Laurel Warner

11/14/2021

Foundations of Programming

Assignment05

Lists and Dictionaries

# Introduction

In this module we continued with some work on lists but also learned about another data type called a dictionary which is another way that Python has for storing data in pairs. We also learned about GitHub and how to set up code for error handling and also the Separation of Concerns programming pattern.

# Dictionaries vs. Lists

Dictionaries and Lists can both hold data in Python. A dictionary hold data in Key: Value pairs only and a List can have as many items in it as you want. An index in a list refers to the position that the item has, and a key is matched with a specific value in a dictionary. To read data from a file into a list you have to open it and then print it out or unpack it. To read data from a file into a dictionary you unpack the elements and then print them into key: value pairs. I think it makes sense to organize data in a 2-dimensional way because it’s just like an Excel sheet or a table, which I am very familiar with.

# Separation of Concerns, Error Handling and Functions

Separation of Concerns is a design principle for separating a computer program into distinct sections such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program. It’s basically a way to keep the code organized so that it’s easier to read and see what is going on. You can use a function to organize your code because they allow you to group a set of statements and make them accessible via a given name. It can save time because you would only type out the code for the function once and then just type in the function name any time you wanted to run that particular code. A script template is useful because then you don’t have to keep typing and retyping the same header in your scripts. If you have information you always want at the top of your script, then just type it in once in the template and then any time you open up a new file it will already be there. Error handling is useful because it allows you to have a way to keep your program running, even when something doesn’t work and your program won’t crash causing you to have to restart it every time. GitHub is a website that allows you to sign up and save your code into “repositories” that you make and share the code with others. It’s a good way to see if someone else has come up with another way to do what you might be trying to do. GitHub’s mascot is a 5 armed “OctoCat” (He’s pretty cute).

# CD Inventory with Dictionaries

For this assignment I had to start with some starter code provided to me as a potential solution to last weeks assignment. I then had to convert all the lists to dictionary functionality and add some additional functions of loading the data from the file and deleting an entry. I thought I might not need any assistance this week because of the fact that we had solution code provided to us and also because the 2 lab exercises were so similar to the assignment. However, that was not the case. With assistance, I did realize that my code in the first couple lines had the wrong brackets for my lstTbl and dicRow variables which I think caused a problem. Once I ran the code it seems as though my deleting code doesn’t really work but I don’t know why. The code looked good when my tutor and I were looking at it, but all the rest of the code seems to be functional as it should. Here is the screenshot of my code working in Spyder: (I don’t’ think I wanted to delete any of my entries in this so I didn’t realize that the deleting function wasn’t working).

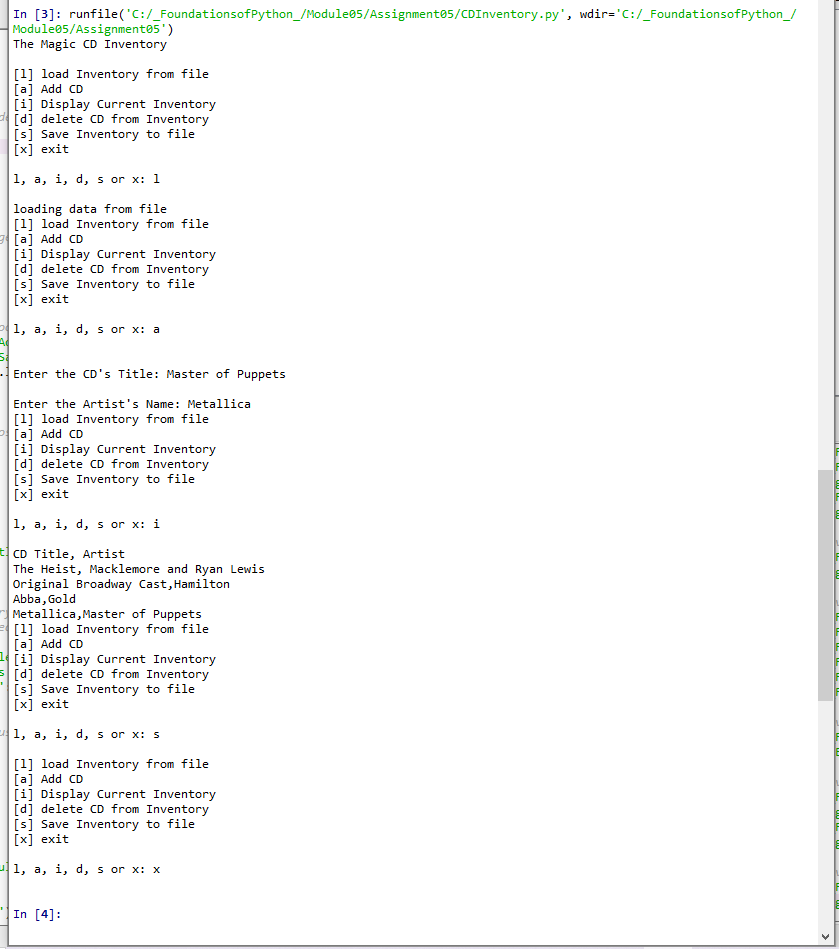


Figure 1

Here is the screenshot of my code in the terminal window: (This was when I first tried to actually delete an entry and found that it’s not working.)

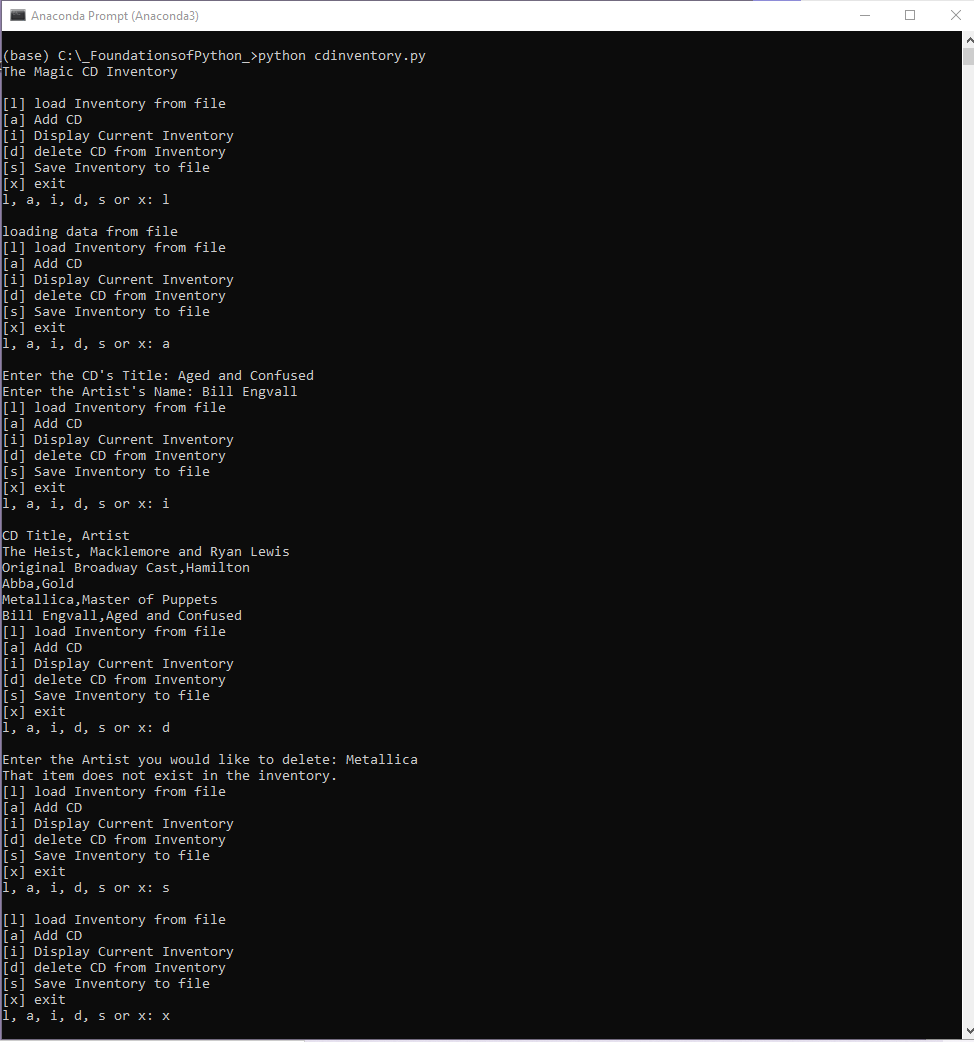


Figure 2

# Summary

I learned the differences between a list and a dictionary, how to set up code for error handling, as well as separation of concerns. I also learned about GitHub and uploaded my first code to a repository. Here is the link for my GitHub repository: https://github.com/lellenn/Assignment\_05

# Appendix

1. *#------------------------------------------#*
2. *# Title: CDInventory.py*
3. *# Desc: Added Functionality in a CD Inventory*
4. *# Change Log: (Who, When, What)*
5. ***# DBiesinger, 2030-Jan-01, Created File***
6. *# LWarner, 2021-Nov-11, Modified and added code*
7. *#------------------------------------------#*
9. *# Declare variables*
11. strChoice = '' *# User input*
12. lstTbl = [] *# list of dicts to hold data*
13. dicRow = {} *# dictionary of data row*
14. strFileName = 'CDInventory.txt' *# data storage file*
15. **objFile = None *# file object***
17. *# Get user Input*
18. **print**('The Magic CD Inventory**\n**')
19. **while** True:
20. ***# 1. Display menu allowing the user to choose:***
21. **print**('[l] load Inventory from file**\n**[a] Add CD**\n**[i] Display Current Inventory')
22. **print**('[d] delete CD from Inventory**\n**[s] Save Inventory to file**\n**[x] exit')
23. strChoice = input('l, a, i, d, s or x: ').lower() *# convert choice to lower case at time of input*
24. **print**()
26. **if** strChoice == 'x':
27. *# 5. Exit the program if the user chooses so*
28. **break**
29. **if** strChoice == 'l':
30. **print('loading data from file')**
31. lstTbl.clear()
32. objFile = open(strFileName, 'r')
33. **for** row **in** objFile:
34. lstRow = row.strip().split(',')
35. **dicRow = {'Artist': lstRow[0], 'Title': lstRow[1]}**
36. lstTbl.append(dicRow)
37. objFile.close()
38. **pass**
39. **elif** strChoice == 'a': *# no elif necessary, as this code is only reached if strChoice is not 'exit'*
40. ***# 2. Add data to the table (2d-list) each time the user wants to add data***
41. objFile = open(strFileName, 'a')
42. strTitle = input('Enter the CD**\'**s Title: ')
43. strArtist = input('Enter the Artist**\'**s Name: ')
44. dicRow = {'Artist': strArtist, 'Title': strTitle,}
45. **lstTbl.append(dicRow)**
46. **elif** strChoice == 'i':
47. *# 3. Display the current data to the user each time the user wants to display the data*
48. **print**('CD Title, Artist')
49. **for** row **in** lstTbl:
50. **strRow = ''**
51. **for** item **in** row.values():
52. strRow += str(item) + ','
53. strRow = strRow[:-1]
54. **print**(strRow)
55. **elif strChoice == 'd':**
56. item = input('Enter the Artist you would like to delete: ')
57. **if** item **in** strFileName:
58. **del** strFileName[item]
59. **print**('That item has been deleted')
60. **else:**
61. **print**('That item does not exist in the inventory.')
62. **pass**
63. **elif** strChoice == 's':
64. *# 4. Save the data to a text file CDInventory.txt if the user chooses so*
65. **objFile = open(strFileName, 'w')**
66. **for** row **in** lstTbl:
67. strRow = ''
68. **for** item **in** row.values():
69. strRow += str(item) + ','
70. **strRow = strRow[:-1] + '\n'**
71. objFile.write(strRow)
72. objFile.close()
73. **else**:
74. **print**('Please choose either l, a, i, d, s or x!')