Danielle Rodriguez

28 February 2022

IT FDN 110 B

Assignment 05

Using Lists and Dictionaries

Introduction

This module continues to explore lists and introduces dictionaries. Lists are a convenient way to hold collections of data, while dictionaries are mapping types. Additionally, this module covered how to open files, read files and write to files using both lists and dictionaries. This process highlighted the importance of using separation of concerns and templates to organize scripts in an orderly manner. By utilizing these tools, I improved upon the CD Inventory menu created last week.

Lists

Lists hold collections of data and provide the user with several built-in functions. Figure 1 outlines some of the functions and commands commonly used with lists. The script first defines the variables, including the rows and file name. Then, the file name is called. To work with data, we must load the data from a file. Once the file is open, the script uses split() to separate items in the row with a coma. The second half of the script opens the same file but adds the strip() function to remove the \n from the string, cleaning up the data. Text

Description automatically generated

**Figure 1: Using lists**

Dictionaries

Unlike lists, dictionaries are mapping types that store sequences by storing key:value pairs. When using key:value pairs, the user must define a key to the values being stored. Additionally, dictionaries use the {} operator, as seen in lines 1 and 2 of Figure 2. Dictionaries can be inserted into lists to create 2D tables, which requires a few key steps. As seen in the script below, each key needs to be defined in a specific manner but still allows the user to input their data. This script allows the user to insert and update their email and name to a 2D table. Although not shown in Figure 2, the programmer can also read a row of data from text files by calling each key:value pair.

Text

Description automatically generated

**Figure 2: Using Dictionaries**

Improving the CD Inventory

Using the tools learned in the model, I improved upon the CD Inventory menu. I used the CD Inventory starter file as a starting place and amended the code to fit the assignment's requirements. Line 12 through 16 define the variables and set up the table and file. The following lines allow users to select their choice in the menu and perform the desired action. This code used to execute the menu options resembles closely to last week's assignment but adds a few dictionary specific lines. For example, lines 34 and 57 define each variable's key:value pairs.

I also added two functionalities to the script: viewing all data and deleting the data. To view all of the data saved to the file, I used the .read() command, as seen in line 71. This called the data from the file even if it was not recently added. Writing a script to delete entries was a bit more difficult. Lines 73 through 87 delete the most recent entry from the file. To do this, I created a "temp" table to store the data and delete the most recent entry without overwriting the data stored in the file. While an entry is deleted, this does not allow the user to choose which entry to delete. While this code does not allow the user to choose a specific entry to delete, the script still allows adding, displaying, and viewing data more cleanly.

Summary

This module built upon the tools learned in last week’s assignment and expanded on 2D tables. Using lists and dictionaries, I was able to add more functionality to the CD Inventory Menu and improve the formatting by using separation of concerns.

Appendix

|  |  |
| --- | --- |
| 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 | *#------------------------------------------#*  *# Title: CDInventory.py*  *# Desc: CD Inventory*  *# Change Log: (Who, When, What)*  *# DRodriguez, 2022-02-28, Created File*  *#------------------------------------------#*  *# -- DATA -- #*  *# Declare variables*  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*  *# -- PROCESSING -- #*  *# 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('[d] display data**\n**[v] view existing data**\n**[x] to delete**\n**[exit] to quit')  strChoice = input('a, w, r, d, v, x, or exit: ').lower() *# convert choice to lower case at time of input*  print('**\n\n**')  **if** strChoice == 'exit':  **break**  **if** strChoice == 'a': *# no elif necessary, as this code is only reached if strChoice is not 'exit'*  *# Add data to list in memory*  strCD\_Title = input('Please enter CD Title: ')  strCD\_Artist = input('Please enter Artist Name: ')  dicRow = {'CD Title': strCD\_Title, 'Artist': strCD\_Artist}  lstTbl.append(dicRow)  **elif** strChoice == 'w':  **if** len(lstTbl) != 0:  *# List to File*  objF = open(strFileName, 'a')  **for** row **in** lstTbl:  strRow = ''  **for** item **in** row.values():  strRow += str(item) + ','  strRow = strRow[:-1] + '**\n**'  objF.write(strRow)  objF.close()  lstTbl.clear()  print('Saved to file.**\n**')  **else**:  print('No data available.**\n**')  **elif** strChoice == 'r':  *# File to print*  lstTbl.clear()  objF = open(strFileName, 'r')  **for** row **in** objF:  lstRow = row.strip().split(',')  dicRow = {'CD Title': lstRow[0], 'Artist': lstRow[1]}  lstTbl.append(dicRow)  objF.close()  print('Artist, Title')  **for** row **in** lstTbl:  print(row)  **elif** strChoice == 'd':  *# Display data*  print('Artist, Title')  **for** row **in** lstTbl:  print(\*row.values(), sep = ', ')  **elif** strChoice == 'v':  *# Load existing data*  objF = open(strFileName, 'r')  print(objF.read())  input('Press enter to continue.')  **elif** strChoice == 'x':  *# Delete an entry*  objF = open(strFileName, 'r')  temp = [] # create a temporary table to store the data  **for** row **in** objF:  temp.append(row)  objF.close()  **if** len(temp) != 0:  objF = open(strFileName, 'w')  temp = temp[:-1]  **for** row **in** temp:  objF.write(row)  **else**:  print('No data to delete.**\n**')  objF.close()  **else**:  print('Please choose either a, w, r, d, v, or exit!') |

Appendix : CD Inventory Script

Text

Description automatically generatedText

Description automatically generated

Appendix : CD Inventory in terminal

Text

Description automatically generatedA screenshot of a computer screen

Description automatically generated with medium confidence

Appendix : CD Inventory in Spyder