Kelly Ros

2020-Feb-22

Foundations of Programming, Python

Assignment 05

Lists and Dictionaries

# Introduction

In this week’s assignment, we continue with our coverage of lists, and go over dictionaries. We also utilize GitHub in order to upload our code for review by our peers.

# Lists (Continued) and Dictionaries

In this week’s module, we continued with some built-in functions of lists and using lists to write to and read from files. We learned to unpack lists using the \* operator, which is a convenient way to unpack and format contents of a list using the print function. Lists can also be used to read from and write to files. An example of this is shown in Lab 05-A, the user is prompted with 5 options: add, write, read, display, or quit. Based on the users input, a different portion of the code is run. In the “add” option, we’re creating new list of data, and appending it into the list table, to create a 2-D list. The “write” option will take the 2-D list, format and then write the data into a text file. The “read” and “display” functions are similar, however the read function reads the data from the text file and outputs it onto the screen and the display function displays the in-memory list. These functions can be seen in listing 1.

1. **if** strChoice == 'exit':
2. **break**
3. **if** strChoice == 'a': # no elif necessary, as this code is only needed if strChoice is not 'exit'
4. # Add data to list in memory
5. # TODO ask user to input data and store it in the in-memory list
6. cdTitle = input("Enter CD title: ")
7. cdArtist = input("Enter CD artist: ")
8. lstRow = [cdTitle, cdArtist]
9. lstTbl.append(lstRow)
10. **elif** strChoice == 'w':
11. # List to File
12. # TODO add code here to write from in-memory list to file
13. **print**("Writing to file!")
14. strRow = ''
15. objFile = open(strFileName, 'w')
16. **for** row **in** lstTbl:
17. **for** item **in** row:
18. strRow += str(item) + ','
19. strRow = strRow[:-1]+ '\n'
20. objFile.write(strRow)
21. objFile.close()
22. **elif** strChoice == 'r':
23. # File to print
24. # TODO add code here to write from in-memory list to file
25. **print**("Reading from file!")
26. objFile = open(strFileName, 'r')
27. **for** row **in** objFile:
28. lstRow = row.strip().split(',')
29. **print**(lstRow)
30. objFile.close()
31. **elif** strChoice == 'd':
32. # Display data
33. # TODO display the data to the user.
34. **print**("Displaying CD Inventory")
35. **for** row **in** lstTbl:
36. **for** item **in** row:
37. **print**(item, end=' | ')
38. **print**('')
39. **else**:
40. **print**('Please choose either a, w, r, d, or exit!') # added missing d option

Listing 1 – Functions from Lab05\_A.py

A lot of these lines of code are re-used from the previous module, but the only new parts are from the read portion. In the read portion, we open the file, and use a for loop to go through each “row.” The row is defined by the strip and the split functions. The strip function removes the beginning and end characters (in this case “\n”) and the split will “split” the row into individual elements of a list at the commas.[[1]](#footnote-1) A sample output of Lab 05A can be seen in figure 1.

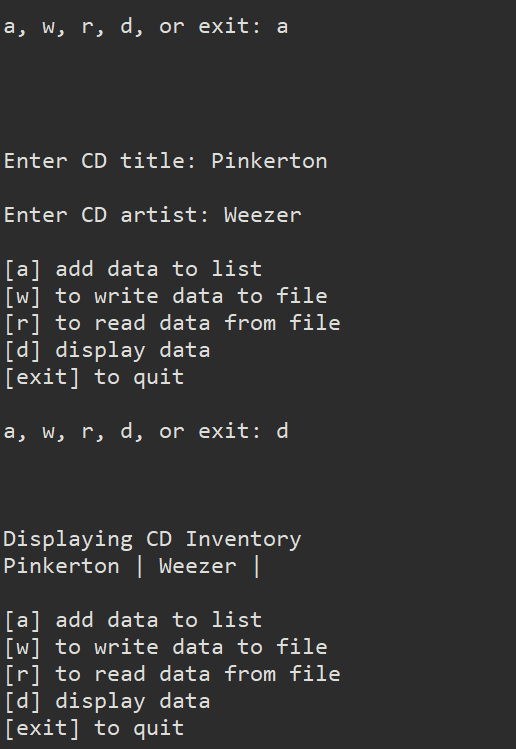


Figure 1 – Example of Lab05\_A.py adding and displaying information

The next portion of the module covers dictionaries. Dictionaries are another way to organizing information, however, unlike tuples and lists, you store information in pairs.[[2]](#footnote-2) Dictionaries are formatted as such {key: value}, you would look up the key, and the value would be the output. Lab 05-B is the same as Lab 05-A, however instead of using a list of lists, we use a list of dictionaries. The functionality parts of the program can be seen in listing 2.

1. **if** strChoice == 'exit':
2. **break**
3. **if** strChoice == 'a': # no elif necessary, as this code is only needed if strChoice is not 'exit'
4. # Add data to list in memory
5. # TODO ask user to input data and store it in the in-memory list
6. cdTitle = input("Enter CD title: ")
7. cdArtist = input("Enter CD artist: ")
8. dctRow = {'Title':cdTitle, 'Artist':cdArtist}
9. lstTbl.append(dctRow)
10. **elif** strChoice == 'w':
11. # List to File
12. # TODO add code here to write from in-memory list to file
13. **print**("Writing to file!")
14. strRow = ''
15. objFile = open(strFileName, 'w')
16. **for** row **in** lstTbl:
17. **for** val **in** row.values():
18. strRow += str(val) + ','
19. strRow = strRow[:-1]
20. strRow = strRow + '\n'
21. objFile.write(strRow)
22. objFile.close()
23. **elif** strChoice == 'r':
24. # File to print
25. # TODO add code here to write from in-memory list to file
26. **print**("Reading from file!")
27. objFile = open(strFileName, 'r')
28. **for** row **in** objFile:
29. lstRow = row.strip().split(',')
30. dctRow = {'Title':lstRow[0], 'Artist':lstRow[1]}
31. **print**(dctRow)
32. objFile.close()
33. **elif** strChoice == 'd':
34. # Display data
35. # TODO display the data to the user.
36. **print**("Displaying CD Inventory")
37. **for** row **in** lstTbl:
38. **for** key, val **in** row.items():
39. **print**(key, ':', val, end=' | ')
40. **print**('')
41. **else**:
42. **print**('Please choose either a, w, r, d, or exit!') # added missing d option

Listing 2 – Lab05\_B.py using dictionaries

In this lab, we do exactly the same as the first lab, but using dictionaries. Some notable differences are in lines 8, 30, 38, and 39. In lines 8 and 30, we are just defining a dictionary using the correct syntax. However, the for loop in line 38 and 39 is different. In this for loop, we’re looking through the rows of the 2-D list, which are individual dictionaries. We use the items() function to return a two-item tuple[[3]](#footnote-3) and the variables key and val are the items from the tuple. A sample can be seen in figure 2.

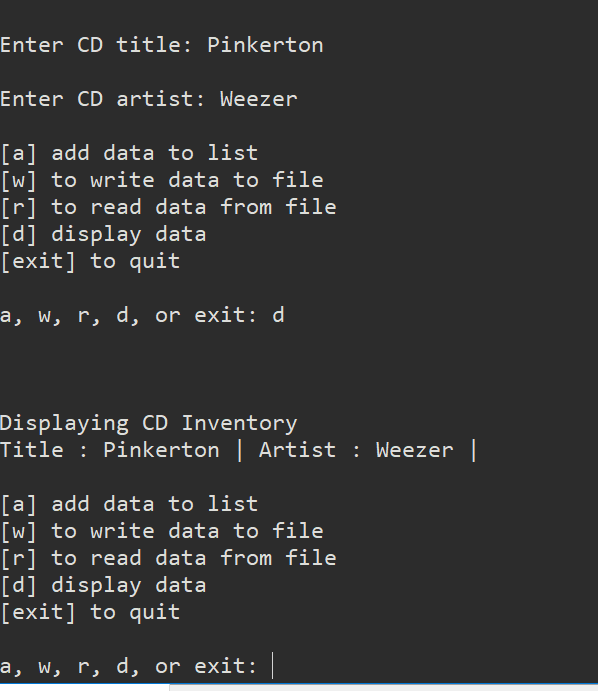


Figure 2 – Lab05\_B.py sample add and display functionality

# CD Inventory Program

For the assignment this week, we are working on a code that is provided to us. The program is similar to the labs, where we’ll be using 2-D lists with a dictionary, prompting the user for an input, adding, displaying, and saving lists. Some new functionality needs to be added into the code, where the user can also load data from a file into our 2-D list and deleting an entry from the list. First, the program utilized a list of lists, and we needed to convert the inner lists into dictionaries. Once we changed the nested lists into dictionaries, we had to edit parts that utilized the original lists which is similar to Lab 05-B of this week’s module.

The loading data functionality reads a file called “CDInventory.txt” and goes through each line of the file, converts it into a dictionary, and then appends it to the 2-D list. This is similar to the reading files portions of Lab05-A and Lab05-B, except we have to populate the keys and values of the lines from the text file. The loading data portion of the code can be seen in listing 3.

1. **if** strChoice == 'l':
2. **print**('Loading from', strFileName)
3. objFile = open(strFileName, 'r')
4. **for** row **in** objFile:
5. lstRow = row.strip().split(',')
6. dictRow = {'ID':lstRow[0], 'Title':lstRow[1], 'Artist':lstRow[2]}
7. lstTbl.append(dictRow)
8. objFile.close()

Listing 3 – Loading data from file

In this excerpt, we open the file, and using the strip and split functions as we did before, we create a list. We added in the keys and values using the list that was created and append the dictionary to the 2-D list. After the for loop is done going through each row, we close the file. We load the file and display the information in figure 3.

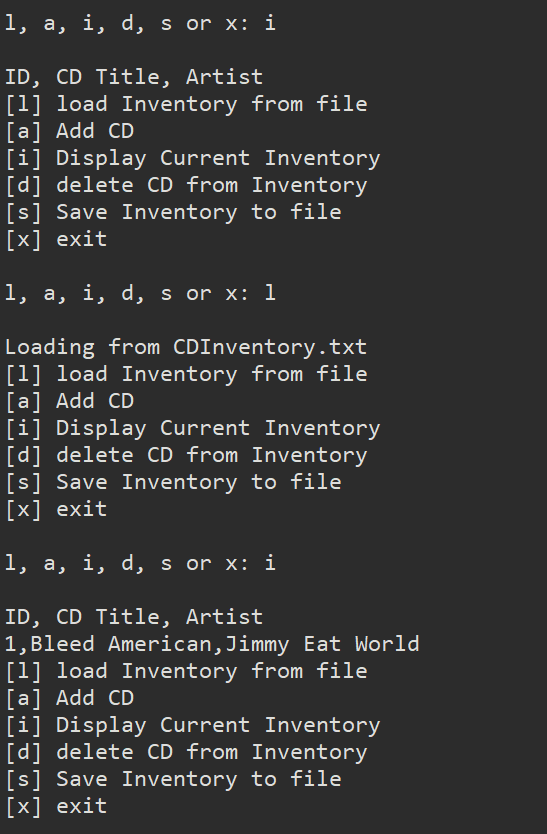


Figure 3 – loading information from a file and displaying it

The deleting option allows the user to remove a line from the table using the ID number. For this portion, my idea was to search through each row using a for loop, and then searching each dictionary using the values(), looking for the ID number. If the ID number existed in that row, the index() function that was used would return the index value of the row and delete it from the 2-D list. The code for this is shown in listing 4.

1. **elif** strChoice == 'd':
2. idDel = input("Which ID would you like to delete?: ")
3. **for** row **in** lstTbl:
4. **if** idDel **in** row.values():
5. **del** lstTbl[lstTbl.index(row)] # deletes the index that contains user input

Listing 4 – Deleting function

The user is asked to input an ID number and the for loop will search through each row. Instead of using another for loop to search the nested dictionaries, the values() function was used in order to check if the ID that the user entered in was in the dictionary. If the ID number existed, the if statement would delete the row located in the index. The index() function returns the index number of the first occurrence of the value in lstTbl.[[4]](#footnote-4) We can see an example of this working in figure 4. We can also see the other functions of the program in figures 5, 6, and 7

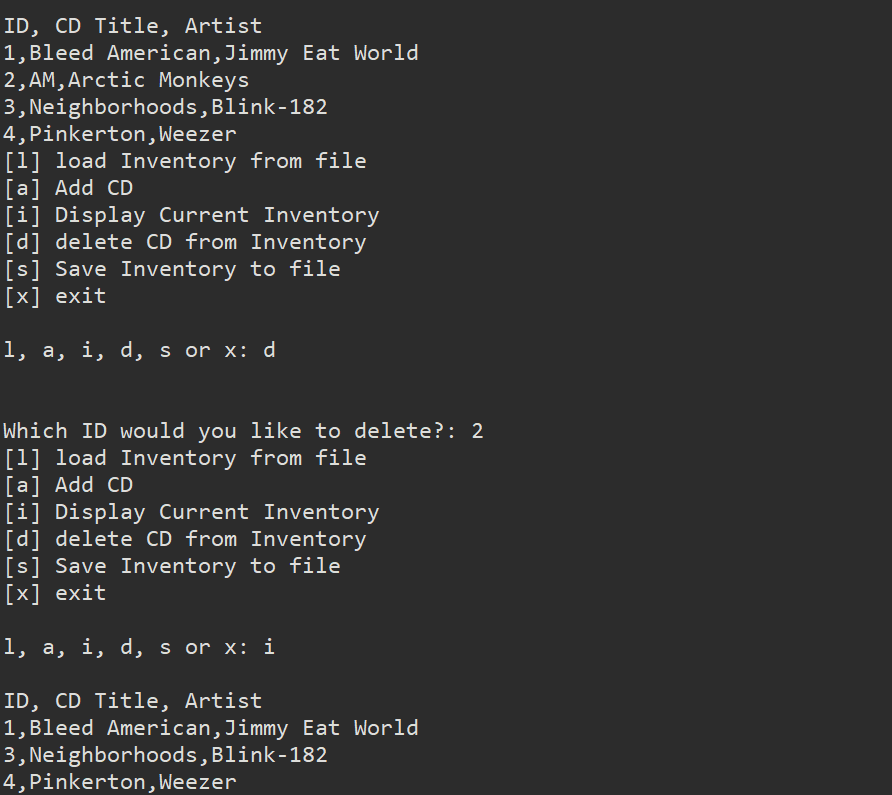


Figure 4 – Deleting a row from 2-D list

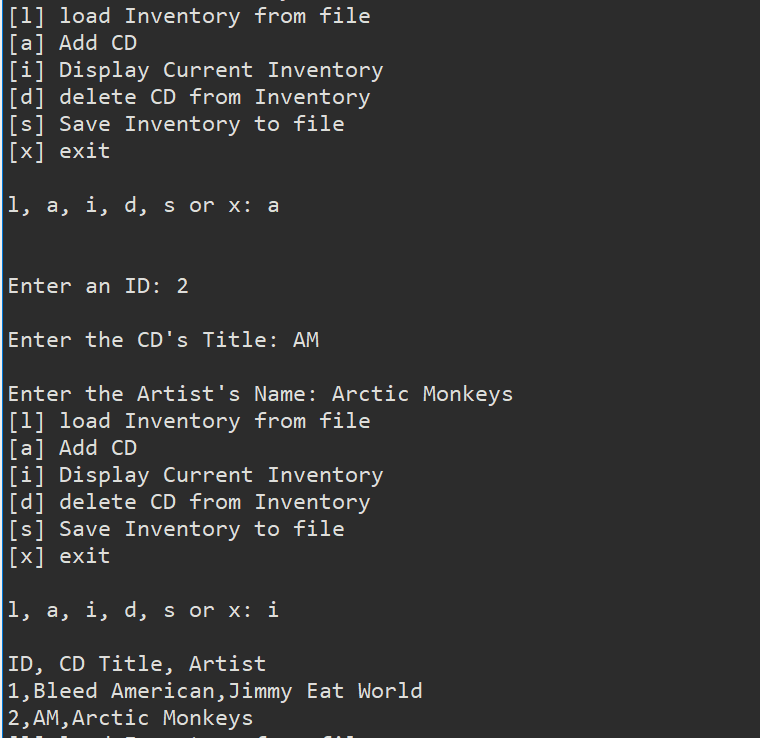


Figure 5 – Adding a new entry to the 2-D list

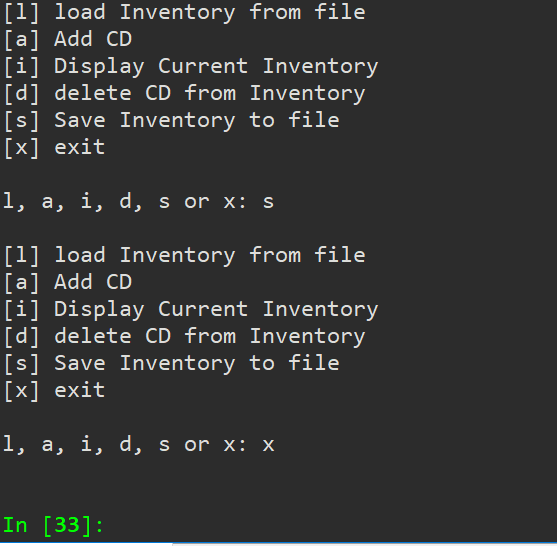


Figure 6 – Saving 2-D list to CDInventory.txt

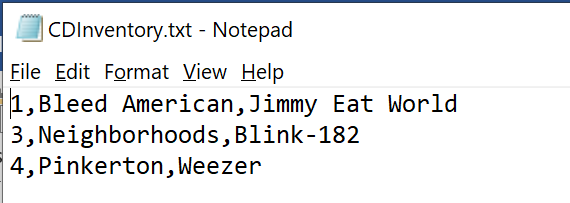


Figure 7 – CDInventory.txt that was saved. ID number 2 was deleted

# Summary

In this assignment, we continued with lists and went over a new collection of data, known as dictionaries. Dictionaries are similar to tuples and lists, but store information in pairs known as keys and values. In this module, we were provided with a program that we needed to make edits. It can be both difficult and easy to work with someone else’s code, depending on how readable and how commented it is. It’s impossible to tell what the original creator of the program was thinking when they initially wrote the program, so we rely on comments that they may have written in. We also briefly went over code structures and functions, but will cover those in the next module. We also covered some topics of GitHub, where users can upload projects or programs onto the internet for other users to see.

# Appendix

## Listing Lab05\_A.py

1. #---------------------------------#
2. # Title: Lab05\_A.py
3. # Desc: Lab05-A script
4. # Change Log: (Who, When, What)
5. # DBiesinger, 2030-Jan-01, Created File
6. # KRos, 2020-Feb-17, Copied in starter code
7. # KRos, 2020-Feb-18, Added code under add data, write, and display block
8. # KRos, 2020-Feb-19, changed write block to use 'w' and added read block
9. #---------------------------------#
11. # Declare variables
13. strChoice = '' # User input
14. lstTbl = [] # list of lists to hold data
15. lstRow = [] # list of data row
16. strFileName = 'CDInventory.txt' # data storage file
17. objFile = None # file object
19. # Get user Input
20. **print**('Write or Read file data.')
21. **while** True:
22. **print**('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
23. **print**('[d] display data\n[exit] to quit')
24. strChoice = input('a, w, r, d, or exit: ').lower() # convert choice to lower case at time of input
25. **print**('\n\n')
27. **if** strChoice == 'exit':
28. **break**
29. **if** strChoice == 'a': # no elif necessary, as this code is only needed if strChoice is not 'exit'
30. # Add data to list in memory
31. # TODO ask user to input data and store it in the in-memory list
32. cdTitle = input("Enter CD title: ")
33. cdArtist = input("Enter CD artist: ")
34. lstRow = [cdTitle, cdArtist]
35. lstTbl.append(lstRow)
36. **elif** strChoice == 'w':
37. # List to File
38. # TODO add code here to write from in-memory list to file
39. **print**("Writing to file!")
40. strRow = ''
41. objFile = open(strFileName, 'w')
42. **for** row **in** lstTbl:
43. **for** item **in** row:
44. strRow += str(item) + ','
45. strRow = strRow[:-1]+ '\n'
46. objFile.write(strRow)
47. objFile.close()
48. **elif** strChoice == 'r':
49. # File to print
50. # TODO add code here to write from in-memory list to file
51. **print**("Reading from file!")
52. objFile = open(strFileName, 'r')
53. **for** row **in** objFile:
54. lstRow = row.strip().split(',')
55. **print**(lstRow)
56. objFile.close()
57. **elif** strChoice == 'd':
58. # Display data
59. # TODO display the data to the user.
60. **print**("Displaying CD Inventory")
61. **for** row **in** lstTbl:
62. **for** item **in** row:
63. **print**(item, end=' | ')
64. **print**('')
65. **else**:
66. **print**('Please choose either a, w, r, d, or exit!') # added missing d option

## Listing Lab05\_B.py

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

## Listing 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. # KRos, 2020-Feb-21, Replaced lists with dictionary
7. # KRos, 2020-Feb-21, added loading and deleting functionality
8. # KRos, 2020-Feb-22, fixed the display functionality
9. #------------------------------------------#
11. # Declare variables
13. strChoice = '' # User input
14. lstTbl = []  # list of lists to hold data
15. # TODO replace list of lists with list of dicts
16. lstRow = []
17. dictRow = {}  # dictionary of data row
18. strFileName = 'CDInventory.txt'  # data storage file
19. objFile = None  # file object
21. # Get user Input
22. **print**('The Magic CD Inventory\n')
23. **while** True:
24. # 1. Display menu allowing the user to choose:
25. **print**('[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
26. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
27. strChoice = input('l, a, i, d, s or x: ').lower() #convert choice to lower case at time of input
28. **print**()
30. **if** strChoice == 'x':
31. # 5. Exit the program if the user chooses so
32. **break**
33. **if** strChoice == 'l':
34. **print**('Loading from', strFileName)
35. objFile = open(strFileName, 'r')
36. **for** row **in** objFile:
37. lstRow = row.strip().split(',')
38. dictRow = {'ID':lstRow[0], 'Title':lstRow[1], 'Artist':lstRow[2]}
39. lstTbl.append(dictRow)
40. objFile.close()
41. **elif** strChoice == 'a':
42. # 2. Add dictionary to the 2-D list each time the user wants to add data
43. strID = input('Enter an ID: ')
44. strTitle = input('Enter the CD\'s Title: ')
45. strArtist = input('Enter the Artist\'s Name: ')
46. intID = int(strID)
47. dictRow = {'ID':strID, 'Title':strTitle, 'Artist':strArtist}
48. lstTbl.append(dictRow)
49. **elif** strChoice == 'i':
50. # 3. Display the current data to the user each time the user wants to display the data
51. **print**('ID, CD Title, Artist')
52. **for** row **in** lstTbl:
53. strRow = ''
54. **for** val **in** row.values(): # cycles through dictionary and gets values
55. strRow += str(val) + ','
56. strRow = strRow[:-1] # formatting
57. **print**(strRow)
58. **elif** strChoice == 'd':
59. idDel = input("Which ID would you like to delete?: ")
60. **for** row **in** lstTbl:
61. **if** idDel **in** row.values():
62. **del** lstTbl[lstTbl.index(row)] # deletes the index that contains user input
63. **elif** strChoice == 's':
64. # 4. Save the data to a text file CDInventory.txt if the user chooses so
65. objFile = open(strFileName, 'a')
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!')

1. From <https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>. Retrieved on 2020-Feb-22 [↑](#footnote-ref-1)
2. From Python Programming for the Absolute Beginner, Third Edition. Page 140. Retrieved on 2020-Feb-19 [↑](#footnote-ref-2)
3. From Python Programming for the Absolute Beginner, Third Edition. Page 148. Retrieved on 2020-Feb-19 [↑](#footnote-ref-3)
4. From <https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>. Retrieved on 2020-Feb-22 [↑](#footnote-ref-4)