Palmer Bandy
August 12, 2020
IT FDN 110 B Su 20
Assignment 05

Dictionaries & The CD Inventory Program Continued

Introduction

In this assignment, I will discuss what a dictionary is and how to access data in a dictionary. Then, I will walk through how I modified the existing CD Inventory program to use a list of dictionaries instead of a list of lists, including how the program now reads data from the existing file into memory.

Dictionaries

Dictionaries in python are a data structure that allows you to specify names for individual elements as opposed to requiring to use indexing as with a list or a tuple. A dictionary is similar to JSON, but it is a data type specific to Python that is more flexible than JSON.

With a list in Python, you access individual elements using their index, like so: `myList[0]`. This would access the first element in the list. With a dictionary in Python, each element is actually a pair of a key and a value, where the key is a name that is used to identify the element, and the value is the data to be operated on. For example, accessing data from a dictionary would look like this: `myDict['myKey']`.

The CD Inventory Program

For this assignment, I updated the CDInventory_Starter.py file to use dictionaries as the inner data structure of lists instead of using a list of lists. I started out by adding code that would load any data from an existing CDInventory.txt file into memory so that the user of the program could load and view the data in the file. The program assumes that if there is an existing file, any data will be loaded into memory and then concatenated with any user-input data, so the distinction between the "Load Inventory from File" choice and the "Display Current Inventory" choice is primarily noticeable if the user has input a new data entry. For example, the first time you start the program and issue the "Display Current Inventory" command, you should see any data that is already in the file (including an extraneous header that is printed by the program itself). Once you add a CD through the "Add CD" option and

run “Display Current Inventory” again, you should see both your new entry and the existing data in the file displayed. However, if you select the “Load Inventory from File” option at this point, you should still only see the data that is in the file, since you have not yet saved your new user-input data to disk.

Once I had that working properly, I added the code to save all in-memory data to disk. Because the program loads data from the file into memory and the user-input data is then merged with it, it’s possible to always truncate the existing file within the “Save Inventory to File” flow but *before* actually saving the data to disk. This ensures that my file only contains the data that the user verified would be saved to disk when they chose to “Display Current Inventory.”

Here is my program running in Spyder:

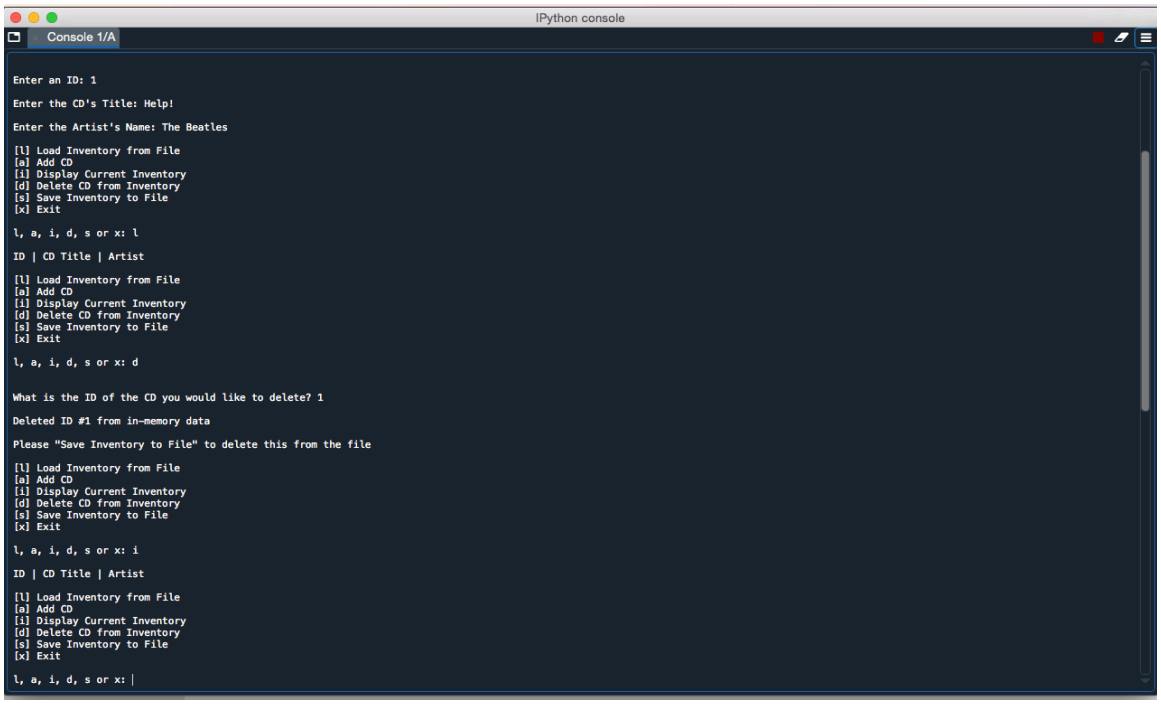

Figure 1 - Running in Spyder

And here is running it in my terminal:

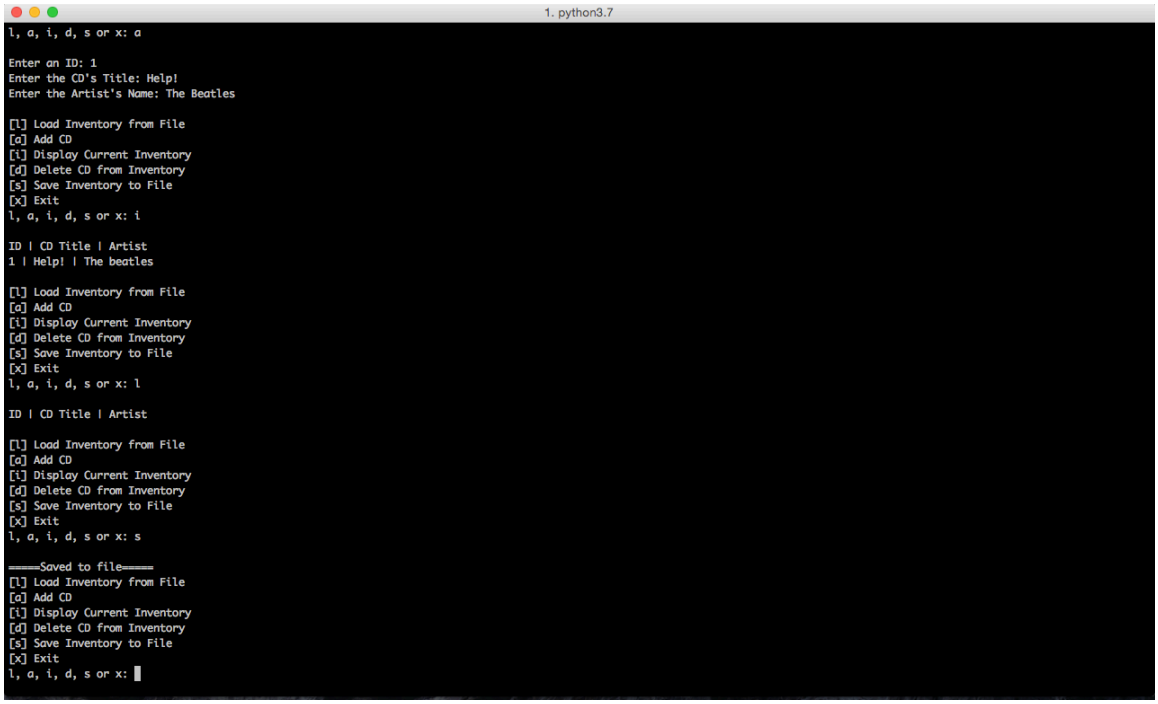

Figure 2 - Running from Terminal

Summary

For this assignment, I discussed what a dictionary is and how to access data in a dictionary. Then, I went over how I modified the existing CD Inventory program to use a list of dictionaries instead of a list of lists, including how the program now reads data from the existing file into memory.