as a list, separated by comma. The functionality will then convert the extracted list to a dictionary with assigned keys ['id':, 'Title':, 'Artist':] for a corresponding value in list. Finally, the functionality will append each dictionary to the table lstTbl, and close the file.

**if** strChoice == 'l':

*# TODO Add the functionality of loading existing data*

*# @fabiosacca: Added functionality and Structured Error Handling if there is no file to load*

*# 2. Load existing data*

**try**:

lstTbl = []

objFile = open(strFileName, 'r')

**for** row **in** objFile:

lstRow = row.strip().split(',')

lstRow[0] = int(lstRow[0]) *# converted ID to integer*

dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}

lstTbl.append(dicRow)

objFile.close()

**except**:

print('**\n**An error occurred. No Inventory file found. **\n**')

### 1.2. Add functionality of deleting an entry

To meet this requirement, I declared an input variable (delRow) assigned to a value entered by the user during the program execution. Then, I created a functionality that uses a FOR loop to unpack the rows and values within the current data table, and added an IF statement that checks if any of the values on each row is equal to the value entered by the user. I had to ensure that the ID values in each row of the 2d table ('id': lstRow[0]) and the user entry (delRow) are integers. The program will remove the row within the table that returns a ‘True’ result to the IF statement: value == delRow.

**elif** strChoice == 'd':

*# TODO Add functionality of deleting an entry*

*# @fabiosacca: added functionality and error handling rule if ID entered is not in table*

*# 5. Delete an entry from current data*

delRow = int(input('Enter the ID you want to delete: '))

**for** row **in** lstTbl:

**for** value **in** row.values():

**if** value == delRow:

lstTbl.remove(row)

Limitations:

* This program allows more than one dictionary to have the same value associated to the key ‘id’. When this occurrs, the functionality will delete only the first row that returns a ‘True’ result to the IF statement.
* This functionality does not handle errors that may result from an invalid entry (user enters a string instead of an integer), or from a valid entry that does not exist in the current data

### Ensure the script uses a list of dictionaries as 2D table

To meet this requirement, I had to declare a new variable dicRow = {} at the start of the script and convert the rows of the 2D table into dictionaries in two sections: 2. Loading existing data, and 3. Add data to the table.

**if** strChoice == 'l':

*# TODO Add the functionality of loading existing data*

*# @fabiosacca: Added functionality and Structured Error Handling if there is no file to load*

*# 2. Load existing data*

**try**:

lstTbl = []

objFile = open(strFileName, 'r')

**for** row **in** objFile:

lstRow = row.strip().split(',')

lstRow[0] = int(lstRow[0]) *# converted ID to integer*

dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}

lstTbl.append(dicRow)

objFile.close()

**except**:

print('**\n**An error occurred. No Inventory file found. **\n**')

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

*# 3. 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]

dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}

lstTbl.append(dicRow)

*# @fabiosacca: corrected starter script to ensure row is a dictionary*

### 1.3. Structured Error Handling

While not a requirement for this assignment, I tried to follow the error handling model principle and added code where user entries may result into a program error, in particular:

* If the user attempts to load a 'CDInventory.txt' file (by selecting 'l'), when the file has not been created yet.

**if** strChoice == 'l':

*# 2. Load existing data*

**try**:

lstTbl = []

objFile = open(strFileName, 'r')

**for** row **in** objFile:

lstRow = row.strip().split(',')

lstRow[0] = int(lstRow[0]) *# converted ID to integer*

dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}

lstTbl.append(dicRow)

objFile.close()

**except**:

print('**\n**An error occurred. No Inventory file found. **\n**')

This principle could offer opportunities to improve the script in any situations where a user entry may result in an error, for example: if the user inputs an invalid entry (i.e., a string value where an integer is expected).

## Topic 2: Running my script

I saved the script as 'CDInventory.py' in a folder called 'Assignment05' within the course folder '\_FDProgramming', and ran the script in Spyder. The image below shows that the script works in my computer.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1 - Screen capture of Spyder returning CDInventory.py script

A screenshot of a computer

Description automatically generated with medium confidence

Figure 2 - Screen capture of Spyder returning CDInventory.py script (continued)

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3 - Screen capture of Spyder returning CDInventory.py script (continued)

I repeated this step in a terminal window. The image below shows that the script works in my computer.

A picture containing text

Description automatically generated

Figure 4 - Screen capture of OS Terminal returning CDInventory.py script

## Graphical user interface, text, application Description automatically generated

Figure 5 - Screen capture of OS Terminal returning CDInventory.py script

## Topic 3: Verifying that my script worked

I located the file ‘CDInventory.txt’ and opened it in a text editor. The image below shows that the data I entered has been written to the file in the correct format.

A picture containing text

Description automatically generated

Figure 6 - Screen capture of CDInventory.txt displaying data entered.

## Topic 4: Submitting my work on GitHub

Both the CDInventory.py and the present knowledge document are posted on the following public repository

## <https://github.com/fabiosacca/Assignment_05>

## Summary

In this assignment, I covered the steps needed to:

* Update the “CD Inventory” program from a starter script replacing the inner data structures with ‘dictionaries’, and adding functionalities of loading existing data, and deleting and entry
* Run the script in Spyder and a terminal window.
* Verify that the script in fact wrote in CDInventory.txt the data I entered in the correct format.
* Submit my work on a public GitHub repository.

# Appendix A

## Code of CDInventory.py script presented with [**Syntax Highlighter**](https://saravjishut.org/syntax)**[[1]](#footnote-1)**

|  |  |
| --- | --- |
| 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  78  79  80  81  82  83  84  85  86  87 | *#------------------------------------------#*  *# Title: CDInventory.py*  *# Desc: Script for Assignment 05*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, Created File*  *# fabiosacca, 2022-Nov-13, Modified script to use dictionaries*  *#------------------------------------------#*  *# Declare variables*  strChoice = '' *# User input*  lstTbl = [] *# list of dicts to hold data*  lstRow = [] *# list of data row*  dicRow = {} *# dictionary*  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:*  *# @fabiosacca: Capitalized each row for consistency*  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')  print()  strChoice = input('l, a, i, d, s or x: ').lower() *# convert choice to lower case at time of input*  **if** strChoice == 'x':  *# 1. Exit the program if the user chooses so*  **break**    **if** strChoice == 'l':  *# TODO Add the functionality of loading existing data*  *# @fabiosacca: Added functionality and Structured Error Handling if there is no file to load*  *# 2. Load existing data*  **try**:  lstTbl = []  objFile = open(strFileName, 'r')  **for** row **in** objFile:  lstRow = row.strip().split(',')  lstRow[0] = int(lstRow[0]) *# converted ID to integer*  dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}  lstTbl.append(dicRow)  objFile.close()  **except**:  print('**\n**An error occurred. No Inventory file found. **\n**')  **elif** strChoice == 'a': *# no elif necessary, as this code is only reached if strChoice is not 'exit'*  *# 3. 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]  dicRow = {'id': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}  lstTbl.append(dicRow)  *# @fabiosacca: corrected starter script to ensure row is a dictionary*    **elif** strChoice == 'i':  *# 4. 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 = ', ')  *# @fabiosacca: corrected to print only the values from dictionary*    **elif** strChoice == 'd':  *# TODO Add functionality of deleting an entry*  *# @fabiosacca: added functionality. Need to add error handling logic (invalid entries, entry not in table)*  *# 5. Delete an entry from current data*  delRow = int(input('Enter the ID you want to delete: '))  **for** row **in** lstTbl:  **for** value **in** row.values():  **if** value == delRow:  lstTbl.remove(row)    **elif** strChoice == 's':  *# 6. Save the data to a text file CDInventory.txt if the user chooses so*  objFile = open(strFileName, 'w')  **for** row **in** lstTbl:  strRow = ''  **for** value **in** row.values():  strRow += str(value) + ','  strRow = strRow[:-1] + '**\n**'  objFile.write(strRow)  objFile.close()    **else**:  print('Please choose either l, a, i, d, s or x!**\n**') *# @fabiosacca: Added \n to improve formatting* |

1. Last viewed: Nov-13, 2022 (7:30pm PT) [↑](#footnote-ref-1)