John Stevens

February 24, 2020

IT FDN 100 B

Assignment 05

CD Inventory(Expanded) Script

# Introduction

This document will show the modification of a pre coded script in Python that can load an existing list or create a new one, ask a user to input an inventory of CDs, display the inventory, delete any inputted CDs and then save the inventory list to a file with the loading and delete functionally added.

# Script Creation

I created my script in Visual Studio Code 1.41.1 as shown below in Figure 1,

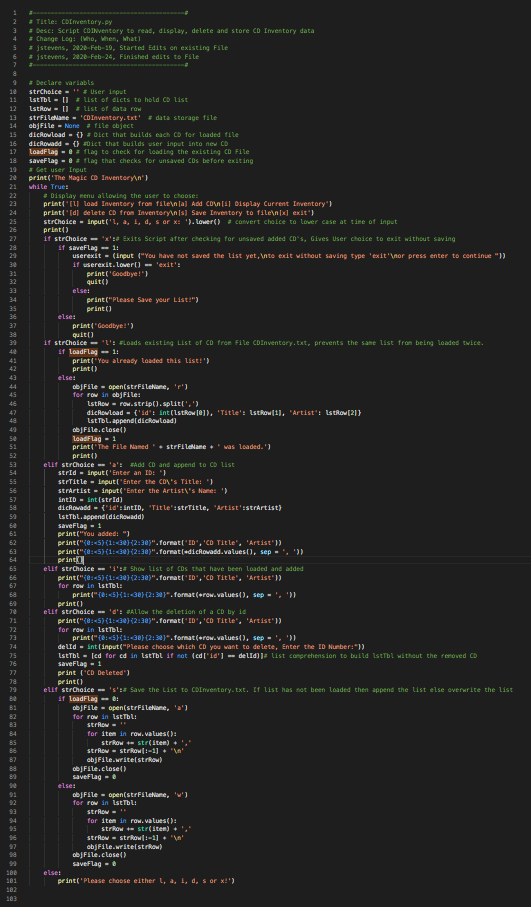


Figure – CDInventory.py script code

I began by declaring some variables not included in the original script, ‘dicRowload’ to set an empty dictionary for loading CDs for the file. ‘dicRowadd’ for adding an empty dictionary for adding CD from user input. Set the variable ‘saveFlag” to 0 for tracking unsaved changes and set ‘loadFlag’ to 0 for tracking when the list is loaded.

Then I added the functionality to load an existing file. It begins by looking at the ‘loadFlag’ so see if the CDInventory.txt file had been loaded, if so then the user is returned to main menu. If the user has not loaded the CDInventory.txt then the script will open the CDInventory.txt file with the r flag using the open fuction [[1]](#footnote-1)[https://www.geeksforgeeks.org/file-handling-python/](https://www.geeksforgeeks.org/file-handling-python/1)[[[2]](#footnote-2)1](https://www.geeksforgeeks.org/file-handling-python/1). The script then loops through the file by row using strip() and split() methods to form a list row stored in ‘lstRow’. Then ‘lstRow’ is used to form a dictionary saved into ‘dicRowload’ with each array of the list set to a value with a key(PYTHON PROGRAMMING FOR THE ABSOLUTE BEGINNER 3RD ADDITION, MICHAEL DAWSON, CENGAGE LEARNING 2010, P. 140-148). This new dictionary is then added to the ‘lstTbl’ with the append() method which will be the list of dictionaries. The file is then closed using the .close() method. Then the ‘loadFlag’ is set to 1 and we display the file that was loaded back to the user who is then returned back to the main menu.

To add the delete functionality I began by using the string .format method https://docs.python.org/3.7/library/string.html#formatspec[[3]](#footnote-3)2,to provide some common column alignment for the list and header to make the display back to user easier to read. Then the list is displayed back to the user and the script asks for the ID number of the CD to be deleted and stores it in ‘delId’. Once entered the script takes ‘delId’ and using list comprehension [[4]](#footnote-4)[https://www.geeksforgeeks.org/comprehensions-in-python/](https://www.geeksforgeeks.org/comprehensions-in-python/3)[[[5]](#footnote-5)3](https://www.geeksforgeeks.org/comprehensions-in-python/3), rebuilds the ‘lstTbl’ by excluding the entered ID in ‘delId’ by matching it to any of the entered id values in the id key value pair. Then the ‘saveFlag’ is set to 1 and the user is displayed a confirmation of the cd being deleted and returned back to the main menu.

I expanded some of the functionality of the existing If statements. For the exit option, I have the script enter another IF statement that looks if the saveflag is set to 1, if so it askes the user if they want to exit without saving. If the user enters ‘exit’ then the script exits using the quit function. Else the user is returned back to the main menu. If the saveflag is not set to 1 then the script exits. For the add option, I have the entered data formed into a dictionary stored into ‘dicRowadd’ which is then appended to ‘lstTbl’. The saveFlag is set to 1 and I display back to the user the cd information just entered. For the display option I added better formatting with the format() method and have ‘lstTbl’ looped through displaying only the values of each key value pair in each row. For the save option we first look for if the ‘loadFlag’ = 0. If so it means the list had not been loaded and so we should use the append flag for saving the file. If the ‘loadFlag’ = 1 then the list had been loaded and we should save the file with the write flag so we overwrite the file with the changes to the list.

# Performing the Script

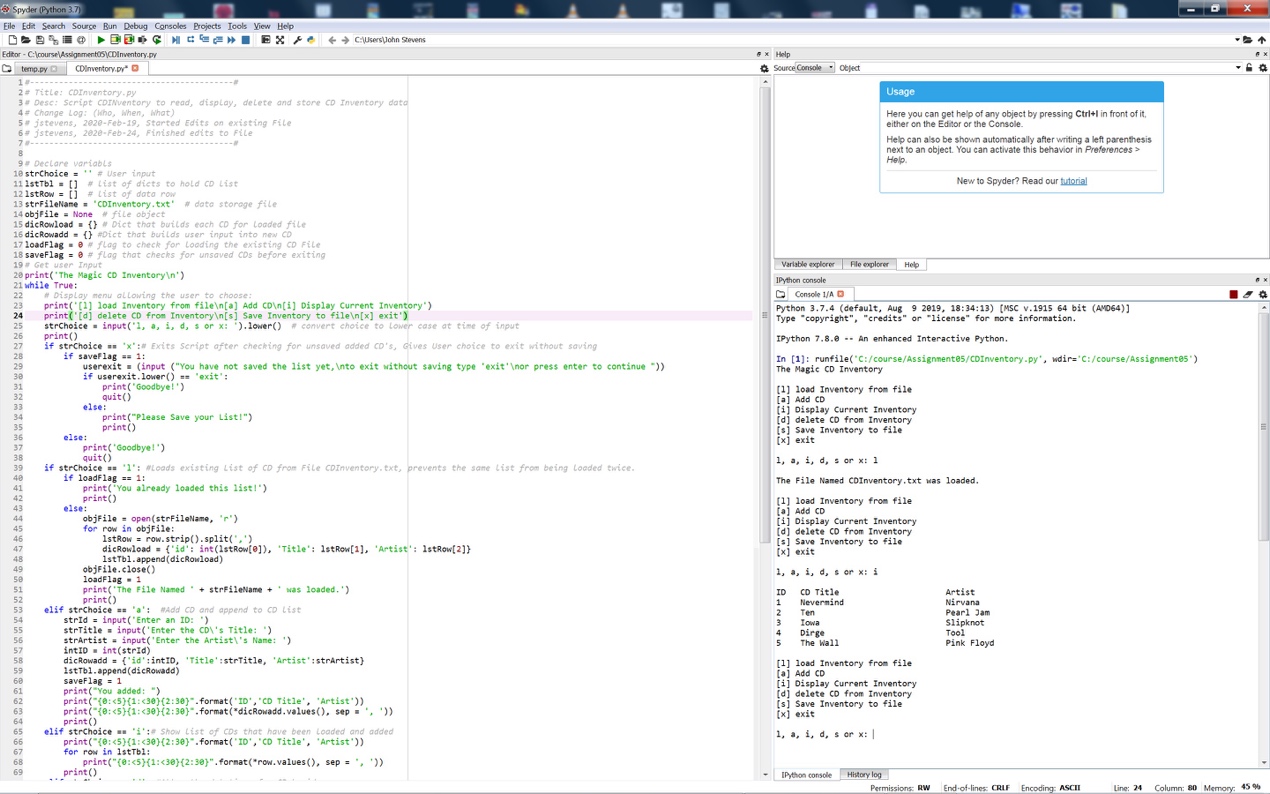


Figure - Script performing in Spyder.

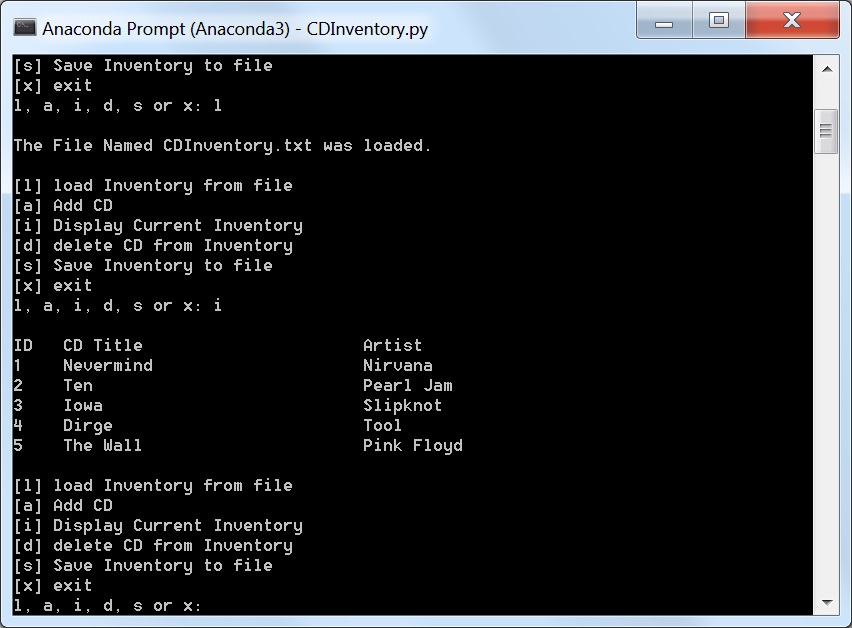


Figure 3 - Script performing in terminal

As seen in Figure 2 and Figure 3 I ran the script in Spyder and Terminal respectively. I loaded 4 test CDs from the file CDInventory.txt, then added a 5th and 6th CD and then deleted the 6th CD, displayed the list and saved the list to a file. When I checked the file CDInventory.txt I see that the input was saved correctly as seen in Figure 4.

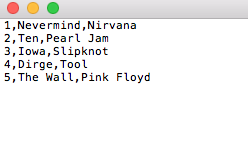


Figure 4 – CD input saved in CDInventory.txt file.

# Summary

In this assignment I was able to modify an existing script to use dictionaries and provide a functionality to load and deleting CDs from a list. The loaded information is stored in a dictionary as well as any CDs that the user inputs. This script could be improved by allowing the dictionary to be sorted and to proving a unique ID number to prevent the id key being used by more than one CD. This script and document and posted at https://github.com/pjfan73/Assignment05

1. 1 Retrieved 2020-Feb-19 [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. 2 Retrieved 2020-Feb-19 [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)
5. 3 Retrieved 2020-Feb-22 [↑](#footnote-ref-5)