Greg Marable

2020-Feb-24

IT FDN 100 B Wi 20: Foundations of Programming: Python

Assignment05

Python Script 05 – CDInventory.py

# Introduction

This will be a three-part homework assignment much like most of the other weeks before. First, I will go over the eleven questions we were trying to answer. Second, I will show the results of the labs and tell why we got the results that we did. Next, I’ll show the updated results of the CDInventory.py script and my thoughts on the task. Finally, a summary to wrap everything up

# Topic 1 – Questions

• What is the difference between a Dictionary and a List?

A list is a collection type and a dictionary are a mapping type. They are both used to store and retrieve data but do it in different ways. Lists use indices to find where information is and dictionaries use keys.

• What is the difference between an index and a key?

An index is like a point on a map. It shows you exactly where the information is being held. A key is like a category and the information is held behind it. (Module 05 Page 07-08)

• How do you read data from a file into a list?

Declare the variables. lstRow = [] for a list, strFileName = ‘text.txt’ for whatever document that needs to be opened. And the you open use the open function and specify what the plan is after opening the document. After that the information that is pulled from the document has to be formatted into a list.

• How do you read data from a file into a dictionary?

The process is very similar to a list. The major difference comes from the formatting. {} specifies a dictionary. The information needs to be assigned a key and a value.

• Why is it making sense to organize data in a 2-dimensional way?

It makes sense because there can be multiple categories of things and what are you going to go. Just make a single column of things and make it long? No. By organizing information in a 2-D way we are able to have different assignments to different values and access them in a relatively easy to understand way.

• What is the programming pattern ‘Separation of Concerns”?

A design principle that says to separate a computer program into different and distinct sections so they address different things in your code. (Module 05 Page 12-13)

• How would you use a function to organize your code?

I would use functions to define the separate ways I want to display outputs from my code. I could that that function into the processing area and use the function where ever I need it. (Module 05 Page 14)

• Why is a script template useful?

It standardizes headers and gives basic template to start writing code in to. (Module 05 Page 15-16)

• Why is error handling (try-except) useful?

Error handling with the try-except is useful because it can keep our script up and running and even tell us what happened and why. (Module 05 Page 16-17)

• What is GitHub and why is it used?

GitHub is based off of Git, which was developed for Linux as a version control system. It lets developers create and change code on the fly and located in a central repository. People can see the changes made to a program and also make more changes to the program. GitHub uses this idea and expands upon it. Adding extra features such as an added GUI and even a social media platform. (Module 05 Page 18)

• What is GitHub’s mascot?

GitHub’s mascot is an anthropomorphized “octocat” with five octopus-like arms ([GitHub Mascot](https://en.wikipedia.org/wiki/GitHub#Mascot)).

# Topic 2 – Labs

# LAB04-A

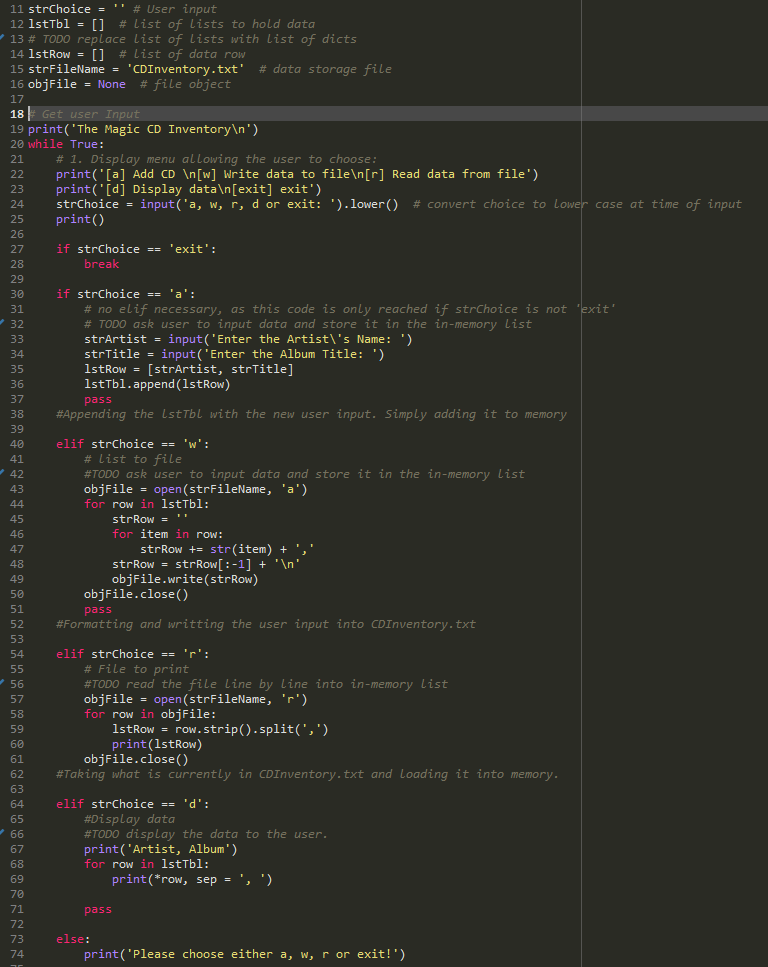


Figure - LAB05-A - Script

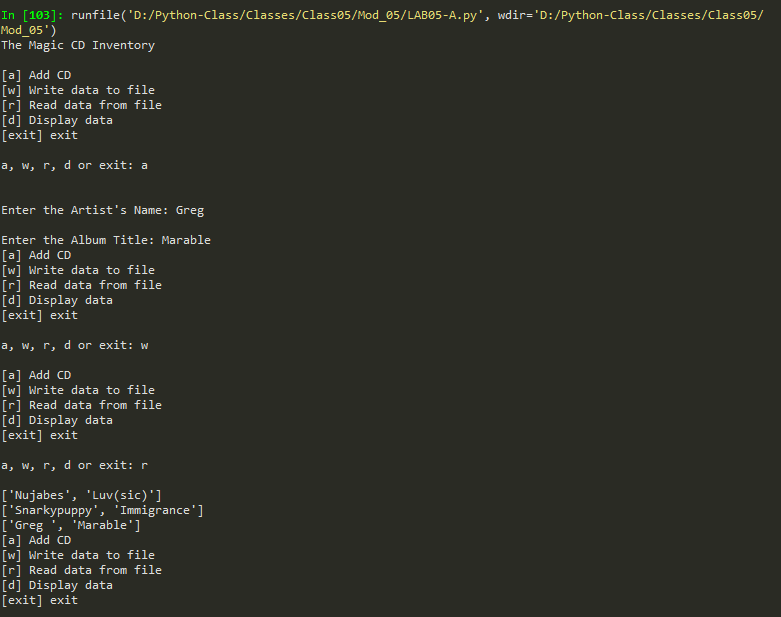
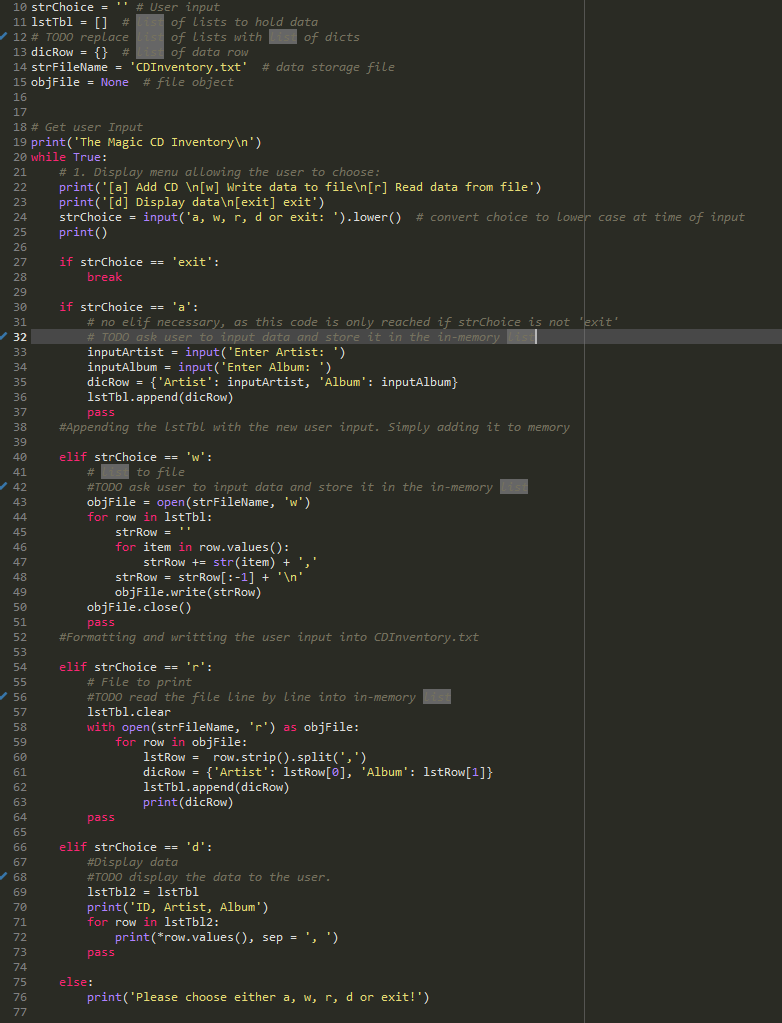


Figure - LAB05-A - Output

# LAB04-B



Things that I wanted to improve when I saw the output from LAB05-B was to change the letter assignments to numbers assignments on the menu. To me it’s a little easier to go from 1 to whatever number I need instead of random letters on the keyboard. I changed the menu options from letters to numbers in the assignment and it looks nicer to me. I’m not sure if it will be the same for everyone though. Another thing I wanted to change was positioning. I like knowing what data I’m working with first, so displaying what is in the file first should have the top position in my book. Then add to list, then display, then write to file, and finally exit. To me that makes the most sense. On top of those two things I need more visuals with what’s happening. A few well-placed print() and you could see what the computer is seeing and know that what you’re processing the right stuff.

Figure - LAB05-B - Script

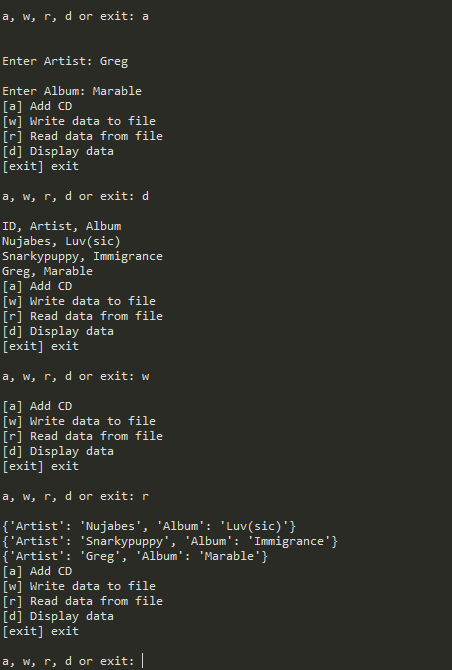
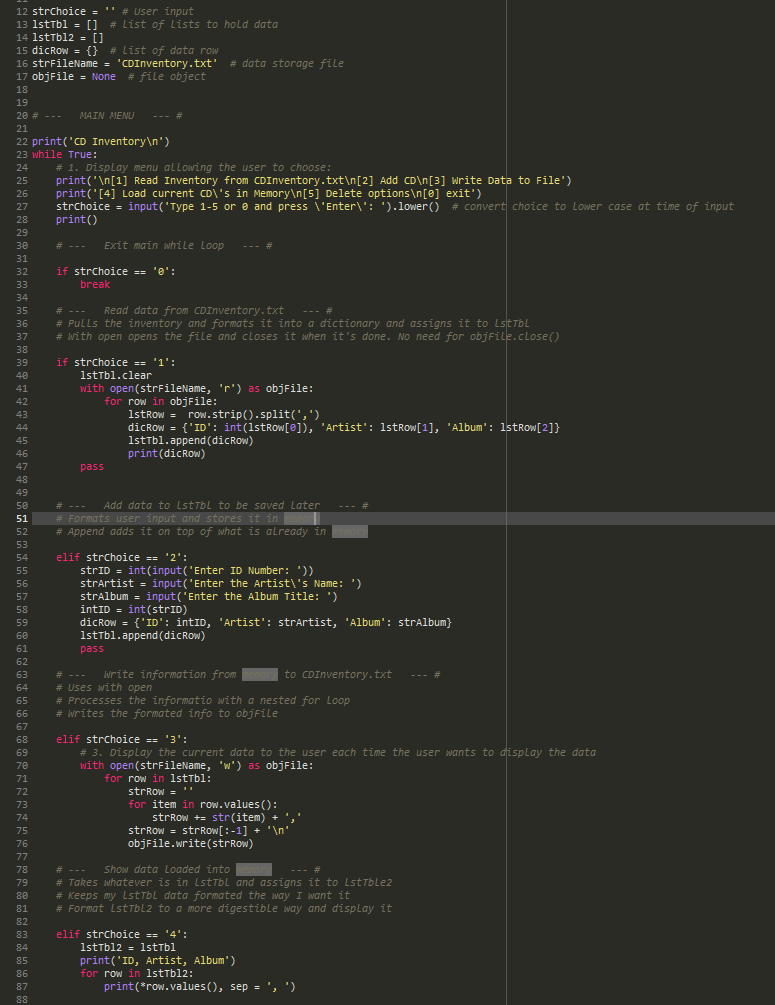


Figure - LAB05-B - Output

# Topic 3 – Assignment05 – CDInventory.py



This assignment was fun for me. It was an excellent opportunity to change things around in the script and add several things that the script really needed. The main thing that I added was two more while loops for a delete menu. I don’t think that it was totally needed but I liked the idea of being able to either delete single entries on my lists or just delete the entire thing. Though on the final while loop to delete your entire list I made the user have to type in ‘yes’ to agree to delete the list instead of just pressing another number like the rest of the script. Overall, I think the delete menu worked out and it’s a nice addition. The second major thing that I did differently was to add with open. I’m a fan. It’s the small things like not have to add .close() everywhere to the file isn’t constantly open. With the with open I just know that it’ll automatically close when I’ve done what I need to do. Finally it’s minor but from the extra reading I really thought .pop seemed like a great way to delete things from my dictionary. I had to make it work in my script. I’m glad I did. It didn’t require much to make it work.

Figure - Assignment05 - CDInventory.py - Script pt-1



Figure - Assignment05 - CDInventory.py - Script pt-2

# Summary

This assignment had totally different problems than last week. Overall, I think that this one was more enjoyable because the concepts from last week started to sink in a little bit more. I really enjoyed adding some of the extra bits from the additional reading onto my script. The ‘with open’ was a great way for me not have to worry about closing the file I’m working on. Also, the .pop is a fun way to delete something in a dictionary. I can see how what we’re learning in these last two assignments can really become something very powerful down the road. The main thing for me is that I really do need to change the way I think to understand these concepts.

# Appendix

1. #------------------------------------------#
2. # Title: CDInventory.py
3. # Desc: Script CDINventory to store CD Inventory data
4. # Change Log: (Who, When, What)
5. # GMarable, 2020-Feb-18, Created
6. # Gmarable, 2020-Feb-23, File Modified to Assignment05
7. #
8. #------------------------------------------#
10. strChoice = '' # User input
11. lstTbl = []  # list of lists to hold data
12. lstTbl2 = []
13. dicRow = {}  # list of data row
14. strFileName = 'CDInventory.txt'  # data storage file
15. objFile = None  # file object
17. **print**('CD Inventory\n')
18. **while** True:
19. **print**('\n[1] Read Inventory from CDInventory.txt\n[2] Add CD\n[3] Write Data to File')
20. **print**('[4] Load current CD\'s in Memory\n[5] Delete options\n[0] exit')
21. strChoice = input('Type 1-5 or 0 and press \'Enter\': ').lower()
22. **print**()
24. **if** strChoice == '0':
25. **break**
27. **if** strChoice == '1':
28. lstTbl.clear
29. with open(strFileName, 'r') as objFile:
30. **for** row **in** objFile:
31. lstRow =  row.strip().split(',')
32. dicRow = {'ID': int(lstRow[0]), 'Artist': lstRow[1], 'Album': lstRow[2]}
33. lstTbl.append(dicRow)
34. **print**(dicRow)
35. **pass**
37. **elif** strChoice == '2':
38. strID = int(input('Enter ID Number: '))
39. strArtist = input('Enter the Artist\'s Name: ')
40. strAlbum = input('Enter the Album Title: ')
41. intID = int(strID)
42. dicRow = {'ID': intID, 'Artist': strArtist, 'Album': strAlbum}
43. lstTbl.append(dicRow)
44. **pass**
46. **elif** strChoice == '3':
47. with open(strFileName, 'w') as objFile:
48. **for** row **in** lstTbl:
49. strRow = ''
50. **for** item **in** row.values():
51. strRow += str(item) + ','
52. strRow = strRow[:-1] + '\n'
53. objFile.write(strRow)
55. **elif** strChoice == '4':
56. lstTbl2 = lstTbl
57. **print**('ID, Artist, Album')
58. **for** row **in** lstTbl2:
59. **print**(\*row.values(), sep = ', ')
61. **elif** strChoice == '5':
63. **while** True:
64. **print**('\n[l] List all current enteries\n[2] Delete a single entry\n[3] Delete all entries\n[4] Return')
65. usrChoice1 = input('Select from 1 to 4 and press Enter: ')
67. **if** usrChoice1 == '4':
68. **break**
70. **if** usrChoice1 == '1':
71. lstTbl2 = lstTbl
72. **print**('ID, Artist, Album')
73. **for** row **in** lstTbl2:
74. **print**(\*row.values(), sep = ', ')
76. **if** usrChoice1 == '2':
77. #elitem = None
78. delitem = int(input('Enter CD ID to delete: ')) - 1
79. **print**(lstTbl2)
80. with open(strFileName, 'r') as objFile:
81. lstTbl.pop(delitem)
82. **print**('\nDeleted ID - ',delitem)
84. **if** usrChoice1 == '3':
85. **while** True:
86. usrChoice2 = input('Are you sure you want to delete all of your entries?'
87. '\n\nType \'Yes\' or \'No\' and press Enter: ').lower()
88. **print**('\n[Yes] Delete all entries\n[No] Return')
90. **if** usrChoice2 == 'yes':
91. with open(strFileName, 'r') as objFile:
92. lstTbl.clear()
93. **print**('\nAll CD inventory deleted and file saved')
94. **break**
95. **break**
97. **if** usrChoice2 == 'no':
98. **break**
100. **else**:
101. **print**('Please choose either 1, 2, 3, 4, 5 or 0!')