Gavin Chan

02/21/2020

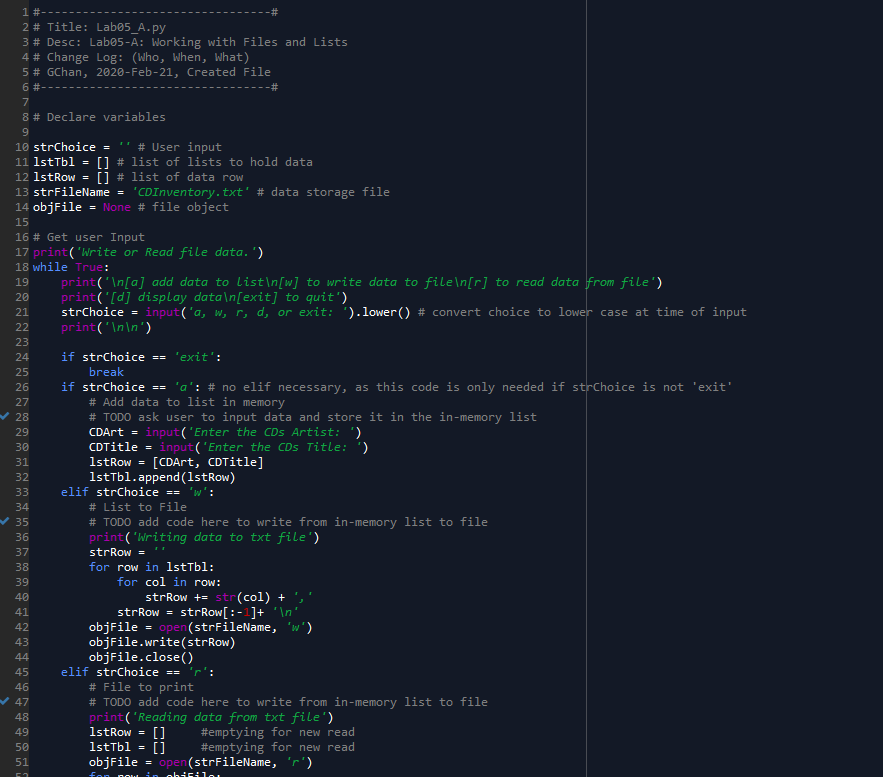
IT FDN 100

Assignment 05

# Introduction

In this week’s module and assignment, the student was introduced to working with the concept of “building off of code already provided”. Dictionaries were also introduced, and implemented in the lab.

# Lab 05-A: Working with Files and Lists

Lab 05-A has the student build off of a sample code provided by the instructor. The script is to take an input from the user (a, w, r, d, or exit) and perform an action based on the input. Figure 1 contains a snippet of the code, and Figure 2 is the output. The code will also be found in the appendix (as it is rather lengthy).

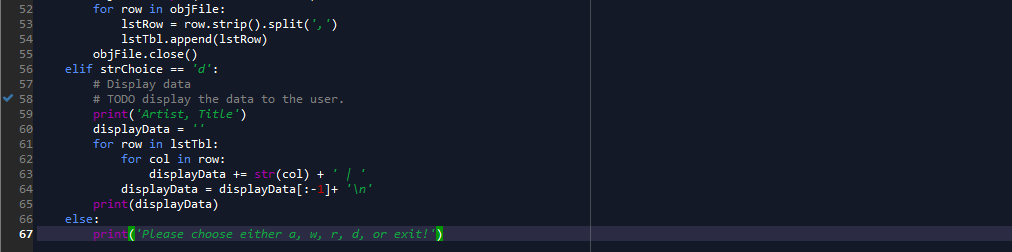
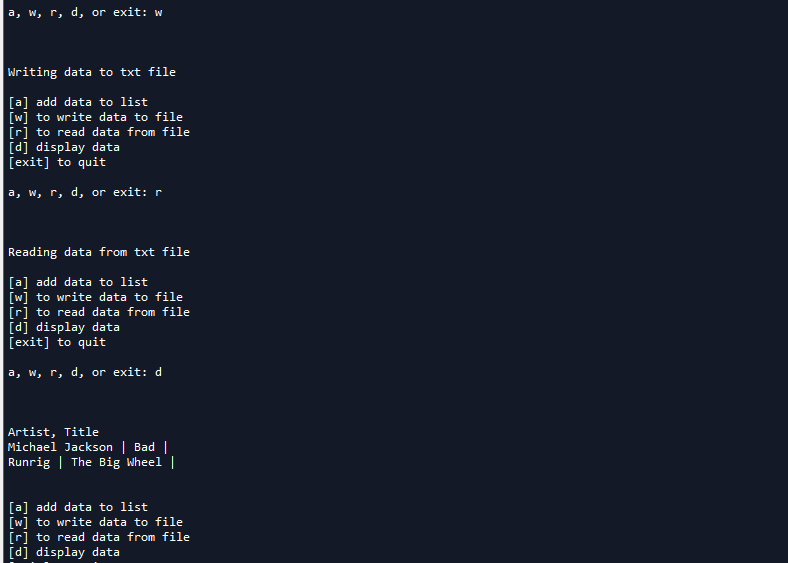
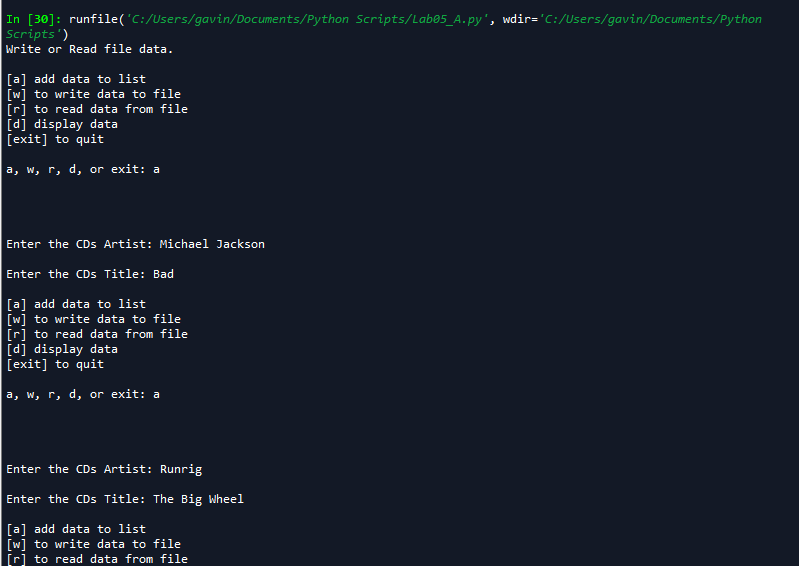


Figure - Code for Lab05-A



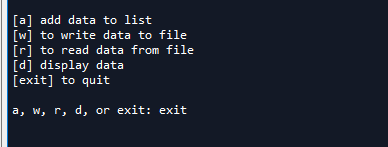


Figure - Output for Lab 05-A

This code works by using 2D lists (similar to Assignment 4). For option “a”, it takes the inputs from the user and appends the inputs onto the 2D list created in Lines 11-12. Option “w” will break the list up into individual components and uses a nested for loop to write into the .txt file. Option “r” reads what is from the text file. The empty lists in Lines 49-50 are needed because if option “a” is run before “r”, option “r” will add onto the already created lists, leading to an incorrect output. Option “d” uses a nested for loop similar to option “w”, but actually prints the list.

## Lab 05-B: Working with Dictionaries

Lab 05-B takes the code from Lab 05-A and uses dictionaries instead of lists. Figure 3 is the code that I used for the lab, and Figure 4 is a snippet of the only part of the output that is different from Lab 05-A. The code will also be found in the Appendix.

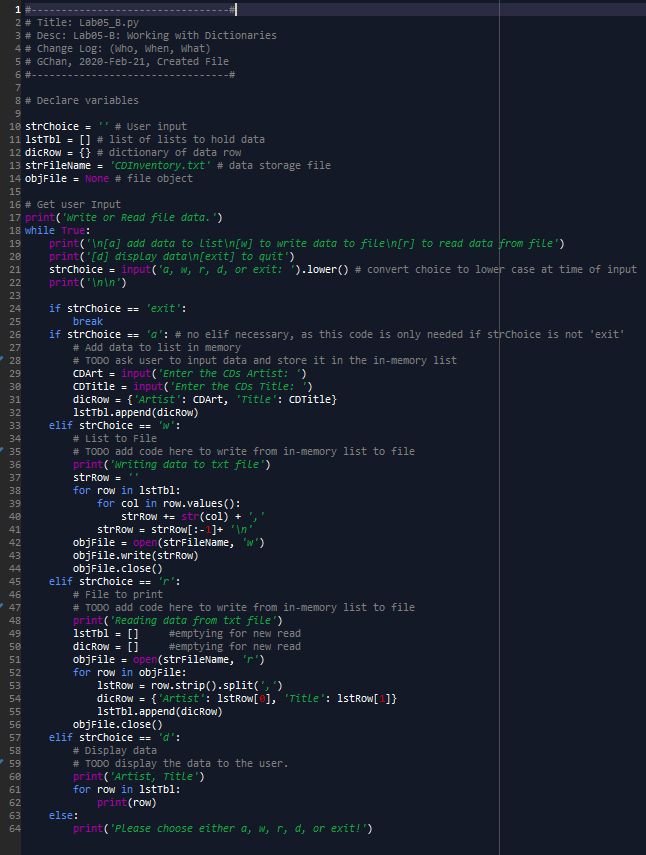


Figure - Code for Lab 05-B

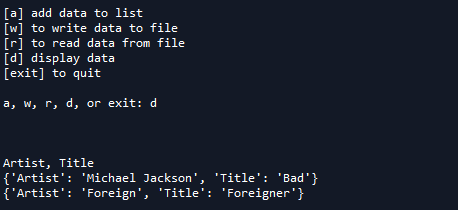


Figure - Output for Code from Figure 3: Option “d”

The first key difference in the code is line #12 which sets up the empty initial dictionary. Line #31 puts the inputs into a dictionary which then gets added into the master list.

In Line #39, the “.values()” must be used in order to pull only the individual values of the dictionary without the keys to be written into the text file.

Another difference is Line #54, which uses a different syntax to read from the .txt file than Lab 05A used. Lastly, the loop for the “d” input is much simpler, not requiring a for loop in order to get the desired output.

Figure 5 is a snippet of the .txt file to show that the “w” input produces the desired results.

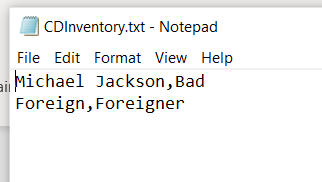
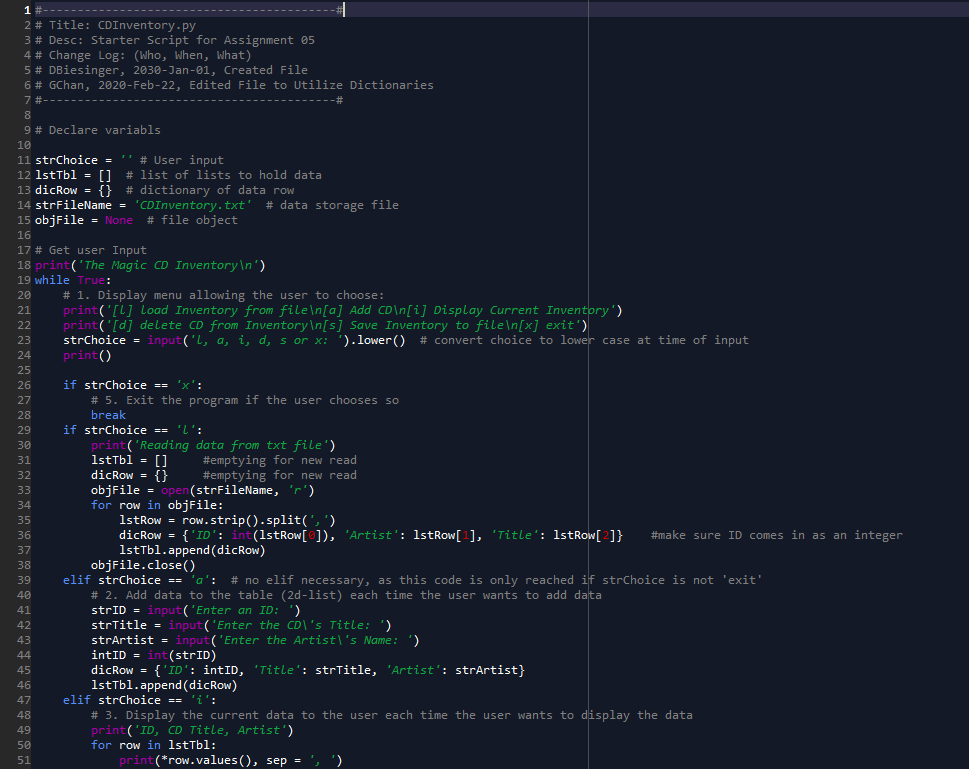


Figure - CDInventory.txt - A Product of the "w" Input

One major way to improve the code would be a style called “Separation of Concerns”. Essentially, the data, processing, and presentation would be handled in different sections of the code. Functions could also be used to streamline the code and make it more efficient in the body. These topics will be explored more in next week’s assignment, Assignment 06.

## Assignment 05: Editing Someone Else’s Code

Assignment 05 introduces a concept that is quite common in the daily workplace; the student is tasked with editing some code that has already been provided to them. The student must also add in code that allows the user to “delete” an entry in their table; syntax for the delete function can be found in the class textbook[[1]](#footnote-1). A snippet of the edited code can be found in Figure 6, and the outputs can be found on Figure 7. The code can also be found in the Appendix of this document.



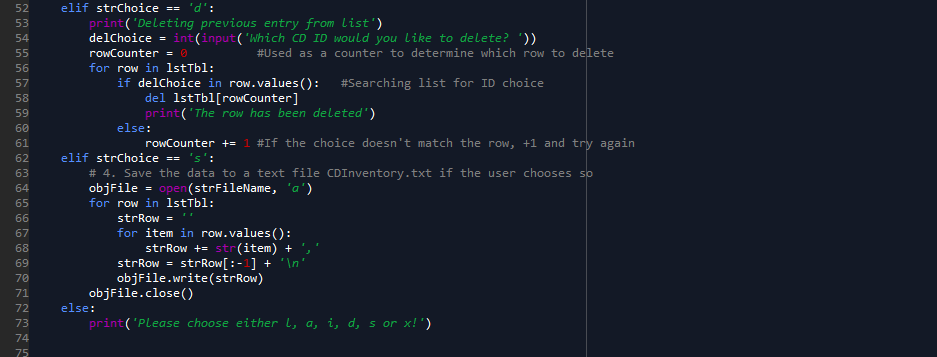
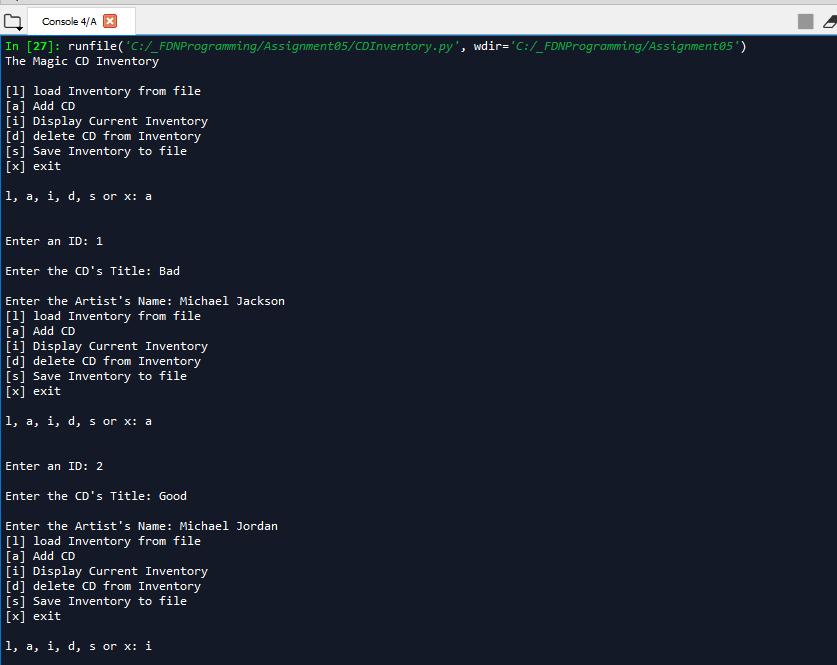


Figure - Code for Assignment 05



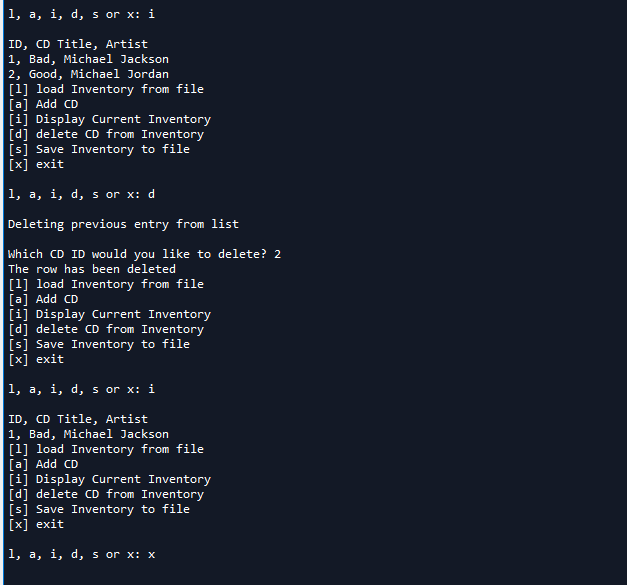


Figure - Sample Output for Figure 6's Code

The first “TODO” was to change the list of lists with list of dictionaries, which is seen in Line #13 of the code. The “l” input was the same as the code that was used in Lab 05B; the lists are emptied and the txt file is then read to retrieve the values. It is important to note that the ID needs to be an integer for later in the code. The final “todo” was to create a delete function if the user inputted “d”. I created an integer called “rowCounter” set to 0 initially. A for loop would search the row in the table for the user’s inputted ID #; if it was not found, the rowCounter would increase by 1 and the loop would run again. Once the ID was found, the “del” command is used to delete the desired row. “rowCounter” is different than the ID to cover for the cases where the user might add CD’s out of number order.

Lastly, as warned by the assignment, changing aspects of the code will sometimes cause other parts of the code to “break”. One such example was in the saving of the data. Because the data is now stored in dictionaries rather than lists, the “.values” command was required in order to pull out only the values and not the keys from the dictionaries into the notepad file.

# Saving the Python File

The python script was saved in the following directory:

C:\\_FDNProgramming\Assignment05

The code itself was named CDInventory.py. The text file with the results is named CDInventory.txt, as named by the script.

The link to the Github is as follows:

<https://github.com/NotGavin/Assignment_05/blob/master/CDInventory.py>

# Summary

In this module, we learned about the importance of working with other people’s code. Students were also required to submit their code onto Github, as well as peer-review one other student’s code. This was a fantastic learning experience, and it was very interesting to see how other student’s had both similar and different thought processes from my own.

# Appendix

Lab 05-A Script:

1. #---------------------------------#
2. # Title: Lab05\_A.py
3. # Desc: Lab05-A: Working with Files and Lists
4. # Change Log: (Who, When, What)
5. # GChan, 2020-Feb-21, Created File
6. #---------------------------------#
8. # Declare variables
10. strChoice = '' # User input
11. lstTbl = [] # list of lists to hold data
12. lstRow = [] # list of data row
13. strFileName = 'CDInventory.txt' # data storage file
14. objFile = None # file object
16. # Get user Input
17. **print**('Write or Read file data.')
18. **while** True:
19. **print**('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
20. **print**('[d] display data\n[exit] to quit')
21. strChoice = input('a, w, r, d, or exit: ').lower() # convert choice to lower case at time of input
22. **print**('\n\n')
24. **if** strChoice == 'exit':
25. **break**
26. **if** strChoice == 'a': # no elif necessary, as this code is only needed if strChoice is not 'exit'
27. # Add data to list in memory
28. # TODO ask user to input data and store it in the in-memory list
29. CDArt = input('Enter the CDs Artist: ')
30. CDTitle = input('Enter the CDs Title: ')
31. lstRow = [CDArt, CDTitle]
32. lstTbl.append(lstRow)
33. **elif** strChoice == 'w':
34. # List to File
35. # TODO add code here to write from in-memory list to file
36. **print**('Writing data to txt file')
37. strRow = ''
38. **for** row **in** lstTbl:
39. **for** col **in** row:
40. strRow += str(col) + ','
41. strRow = strRow[:-1]+ '\n'
42. objFile = open(strFileName, 'w')
43. objFile.write(strRow)
44. objFile.close()
45. **elif** strChoice == 'r':
46. # File to print
47. # TODO add code here to write from in-memory list to file
48. **print**('Reading data from txt file')
49. lstRow = []     #emptying for new read
50. lstTbl = []     #emptying for new read
51. objFile = open(strFileName, 'r')
52. **for** row **in** objFile:
53. lstRow = row.strip().split(',')
54. lstTbl.append(lstRow)
55. objFile.close()
56. **elif** strChoice == 'd':
57. # Display data
58. # TODO display the data to the user.
59. **print**('Artist, Title')
60. displayData = ''
61. **for** row **in** lstTbl:
62. **for** col **in** row:
63. displayData += str(col) + ' | '
64. displayData = displayData[:-1]+ '\n'
65. **print**(displayData)
66. **else**:
67. **print**('Please choose either a, w, r, d, or exit!')

Lab 05-B Script:

1. #---------------------------------#
2. # Title: Lab05\_B.py
3. # Desc: Lab05-B: Working with Dictionaries
4. # Change Log: (Who, When, What)
5. # GChan, 2020-Feb-21, Created File
6. #---------------------------------#
8. # Declare variables
10. strChoice = '' # User input
11. lstTbl = [] # list of lists to hold data
12. dicRow = {} # dictionary of data row
13. strFileName = 'CDInventory.txt' # data storage file
14. objFile = None # file object
16. # Get user Input
17. **print**('Write or Read file data.')
18. **while** True:
19. **print**('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
20. **print**('[d] display data\n[exit] to quit')
21. strChoice = input('a, w, r, d, or exit: ').lower() # convert choice to lower case at time of input
22. **print**('\n\n')
24. **if** strChoice == 'exit':
25. **break**
26. **if** strChoice == 'a': # no elif necessary, as this code is only needed if strChoice is not 'exit'
27. # Add data to list in memory
28. # TODO ask user to input data and store it in the in-memory list
29. CDArt = input('Enter the CDs Artist: ')
30. CDTitle = input('Enter the CDs Title: ')
31. dicRow = {'Artist': CDArt, 'Title': CDTitle}
32. lstTbl.append(dicRow)
33. **elif** strChoice == 'w':
34. # List to File
35. # TODO add code here to write from in-memory list to file
36. **print**('Writing data to txt file')
37. strRow = ''
38. **for** row **in** lstTbl:
39. **for** col **in** row.values():
40. strRow += str(col) + ','
41. strRow = strRow[:-1]+ '\n'
42. objFile = open(strFileName, 'w')
43. objFile.write(strRow)
44. objFile.close()
45. **elif** strChoice == 'r':
46. # File to print
47. # TODO add code here to write from in-memory list to file
48. **print**('Reading data from txt file')
49. lstTbl = []     #emptying for new read
50. dicRow = []     #emptying for new read
51. objFile = open(strFileName, 'r')
52. **for** row **in** objFile:
53. lstRow = row.strip().split(',')
54. dicRow = {'Artist': lstRow[0], 'Title': lstRow[1]}
55. lstTbl.append(dicRow)
56. objFile.close()
57. **elif** strChoice == 'd':
58. # Display data
59. # TODO display the data to the user.
60. **print**('Artist, Title')
61. **for** row **in** lstTbl:
62. **print**(row)
63. **else**:
64. **print**('Please choose either a, w, r, d, or exit!')

Assignment 05’s Code

1. #------------------------------------------#
2. # Title: CDInventory.py
3. # Desc: Script for Assignment 05
4. # Change Log: (Who, When, What)
5. # DBiesinger, 2030-Jan-01, Created File
6. # GChan, 2020-Feb-22, Edited File to Utilize Dictionaries
7. #------------------------------------------#
9. # Declare variabls
11. strChoice = '' # User input
12. lstTbl = []  # list of lists 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**('Reading data from txt file')
31. lstTbl = []     #emptying for new read
32. dicRow = {}     #emptying for new read
33. objFile = open(strFileName, 'r')
34. **for** row **in** objFile:
35. lstRow = row.strip().split(',')
36. dicRow = {'ID': int(lstRow[0]), 'Artist': lstRow[1], 'Title': lstRow[2]}    #make sure ID comes in as an integer
37. lstTbl.append(dicRow)
38. objFile.close()
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. strID = input('Enter an ID: ')
42. strTitle = input('Enter the CD\'s Title: ')
43. strArtist = input('Enter the Artist\'s Name: ')
44. intID = int(strID)
45. dicRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
46. lstTbl.append(dicRow)
47. **elif** strChoice == 'i':
48. # 3. Display the current data to the user each time the user wants to display the data
49. **print**('ID, CD Title, Artist')
50. **for** row **in** lstTbl:
51. **print**(\*row.values(), sep = ', ')
52. **elif** strChoice == 'd':
53. **print**('Deleting previous entry from list')
54. delChoice = int(input('Which CD ID would you like to delete? '))
55. rowCounter = 0          #Used as a counter to determine which row to delete
56. **for** row **in** lstTbl:
57. **if** delChoice **in** row.values():   #Searching list for ID choice
58. **del** lstTbl[rowCounter]
59. **print**('The row has been deleted')
60. **else**:
61. rowCounter += 1 #If the choice doesn't match the row, +1 and try again
62. **elif** strChoice == 's':
63. # 4. Save the data to a text file CDInventory.txt if the user chooses so
64. objFile = open(strFileName, 'a')
65. **for** row **in** lstTbl:
66. strRow = ''
67. **for** item **in** row.values():
68. strRow += str(item) + ','
69. strRow = strRow[:-1] + '\n'
70. objFile.write(strRow)
71. objFile.close()
72. **else**:
73. **print**('Please choose either l, a, i, d, s or x!')

1. Retrieved 2020-Feb-22. Python Programming for the Absolute Beginner, Third Edition Pg 146 [↑](#footnote-ref-1)