Miroslava Meza

02/27/2022

IT FDN 110B Wi 22: Foundations of Programming Python

Assignment 05, Module 05

# Lists, Dictionaries, script templates and GitHub

## Introduction

This assignment had the goal of building upon module 04 use of lists, loops and menus. During this activity, the pervious 2D data structure is modified to use dictionaries as the inner data type.

Based on the last week’s assignment started script, I modified the script and replaced the inner data structure by dictionaries. This version allowed to load existing data, to erase an entry, and to add new entries while saving the information into a .txt file.

## Dictionaries as inner data structure.

The goal of this step was to use the original “starter script” and to create dictionaries as part of the data structure.

1. Declaring variables change, in the starter script, lstRow =[] was declared as an empty list-variable. That was substituted by another variable of dictionary type, dicRwo={}. See row 14, figure 1.

Figure . Declaring variables. Including a directory variable

Text

Description automatically generated

1. Creating tables with dictionary data structure instead of having 2 dimensional lists to being appended. This way the properties of the dictionary are passed to the new table (lstTbl). See rows 38 and 39 in Figure 2.
2. Future data entries are created under a directory structure and appended into a variable called “lstTbl”. That variable is the one used to write the information into an external .txt file.
3. For all the modifications on the script, see the Appendix "Full Script” in this report.

## Adding the functionality of loading existing data.

The first step consists in reading the information stored in an external file, or creating the file if it does not exist yet. To do it, I used the function ***open(object, ‘r’)*** . I used the operator ‘r’ and not ‘w’ because the goal is not to write new information into the file, but to read and import into a variable (lstTbl) the values currently stored in the txt file.

When reading the information, I broke it by commas and eliminated spaces ( Figure 2. Loading data from a txt file. row 37). The pervious format, allowed to structure the information into a directory where the fist position [0] is the id, followed by the artist’s name [1] and finally by the cd title [2] (Figure 2. Loading data from a txt file, row 38). Finally, that row by row information was stored into a memory variable (lstTbl) see Figure 2. Loading data from a txt file, row 39. I used the data structure of a dictionary for each row.

At the end of this step, the values from the txt file are stored in memory, using a directory structure within a variable.

Figure . Loading data from a txt file.

Text

Description automatically generated

## Adding the functionality of deleting an entry

This was the most challenging step of the homework. I originally overcomplicated my solution by including a loop intended to erase entries based on their location. I solved the challenges by simplifying my approach as follows (Figure 4. Deleting an entry):

1. Printing the current inventory, allowing the user to count the rows. When I asked the user for input, I asked which row they want to be deleted/erased.
2. Ensuring that the row information is changed to the actual value for the list by turning the string entry into a integer and then subtracting one, to match the start point =[0].
3. I tried to use the function object.pop() but it gave me an error that I could not fix. As a solution, I used another function, ***.remove[location].*** For the location to remove, I used the row-1 that the user provided as the location.
4. At the end of the row 93, the memory variable was updated eliminating the erased entry. This information was not yet stored in the .txt file.
5. The final step was to “write” the values of the inventory into the .txt external file. Ensuring that the values stored in the memory and the values in the .txt file are the same at the end of this step.

Figure . Deleting an entry.

Text

Description automatically generated

## Running the Script in Spider

The script was tested following the sequence below:

If there was no previous file 'CDInventory.txt'

1. Started using the option [a] from the menu.
2. Option [w] to save the information to the .txt external file
3. Option [d] to display the information
4. Option [e] to erase an entry.
5. Option [d] to display the current inventory.
6. Option [exit] to close the script.

Second try, if there was a previous file named 'CDInventory.txt’ with stored entries.

1. Started using the option [l] from the menu.
2. Option [a] adding one more inventory entry.
3. Option [w] to save the information to the .txt external file.
4. Option [d] to display the information.
5. Option [e] to erase an entry.
6. Option [d] to display the current inventory.
7. Option [exit] to close the script.

Figure . Loading data from a txt file and adding a new entry.

Text

Description automatically generated

|  |  |
| --- | --- |
| *Figure 5 Loading data from a file. Spider IDLE* Text  Description automatically generated | Figure 6. Displaying data and quitting the script  Text  Description automatically generated |

Figure . Deleting an entry and displaying the new list. Using directories.

Text

Description automatically generated

## Running the Script in Terminal

Figure . loading, adding, writing, and displaying data using the Script.

Text

Description automatically generated

Figure . Erasing an entry, displaying and exiting the script using the Terminal.

Text

Description automatically generated

## Conclusions

This assignment had the goal of building upon module 04 use of lists, loops and menus. During this activity, the pervious 2D data structure was modified to use dictionaries as the inner data type. Based on the last week’s assignment started script, I modified the script and replaced the inner data structure with dictionaries. This script allowed the following functionality:

1. Started using the option [l] from the menu.
2. Option [a] adding one more inventory entry.
3. Option [w] to save the information to the .txt external file.
4. Option [d] to display the information.
5. Option [e] to erase an entry.
6. Option [d] to display the current inventory.
7. Option [exit] to close the script.

## Appendix Full Script

|  |  |
| --- | --- |
| 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  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118 | *#--------------------------------------------------------------------#*  *#Title: Mod05\CDInventory.py*  *#Description: Lab05-b example solution*  *#Change Log: (Who, When, What)*  *#DBisinger, 2030-Jan-01, created file and example solution*  *#Miroslava Meza, 02/26/2022, reproduced scritp and followed the instructions.*  *#Miroslava Meza, 02/27/2022. notated where and how to use directories.*  *#--------------------------------------------------------------------#*  *#Declare variables*  strChoice ='' *#user input*  lstTbl =[] *#list of lists to hold the data*  *#TODO 1: creating a data-structure based in dictionaries*  dicRow={} *#lstRow = [] we substitute with a directory in this example the list of data row*  strFileName = 'CDInventory.txt' *#data storage file*  objFile = **None** *#file object*  *#Get user input*  print('The Magic CD Inventory **\n**')  *# Menu with the options available to the user.*  **while** **True**:  print('**\n**[l] Load data from file**\n**[a] Add data to list**\n**[w] Write data to file')  print('[e] Erase an inventory entry**\n**[d] Display data**\n**[exit] Quit')  strChoice = input('l, a, w, e, d, or exit. ').lower() *#convert choice to lower case*  print() *#prints a blank row*    **if** strChoice =='exit':  **break**  *#-- The first option available in the menu is to load existing data from a file.*  *# TODO 2: Create teh option of loading existing data.*  **if** strChoice =='l': *# not using elif in this one, bacause is only available if not "exit""*  *#open the txt file to print.*  objFile=open(strFileName, 'r') *# we are only reading information not writing.*  **for** row **in** objFile:  lstRow= row.strip().split(',')  dicRow = {'ID': lstRow[0],'Artist': lstRow[1], 'Title': lstRow[2]} *#TODO1*  lstTbl.append(dicRow)  objFile.close()  **for** row **in** lstTbl:  print(\*row.values(), sep =',')  *#I decided not to print the values while the .txt file was open to avoid risk*  *#of corrupting the file.*  print('If you want to see the current inventory, select **\'**d**\'** in the next menu ')  **pass**  **elif** strChoice == 'a':  *#Add data to list in memory*  ID=input('Enter the CD Id: ')  strID=str(ID)  strArtist = input('Enter the CD**\'**s artist: ')  strTitle = input('Enter the CD**\'**s Title: ')  *#Creating the keys for a dictionary.*  dicRow= {'ID': strID, 'Artist': strArtist, 'Title': strTitle}*#TODO1*  *#Each new dictionary entry will be append to lstTbl*  lstTbl.append(dicRow)  *#let the user know that their changes are saved only in temporary memory and reminds them to*  *# use the option Write to save later.*  print('your current changes are saved in temporary memory,**\n**save your changes into the external txt file by choosing [w] from the main menu')  **pass** *#added this item from the original starter file.*  **elif** strChoice=='w':  *#write from list to data file*  objFile=open(strFileName, 'w') *# I did not use the 'a' function because*  *#the user had the opportunity to load the vaues of*  *#the existing values. If they chose not to, this allows*  *#to start all over from zero. If they want to append, they can just use l*  **for** row **in** lstTbl:  strRow=''  **for** item **in** row.values():  strRow += str(item) + ','  strRow = strRow[:-1]+'**\n**'  objFile.write(strRow)  objFile.close()  *#letting the user know that the all their new entries have been added to*  *# the external txt file.*  print('All your recent entries and/or deletions have been saved to the file CDInventory.txt')  **pass**  **elif** strChoice =='e':  *# TODO 3:*    *# 1--- Display updated data to identify the row*  print('Identify the row you want to erase, count top from bottom and type the row number **\n**')  print('ID, Artist, Title')  **for** row **in** lstTbl:  print(\*row.values(), sep = ',')  *# Request the row to erase, The data is a list form, thus whatever the user*  *# provides shoud be ((row\_input)-1)*  eraselist2=int(input('Write the number of row you want removed from your inventory: '))  i=eraselist2-1 *# the list starts in [0].*  lstTbl.remove(lstTbl[i])  *#----Write into txt file*  objFile=open(strFileName, 'w') *# I did not use the 'a' function because*  *#the user had the opportunity to load the vaues of*  *#the existing values. If they chose not to, this allows*  *#to start all over from zero. If they want to append, they can just use l*  **for** row **in** lstTbl:  strRow=''  **for** item **in** row.values():  strRow += str(item) + ','  strRow = strRow[:-1]+'**\n**'  objFile.write(strRow)  objFile.close()    **pass**  **elif** strChoice =='d':  *#Display data*  print('ID, Artist, Title')  **for** row **in** lstTbl:  print(\*row.values(), sep = ',')  **pass**  **else**:  print('Please choose either l, a, w, e, d, or exit. ')  **pass** |