Evan Malina

11/14/2021

Foundations of Programming Python

Assignment 5

# Introduction

For this assignment, I created a script that displays a menu, which requests an input from the user to select one of the options. The options are for the user to input CD data, to print the contents of the CD database, to write the contents of the database to a text file or lastly, to exit. The script includes while loops, inputs, lists, for loops, and opening and writing to files

# Code

## Menu

For the menu, I relied on Lab03-C and wrote the menu options through print(), and requested the user to input a number associated with the 4 menu options. Additionally, I created a starter list containing the 2 CDs provided in the assignment document. After the list is created, a while loop loops through the menu (and continues to display the menu after each selection is run through).

## Input CD Data

An if condition based on the user input allows entry into the CD input block of code. I had to request 3 inputs, and the three variables are added into a list. I wish I could input the variables straight into the list, but I didn’t have any luck with this, and didn’t see any example in the modules. Finally, the new list is appended to the original data.

## Display Data

Sections two through four are all entered using elif statements. A for loop separates each list entry of CD data and prints them on the screen. I chose to not format the results of the 2D list, other than to separate the items by line.

## Write Data to .txt file

After running through this program several times, the .txt file was getting lengthy with the same exact inputs so I changed the way the file was opened to “w” for write, rather than append. Similar to the display section, I added a for loop to loop through each row, but added an additional nested for loop to loop through the *columns*. This allowed me to format the entry to have the | character separating the 3 inputs. Don’t forget to close the .txt file 😊

## Exit

I chose to “break” from the while loop if 4, or anything that was not 1, 2 or 3 was input. This was done by if 1, elif 2, elif 3, elif 4, else.

In hindsight, I would maybe print a note in “write to data” section that the data did indeed get written (rather than just bringing up the menu again). I also would possibly add more blank inputs (‘press any key to continue’).

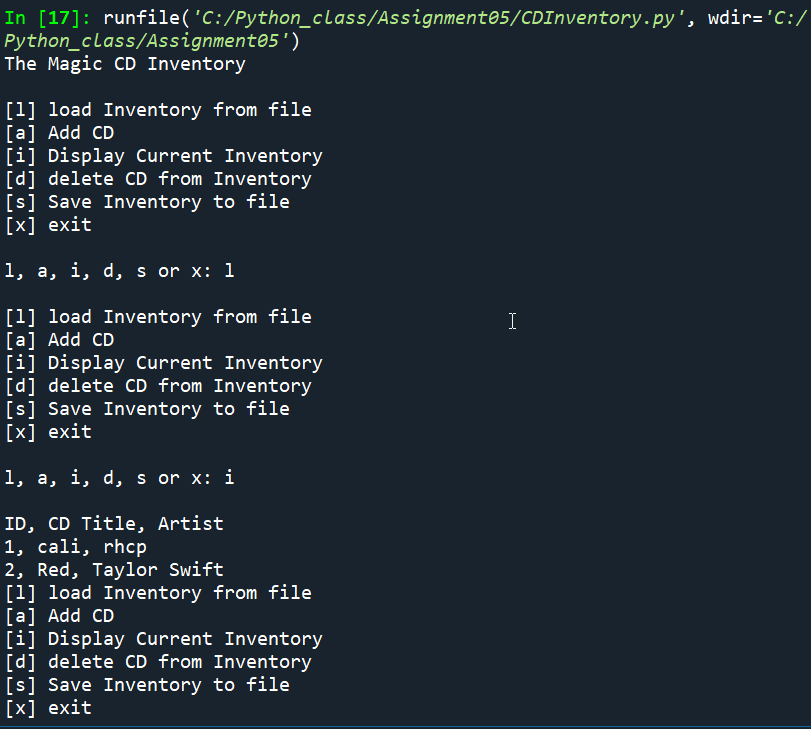


Figure - Spyder script run part 1

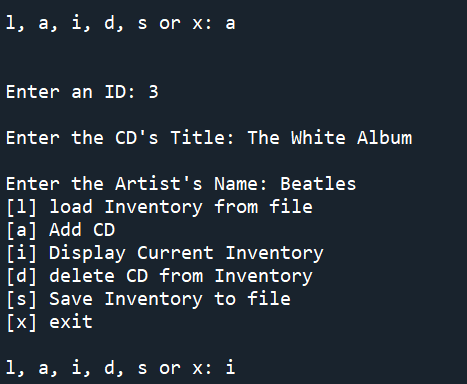


Figure - Spyder script run part 2

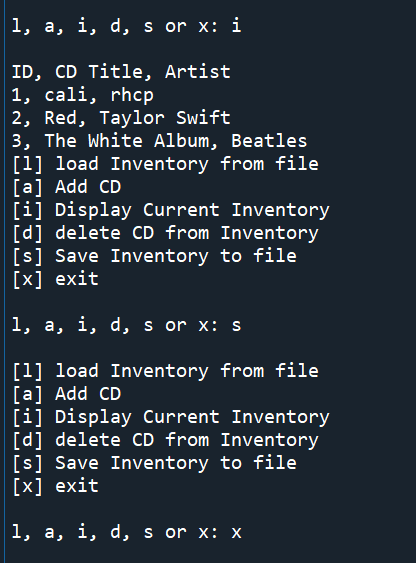


Figure - Spyder script run part 3

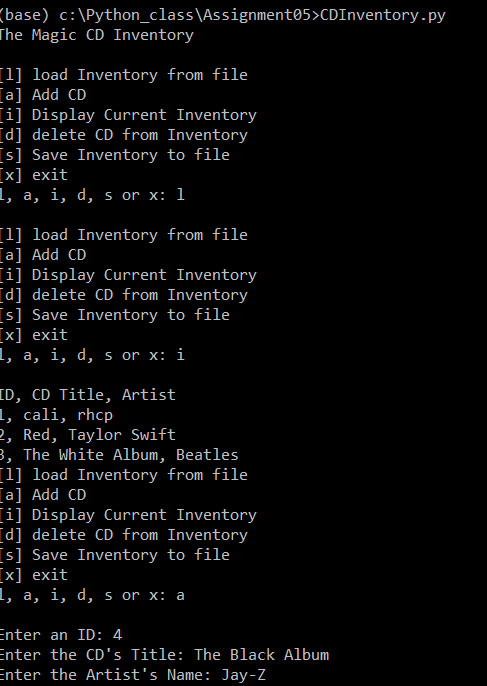


Figure - shell run of script

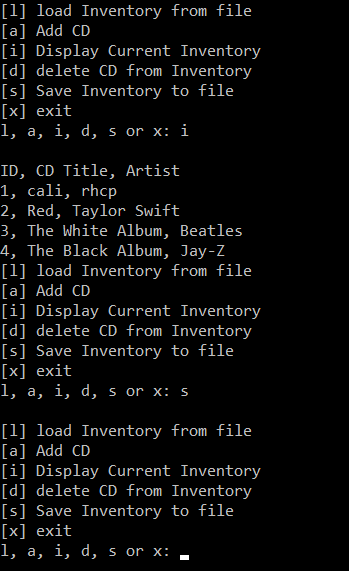


Figure - shell run of script part 2

# Summary

The class material prepared me to complete this assignment. I thoroughly enjoyed writing this more complex program. One nice addition to the program would be to require the ID be a number that didn’t already exist. And eventually, the list could be sorted! I like the way that the shell ran the script, in terms of new line entries (\n). Seeing it run in this formal helped me understand that the Spyder version is automatically adding spaces sometimes.