Matthew Poehler

February 14, 2021

Foundations of Programming: Python

Assignment 05

CD Inventory Script

Introduction

In this document I will be covering the steps taken to modify the CD Inventory script that was provided by Professor Biesinger in Module 05 of the Foundations of Programming: Python course that asks users to input information pertaining to musical CDs and stores them in an inventory before saving to a text file. This will consist of a brief description of the new information provided in the Module 05 material such as dictionaries and the process of reading the data from a text file into a script to be used to in the program followed by the methods of deleting data from a list. I will then conclude with images of the script working in Spyder IDE and Anaconda Prompt.

Dictionaries

One of the main modifications we are asked to make in this script is to change the way the data is to be stored in the working memory of the program from a 2D list of lists to a 2D list of dictionaries. According to the material in Module 05, Dictionaries are like lists and other sequence types but are categorized as a mapping type which replace the index and storing in sequence with storing key and value pairs. Figure 1 provides an example of what dictionaries look like. They use the braces operator, “{}”.

```
In [4]: dicDemo = {'position': 'ST', 'name': 'Erling Haaland', 'number':9}

In [5]: print(dicDemo)
{'position': 'ST', 'name': 'Erling Haaland', 'number': 9}

In [6]: print(dicDemo['position'])
ST

In [7]: print(dicDemo['name'])
Erling Haaland

In [8]: print(dicDemo['number'])
9
```

Figure 1 - Dictionary demonstration in Spyder IDE

Reading Files into Scripts

The next part of the modifications to be made is reading or loading the contents of a text file into the program to be used. Listing 1 demonstrates how this is accomplished. The example shown in Listing 1 is from the CD Inventory script and was originally constructed by Professor Biesinger in the Module 05

material. According to the material this code not only reads data back into the program but formats it so the string of data we get from the file is turned into a list that we then convert to a dictionary to be used in the program.

```
33      objFile = open(strFileName, 'r')
34      for row in objFile:
35          lstRow = row.strip().split(',')
36          dicRow = {'ID': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}
37          lstTbl.append(dicRow) # loading existing data
38      objFile.close()
```

Listing 1 - Portion of CD Inventory script demonstrating how to read data into a program's memory.

Deleting Entries.

Last part of the modifications to be made is the process of deleting an entry. Listing 2 is an example of how this looks. Listing 2 consists of more code from the CD Inventory. Material found on www.w3school.com (external site) was used to create this portion of the script. The ability to remove a single entry was accomplished with the ".pop()" method and the ability to clear the entire contents of a list I used the ".clear()" method.[1]

```
54      print('[e] Delete latest entry \n[t] Delete current inventory')
55      delChoice = input('e or t: ')
56      if delChoice.lower() == 'e':
57          lstTbl.pop() # deleting latest entry
58      pass
59      if delChoice.lower() == 't':
60          lstTbl.clear() # deleting current inventory
```

Listing 2 - Code from CDInventory.py showing two methods to remove data from a list

CD Inventory Script

Now that all assigned modifications have been made Figure 2 and Figure show the CD Inventory script working in Spyder IDE and Anaconda Prompt.

[1] https://www.w3schools.com/python/python_lists_remove.asp retrieved 2021-Feb-14

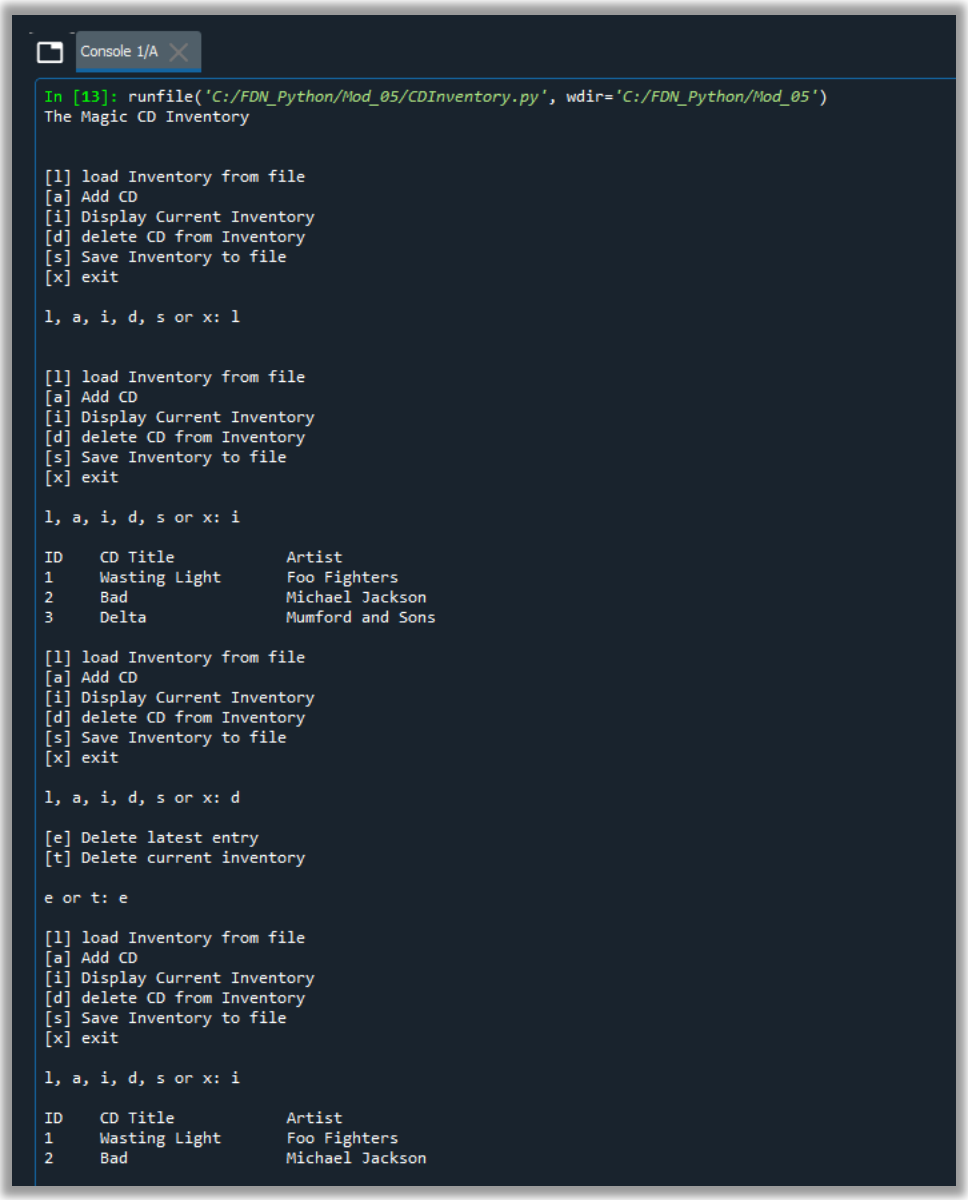

Figure 2 - CDInventory.py executed in Spyder IDE

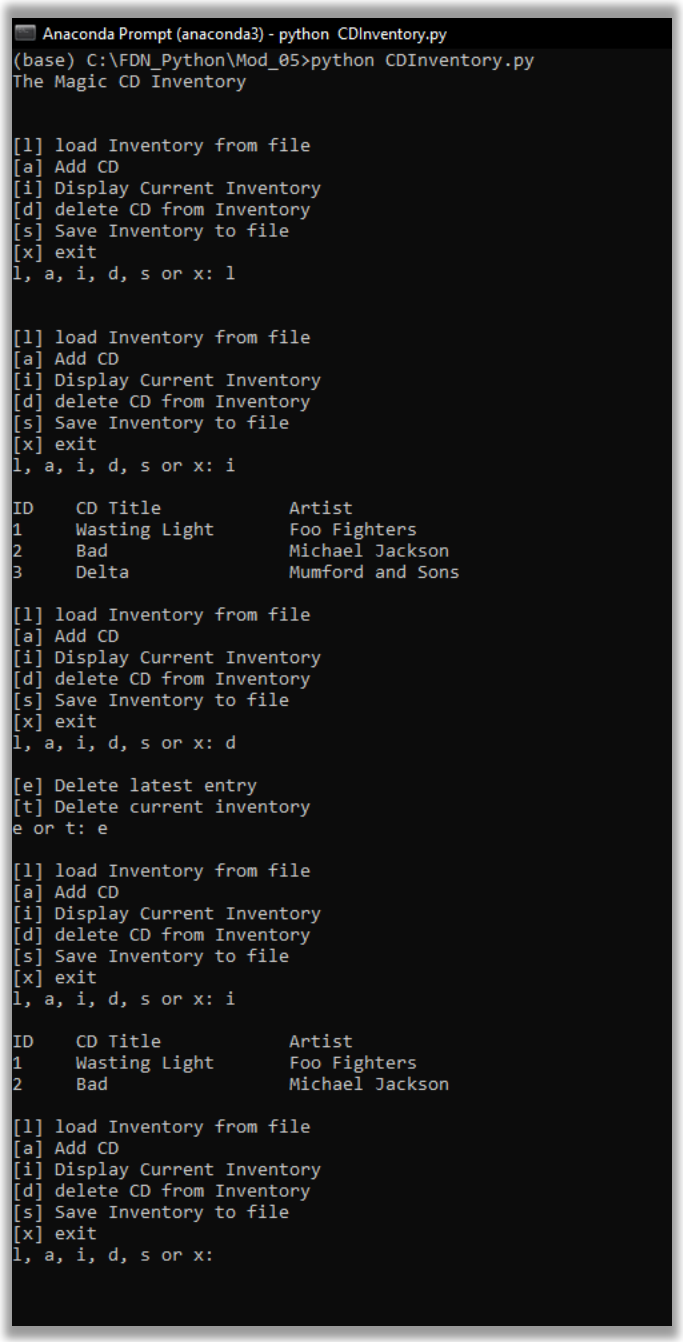

Figure 3 - CDInventory.py being executed in Anaconda Prompt

Summary

This document shows the step taken to modify the CD Inventory script provided by Professor Biesinger. Using the material in Module 05 of Professor Biesinger's Foundation of Programming: Python course as well as material from www.w3schools.com (external site), I was able to add the inner data structure of dictionaries, add the option to load existing data and the ability for users to delete data from the current

inventory. Modifying a script is equally easier and harder than creating one from scratch. I am looking forward to adding more to my python knowledge as this course continues.