Vernon Jackson

Aug 8, 2021

IT FDN 110

Assignment 05

# Program to Store CD Inventory Data

# Introduction

In this document I will be explaining how I added code to the CD Inventory Starter program for assignment 05, that modifies the script as required to replace the inner data structure by dictionaries, Add the functionality of loading existing data and add functionality of deleting an entry.Areas of focus for this document include 1) briefly covering the fundamentals that I learned in this module 2) the steps I took to organize and create the script and 3) a summary of my experience in creating this program.

# Coding Fundamentals

Within this assignment, I learned several new fundamentals of coding. This includes Dictionaries, Index and Keys, and GitHub. I will briefly summarize my own definitions of each used in the assignment in order to substantiate my learning.

**Index and Keys:** Indexes, within Python refers to a position within an ordered list, while keys are unique within a dictionary, that indicate specific sets of values.

**Dictionary:** A dictionary in Python is a container which stores specific mappings of unique keys to values.

**GitHub:** Github is an online platform where you can host your code, and share them with the your colleagues.

# Creation of the Program

The *cd* variable is then used in the *add\_inventory(cd)* function, the *display\_list(cd)* function, and the *save\_cd(cd)* or *save to file* function. The only function that changes the variables value, is the add\_inventory function, which allows you to add new inputs into the list, which those changes can be viewed outside the function in the main program. The other two are simply calling upon the values to either display them to the user (*display\_list)* or write the values (*save\_cd)* to a text file.

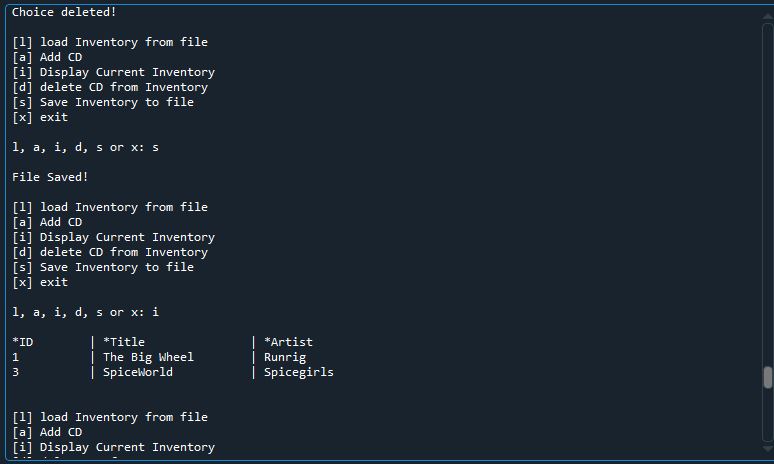


Figure 1 - Showing the program written in Spyder

To test this script, I ran this as a Python script through Spyder IDE and Anaconda Prompt. Executing the script through both, showed that it had the intended functionality which includes a menu structure and allows the user to enter CD data, view the current inventory, save data to a CDInventory.txt data file and exit the program. See appendix Listing CDInventory.pyfor highlighted code.

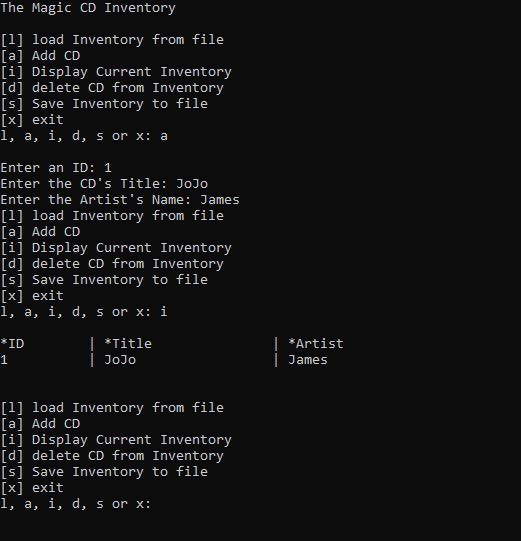


Figure 3 - Shows script being executed through IDLE

# Summary

In this assignment we learned about loops, tuples, lists, and string formatting. This was the most difficult assignment I have done within the course, as it took me most of Friday and Sunday to complete the code. The top 3 issues I ran into were 1) formatting of the strings to print data correctly into a user-friendly table 2) finding the right values to make the while loop for the menu to function correctly and 3) figuring out how to exit the script in the exit\_thanks function as I kept getting errors using solutions from prior assignments. The formatting of the strings was confusing, as there was not much documentation on this in the modules. We were directed to the Python Library, however, I was uncertain of which one I should or would be most appropriate to use. I ended up using the f.strings to display the column headers and the str.format for formatting the data within the lists. I tried to be consistent, but was unable to find a solution that worked for both. For the loop, I wasn’t sure if I should just add a single value of 1 to the initial conditions, but was forced to use 1,2,3 for the selections within the loop. I initially thought I only needed the initial value. Lastly, importing sys was something that took me a while to figure out, but eventually I found the solution through online help, as there was no real recommendations for which is most appropriate. Given that we were saving things to memory, I chose the sys.exit to ensure both the program and system were closed. Also, when writing the initial pseudocode, I end up determining that functions and loops for the menus were the best option (as I have most times), but had a difficult time actually putting them into working code.

Additionally, as far as my learnings within this course, I am finding it very difficult to understand if I’m too far ahead with the code I’m using or too far behind with some of the basics. The modules cover some material briefly, which I believe is needed, so I often spend significant time learning these through additional resources and other programmers in my network. However, I find out that these are covered during the next module. I have high expectations for my code, so I’m rather prone to trying to break it and solving vs just applying the basics. Being that I have basically no experience in coding, learning some of the concepts to solve for these bottlenecks are a significant challenge for me.

# 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 | #------------------------------------------#  # Title: CDInventory.py  # Desc: Script CDINventory to store CD Inventory data  # Change Log: (Who, When, What)  # VJackson, 2021-Aug-01, Created File  #------------------------------------------#    # This is a CD Inventory program    # General formatting notes: ""=display text, ''=input strings    # 0.creating the current list as an object that can be used with other functions  **def** **cd\_list**():  list\_data = [[**1**,'The Big Wheel','Runrig'], [**2**,'Bad','Michael Jackson']]  **return** list\_data  # 1. Display menu allowing the user to choose option  **def** **menu\_choice**():  **print**("=================================")  **print**("**\t**CD Inventory Program Menu")  **print**("=================================")  **print**("\*You can take the following actions:\*")  **print**("1. Add CD Data")  **print**("2. Display CD List")  **print**("3. Save Inventory to Text File: CDInventory")  **print**("4. Exit CD Inventory")  selection = int(input('Please enter your choice: '))  **return** selection    # 2. Add data to the table (2d-list) each time the user wants to add data  **def** **add\_inventory**(cd):  inventory = []  **for** item **in** range(len(cd[**0**])):  **if** item == **0**:  inventory.append(int(input('Enter ID#: ')))  **if** item == **1**:  inventory.append(str(input('Enter CD Title: ')))  **if** item == **2**:  inventory.append(str(input('Enter Artist Name: ')))  cd.append(inventory)  **print** (inventory)  **return** cd    # 3. Display the current data to the user each time the user wants to display the data  **def** **display\_list**(cd):  **print**()  **print**("Here**\'**s the current inventory list...")  **print**()  **print**(f"|{'\*ID' : <10}", f"{'\*Name' : <20}", f"{'\*eMail' : <20}", sep=' | ', end='|')  **print**()  **for** x **in** cd:  **print**("|{: <10} | {: <20} | {: <20}|".format(\*x))  **print**()    # 4. Save the data to a text file CDInventory.txt if the user chooses so  **def** **save\_cd**(cd):  **with** open('CDInventory.txt', 'a') **as** CDInventory:  **for** item **in** cd:  CDInventory.write(','.join([str(a) **for** a **in** item]) + '**\n**')  CDInventory.close()  **print**("File has been saved...")    # 5. Exit the program if the user chooses so.  **def** **exit\_thanks**():  # Thanks the user for using the program before import sys and calling sys.exit()  **print**("=============================================")  **print**("Thank you for using the CD Inventory Program")  **print**("=============================================")  **import** **sys**  sys.exit()    # Main Program #  **print**("....................................................................")  **print**("Hello, welcome to the CD Inventory List Program")  **print**("You may now proceed to explore this inventory list")  **print**("....................................................................")    # defining list function as a variable  cd = cd\_list()  menu = **1**  **while** menu **in** (**1**, **2**, **3**):  menu = menu\_choice()  **if** menu == **1**:  cd = add\_inventory(cd)  **elif** menu == **2**:  display\_list(cd)  **elif** menu == **3**:  save\_cd(cd)  **else**:  exit\_thanks() |