Nicholas Johnson
Aug 11, 2021
Course: IT FDN 110
Assignment: 05
AddressBook.py
List and Dictionaries

**INTRODUCTION**
The *"CDInventory.py"* scrip was an adaptation of the previous assignment. Existing code was provided with some sections missing that needed to be adapted into the code to run. Changing the list to a dictionary, adding a library, and deleting an entry were the functionalities that were needed to be implemented. However more fuctioalities of this script were needed to be changed to work properly. Overall this project took much longer than expected and I didnt document my steps well within the few days it took to run

**DETAILS**
The script was meant to perform 6 different actions that are chosen by a user. The first action was to load the inventory from a file. The other actions were to add a CD to a dictionary, display the current items in the dictionary, delete a item in the dictionary, save the user inputs to the dictionary by writing it to a file, and exit the program .

**LISTS + LOADING**
Changing the lsit to a dictionary was easily accomplished by replacing the list with a *"dic = {}"* dictionary that will hold the key and values.  Loading the existing file was not easy to begin with due to the need to confirm if the code works. Adding a CD to the dictionary and then writing it to the file was first implemented so that I could verify this portion of the script worked.  The existing code was intended to write to a list instead of a dictionary.

**ADDING AND DISPLAYING**
Adding a CD to the inventory had existing code that worked well with the existing code. Populating the dictionary was accomplished though setting the ID number input as the key and the the title and artist inputs  as the two values. Option 3 to Display the current inventory also needed to be changed to print the dictionary structure. The dictionary key was used as the ID and two variables were set to  title and artist to print and display the inputs of the user from Option 2.

**DELETE AND SAVE**
Deleting a row number was written to first capture a users input, change it to a new vanible that is equal to an integer, and then delete the row number input from the dictionary. Finally, saving writes the user inputs to the text file that is a saved text version of he dictionary. This is done by opening the file, writing to it, and then closing it.

**SUMMARY**
Overall this project took much longer than expected and I didnt document my steps well within the few days it took to get the script working. Working with a dictionary made me see how it is applied to in the work I do in Geospatial and how ther is a key and value for data tables.

**SPYDER SCREENSHOT:**

/Users/nickjohnson/_FDNPrograming/Mod_05/Assignment05/CDInventory.py

Source | Editor | Object

f

Generic base class for TextIO and BinaryIO.

This is an abstract, generic version of the return of open().

NOTE: This does not distinguish between the different possible classes (text vs. binary, read vs. write vs. read/write, append–only, unbuffered). The TextIO and BinaryIO subclasses below capture the distinctions between text vs. binary, which is pervasive in the interface; however we currently do not offer a way to track the other distinctions in the type system.

Variable explorer | Help | Plots | Files

temp.py | AddressBook.py | CDInventory.py

```
1   #------------------------------------------------#
2   # Title: CDInventory.py
3   # Desc: Starter Script for Assignment 05
4   # Change Log: (Who, When, What)
5   # DBiesinger, 2030-Jan-01, Created File
6   # NJohnson, 2021-Aug-08, updated
7   #------------------------------------------------#
8
9   # Declare variabls
10
11  strChoice = '' # User input
12  dic = {}
13  # TODO replace list of lists with list of dicts
14
15
16
17  strFileName = 'CDInventory.txt'  # data storage file
18  objFile = None  # file object
19
20  # Get user Input
21  print('The Magic CD Inventory\n')
22  while True:
23      # 1. Display menu allowing the user to choose:
24      print('[1] Load Inventory from File\n[2] Add CD\n[3] D
25      print('[4] Delete CD from Inventory\n[5] Save Inventor
26      strChoice = input('1, 2, 3, 4, 5 or x: ').lower()  # c
27      print()
28
29      if strChoice == 'x':
30          # 5. Exit the program if the user chooses so
31          break
32
33
34
35      if strChoice == '1':
36          with open('CDInventory.txt','r') as f:
37              data = f.read()
38              print(data)
39
40
41
42
43          # TODO Add the functionality of loading existing c
44          pass
45
46
```

Console 1/A

```
File "/Users/nickjohnson/opt/anaconda3/lib/python3.8/site-
packages/ipykernel/kernelbase.py", line 904, in
_input_request
    raise KeyboardInterrupt("Interrupted by user") from None

KeyboardInterrupt: Interrupted by user


In [38]: runfile('/Users/nickjohnson/_FDNPrograming/Mod_05/
Assignment05/CDInventory.py', wdir='/Users/nickjohnson/
_FDNPrograming/Mod_05/Assignment05')

Output from spyder call 'get_namespace_view':
The Magic CD Inventory

[1] Load Inventory from File
[2] Add CD
[3] Display Current Inventory
[4] Delete CD from Inventory
[5] Save Inventory to file
[x] Exit

1, 2, 3, 4, 5 or x:
```

IPython console | History

strTitle

LSP Python: ready | conda: base (Python 3.8.8) | Line 37, Col 12 | ASCII | CRLF | RW | Mem 71%

**TERMINAL SCREENSHOT:**

```
(base) nickjohnson@Nicholass-MacBook-Pro Assignment05 % python CDInventory.py
The Magic CD Inventory

[1] Load Inventory from File
[2] Add CD
[3] Display Current Inventory
[4] Delete CD from Inventory
[5] Save Inventory to file
[x] Exit
1, 2, 3, 4, 5 or x: 1

{2: {'title': 'k', 'artist': 'k'}}
[1] Load Inventory from File
[2] Add CD
[3] Display Current Inventory
[4] Delete CD from Inventory
[5] Save Inventory to file
[x] Exit
1, 2, 3, 4, 5 or x: 2

Enter an ID: 
```