< Jayr Padilla >

< 11/13/22 >

< Fundamentals of Programming: Python >

< Assignment 05 >

Using dictionaries for CD directories

# Introduction

The purpose of this assignment was to modify last week’s assignment while also adding functions for the user to choose from: deleting an entry and loading from exiting data (text file). This was to be completed with the information learned in module 5 such as dictionaries and the proper syntax and pulling data from existing text files.

# Working with Dictionaries

**Continuing from last week’s module, we learned about another sequence type: dictionaries. Working with dictionaries is like working with lists although there are some differences. One major difference is using “keys” instead of index’s when calling values in a dictionary. Each value is assigned to a key which can be used to call the value and referenced in other parts of a code. Specifically looking at figure 1, this is the menu choice of adding data to the CD inventory. Using a while loop, it was needed to first assign the inputs from the user to variables. After receiving inputs, it was then needed to construct the dictionary using the keys previously mentioned and assigning the values. From there I appended the dictionary row to the table. I did struggle understanding when to assign directly into the dictionary or when to assign to variables first. After doing both and experimenting I was able to understand the difference.**

Graphical user interface, text

Description automatically generated

Listing - Adding data to table using dictionaries

# Loading existing data from file

**Learning how to load existing data from file was something that required close attention to what is being completed. When learning this in module 5, I did take the most time dissecting and experimenting with this code and examples**. With working with dictionaries there is one extra line of code that is needed which is convenient. Learning how to do this first with lists allowed me to easily understand loading data from files using dictionaries. Looking at Listing 2, line 68 is the extra line needed when pulling data and entering it in dictionaries. After pulling the data into a list, all that is needed is to use indexing to pull the specific data from the lists and assigning them to their respective keys. I also do like the practice after loading the data to the dictionary and then table to print it out for the user. It not only helps see the data properly being pulled into the memory but also for us, the programmers, to see that there are no errors or data missing. Looking Figure 1, I first added data to the inventory and then saved it to the file. The data being entered and saved to the text file can be seen in Figure 2. After that I pulled the existing data from the file and printed it out to the user. This script and steps were executed in Spyder.

Text

Description automatically generated

Listing 2 -Loading existing data from file

Text

Description automatically generated

Figure 1 - Loading existing from file (Spyder)

Graphical user interface, text, application

Description automatically generated

Figure - CDInventory text file showing entries

# Deleting entries

**This part of my code was the most difficult to go through and required the use of online resources. I knew to complete this task, a for loop was needed to iterate through the table to find the entry the user would like to delete. I also knew that the ID would be what I would use to identify the entry to be deleted. To accomplish this task I used class methods I had not previous used before which are range( ) and len( ). Starting the for loop is where is used these two methods which can be seen in Listing 3, line 81. I first needed to get the number of items in the list table and then called range so the for loop knew how many times to iterate though. Then in line 82 I used two indexes for list table, first the row and then the ID of the dictionary row. This part took the longest for me to complete but I was happy to have worked through this problem and having it execute properly.**

Text

Description automatically generated

Listing 3 - Deleting existing entry

Figure 3 shows me adding two entries into a table then displaying the table to the user so they may know they were properly entered. I then executed the part of the script to delete a desired entry. First prompting the user to enter the ID of the entry they would like to delete. After the string “Entry has been deleted!” is printed to tell the user it has been executed. One final step is taken to make sure the table no longer has that entry, I displayed the current data to the user. These steps were executed and shown in Figure 2 in the terminal window.

Text

Description automatically generated

Figure - Adding and then deleting an entry (Terminal)

# Summary

In this assignment I learned various topics to help me build this script. The main part of this module had a focus of learning about dictionaries and how most functions are like that of lists, as well as the structure of them. Having the knowledge of how to use lists from module 4 allowed me to easily learn apply and interchanging dictionaries with lists. Looking at figure 1 and 2, you can see parts of the script being executed in Spyder and in Terminal, respectively

Using [Saravijust Syntax](https://saravjishut.org/syntax) (external reference)[[1]](#footnote-1) web page

# Appendix

## Listing CDInventory.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  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90 | *#------------------------------------------#*  *# Title: CDInventory.py*  *# Desc: Creating a CD Inventory using dictionaries*  *# Change Log: (Who, When, What)*  *# DBiesinger, 2030-Jan-01, Created File*  *# JPadilla, 2022-Nov-13, Replaced inner data structure by dictionaries*  *# JPadilla, 2022-Nov-13, Reorganizing/ Reformatting the structure of the script allowing it to be easily read*  *#------------------------------------------#*  *# Declare variabls*  strChoice = '' *# User input*  lstTbl = [] *# list of lists to hold data*  *# TODO replace list of lists with list of dicts*  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()  *# Exit the program if the user chooses so*  **if** strChoice == 'x':  **break**  *# Add data to the table (2d-list) each time the user wants to add data*  **if** strChoice == 'a':  *# 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: ')  print()  intID = int(strID)  dicRow = {'ID':intID, 'Title': strTitle, 'Artist': strArtist}  lstTbl.append(dicRow)    *# Display the current data to the user each time the user wants to display the data*  **elif** strChoice == 'i':  print('ID, Title, Artist')  **for** row **in** lstTbl:  print(\*row.values(), sep = ', ')  print()    *# Save the data to a text file CDInventory.txt if the user chooses so*  **elif** strChoice == 's':  objFile = open(strFileName, 'w')  **for** row **in** lstTbl:  objFile.write(str(row['ID']) + ',' + row['Title'] + ',' + row['Artist'] + '**\n**')  objFile.close()  print('Entry saved to file! **\n**')    *# Load exisitng data from text file*  **elif** strChoice == 'l':  lstTbl.clear()  objFile = open(strFileName, 'r')  **for** row **in** objFile:  lstRow = row.strip().split(',')  dicRow = {'ID': lstRow[0], 'Title':lstRow[1], 'Artist': lstRow[2]}  lstTbl.append(dicRow)  objFile.close()  print('Items in Table:')  print('ID, Title, Artist')  **for** row **in** lstTbl:  print(str(row['ID']) + ',' + row['Title'] + ',' + row['Artist'] + '**\n**')  **pass**  *# Give option to user to delete existing entry*  **elif** strChoice == 'd':  delID = int(input('Enter ID you would like to delete: '))  **for** row **in** range(len(lstTbl)):  **if** lstTbl[row]['ID'] == delID:  **del** lstTbl[row]  **break**  print('Entry has been deleted! **\n**')  **pass**    **else**:  print('Please choose either l, a, i, d, s or x!') |

1. Retrieved 2022-Nov-13 [↑](#footnote-ref-1)