Kelly Kauffman

August 11, 2020

IT FDN 110: Introduction to Programming: Python

Assignment 05

Module 05: Dictionaries and Sharing Code

# Introduction

The focus of Module 05 was dictionaries. I got an understanding of what a dictionary is and how to use it to handle data in a program. I also got an introduction to Git Hub, so I can share my code and review my peer’s code. The ability to see my peer’s code was beneficial for me when solving the difficulties I faced while completing Assignment 05. This assignment was to take the starter code, CDInventory\_Starter.py, and modify it. I needed to add the ability for the user to delete a CD entry, to load/view the entries saved to the .txt external file, and to save the data as a list of dictionaries.

# Topic 1: Module 05 Learning and Labs

The learning and labs for this module went a lot smoother than the last module. Since dictionaries are similar to lists, I felt this module’s content was more of a review. The labs did not give me a lot of trouble either. After I figured out there was not a lab starter, I decided to do the labs with the CDInventory\_Starter.py file. This helped me in the long run because I ended up working on the assignment requirements at the same time. At first I was a little overwhelmed looking at all the starter code, but I decided to just focus on one To Do at a time. This helped me feel more confident.

The first To Do I worked on was adding the functionality of deleting a CD entry. Figure 1 shows my first attempt. In the Appendix is the code for this first draft of Lab A. This first attempt resulted in an error that said “TypeError: ‘builtin\_function\_or\_method’ object is not subscriptable”. I knew lstTbl.pop was the correct syntax, so something must be wrong with the end of that line of code. Figure 2 shows my second attempt at the syntax for deleting a row. This second time through I noticed that the artist name from each row was getting deleted. I was getting closer to the correct syntax, but not there yet.

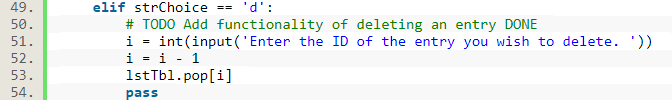


Figure 1 Lab A: Deleting a row first try

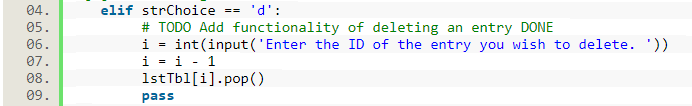


Figure 2 Lab A: Deleting a row second try

Figure 3 shows my third attempt at the syntax. Finally, this was the correct syntax. I noticed that I could not ask the user to enter the ID of the entry because the entries will change position when one is deleted. I had to change the wording in the print ( ) statement to “Enter the position/row of the entry you wish to delete. “. This way of deleting entries does cause the user to have to remember what position each entry is. Later in this module, I wanted to improve this deletion functionality and make it easier for the user.

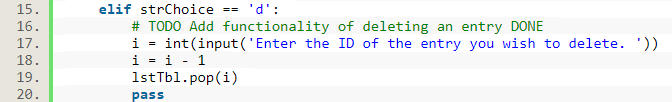


Figure 3 Lab A: Deleting a row third try

In Lab B, the next functionality I tackled was printing out each dictionary in the outer list. I used:

for row in lstTbl

print (row)

This printed both keys and values in each dictionary. I knew I wanted to only print the values but did not immediately know what the syntax would be to do this. I decided to check out the discussion board to see what problems my peers were struggling with and if anyone had the same difficulty as me. Someone did have the same problem and another peer posted the solution, which I used in my code. The following is the solution:

for row in lstTbl

print (\*row.values( ), sep= ‘, ‘)

# Topic 2: Displaying Data from a File

For Assignment 05, the first functionality I worked on was loading the data stored on the external .txt file. The additional video[[1]](#footnote-1) in the Assignment 05 PDF was very helpful in helping me get started writing the code for this functionality. Figure 4 shows my first attempt. Entry #2 in the Appendix contains the full code for my first draft of Assignment 05. This first attempt just displayed the keys from each dictionary. I checked my external .txt file and found that the keys were the only things that were saved.

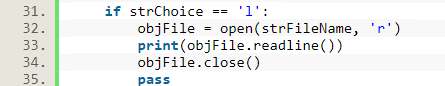


Figure 4 First attempt at loading saved data

I was not sure how to save just the values in the external files, so I turned to GitHub to see some examples of what my peers did to save their data. Once again, row.values( ) was the answer to my problem. I finally understood that I can add a method to the variable, row, in order to use just the parts of the dictionary I need. I updated the portion of my code that saved the CD inventory to a file to what is shown in Figure 5 and kept the code in Figure 4 for loading the data from the file. When I ran the code, the data saved correctly, but still displayed just the keys.

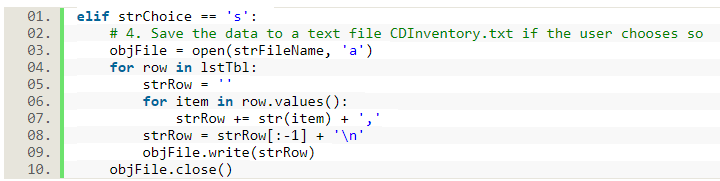


Figure 5 Updating the code for saving data to the external file

After I looked at the pythonforbeginners.com[[2]](#footnote-2) webpage about reading and writing files in python, I realized I needed to change readline( ) to readlines ( ) in line 33 of the code in Figure 4. Now, I was able to load and display all the data from the file. The only problem was that “\n” was showing at the end of each printed row. I tried a couple different ways to get rid of this extra “\n”. These attempted solutions are shown in Figure 6. Neither of these solutions worked. The first attempt gave me an error that said, “unsupported operand type(s) for -: ‘list’ and ‘str’”. The second attempt gave me an error that said, “unsupported operand types(s) for -: ‘str’ and ‘str’”. I learned I cannot just subtract a “\n”. I needed to use a different method. I decided to reach out to our TA, Doug, to see if he could help.

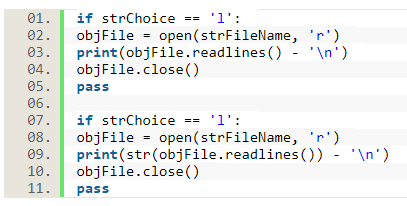


Figure 6 Attempts at removing "\n" when data is loaded from a file

Doug gave me a hint to save the file into a variable; and then from there loop through each line in that variable, strip off the “\n”, and separate each value with a comma. In addition to this hint, I also used the code from the “delete an entry” section of the code (shown in lines 63-70 in entry #3 in the Appendix). Figure 7 shows the revised code to load data from a file. Running this code removed the “\n” from each row but caused the last letter of each row to be deleted. Since only one letter was being removed at the end of each row, I figured the [:-1] in line 38 in Figure 7 was causing this. I removed all of line 38, changed the print ( k ) to print (k, sep= ‘,’), ran the code again, and got the result I wanted. Each row from the .txt file is printing with each value separated by a comma.

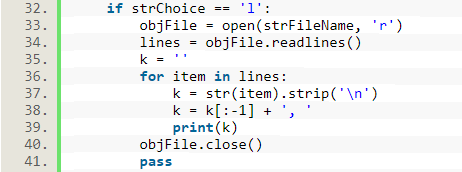


Figure 7 Revised code to load data from a file

# Topic 3: Deleting Rows from a Dictionary

At this point, my code technically met all the requirements of Assignment 05, but I still wanted to improve the functionality of deleting CD inventory entries. I wrote some code that would delete entries based on the ID of the entry. This code is shown in Figure 8. It works well, but our TA, Doug, brought up the possibility that there may be duplicate ID numbers and asked me to think about how I could deal with this situation. For the method I used in Lab A to delete an entry (lstTbl.pop( i )), Doug suggested I could display the row number of each entry, so the user does not have to remember the position of all the entries. Out of these two methods, deleting by ID and deleting by row position, I thought deleting by row position and adding the row numbers would be the easiest. I was not sure how to approach dealing with a duplicate ID.

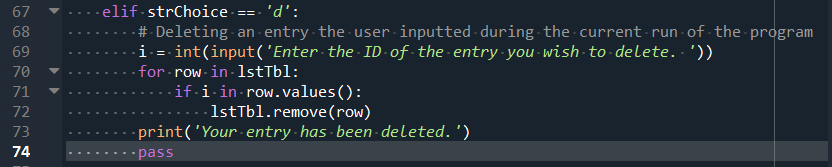


Figure 8 Deleting an entry based on its ID number

The only difficulty I experienced when adding row numbers to the data before asking the user to choose a row to delete was defining the variable, row\_num, inside the for loop instead of right before the loop. Figure 9 shows my final code for the “deleting an entry” part of the CD Inventory program. When row\_num was defined inside the for loop it caused all the row numbers to show up as “1”. I realized that every time the for loop ran, row\_num was re-defined as “0”. I needed the variable to increase by “1” each time without resetting. To fix this, I just needed to move the line defining row\_num as “0” to before the for loop.

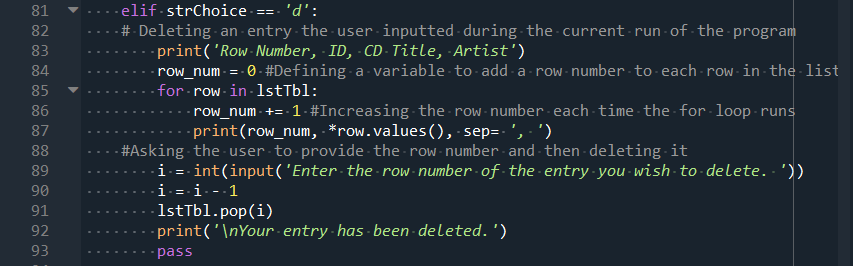


Figure 9 Final code for the "deleting an entry" part of the CD Inventory program

# Topic 4: Adding Finishing Touches

As always, the last steps I took in completing Assignment 05 was editing the formatting to organize the code better and make the program more pleasing for the user to look at. Some of the edits I made included breaking the code up into separate concerns (Data, Processing, and Presentation), adding more print ( ) statements to describe to the user what the program is doing, and spacing out the statements that are printed in the command prompt. In Module 04 I added a header to the external .txt file, but this header was added to the file every time I ran the program and saved the user’s inputs. Doug shared a method that would only write the header to the file if it did not previously exist. I included this method in my Assignment 05 code.

The final code for Assignment 05 is featured in entry #4 in the Appendix. Figure 10, 11, and 12 show my program running in Spyder. Figure 13 shows my program running in the Anaconda prompt.

# A screenshot of a cell phone Description automatically generated

Figure 10 Assignment 05 running on Spyder part 1

A screenshot of a cell phone

Description automatically generated

Figure 11 Assignment 05 running on Spyder part 2

A screenshot of a computer screen

Description automatically generated

Figure 12 Assignment 05 running on Spyder part 3

A screenshot of a cell phone

Description automatically generated

Figure 13 Assignment 05 running in the Anaconda prompt

# Summary

Module 05 taught me about dictionaries and how to modify them; separation of concerns; and GitHub. Functions and error handling were also introduced. Assignment 05 made use of this new knowledge of dictionaries. For Assignment 05 I was asked take the starter code, CDInventory\_Starter.py, and add the ability for the user to delete an entry, allow the user to view the contents of the external .txt file, and use a 2D list of dictionaries instead of a 2D list of lists. I found it easier to complete this assignment because I utilized the knowledge of my peers (via the discussion board and GitHub) and reached out to our class TA instead of trying to figure everything out on my own. I look forward to more opportunities to collaborate and learn from my peers in the upcoming modules.

# Appendix

1. Lab A First Draft
2. #------------------------------------------#
3. # Title: CDInventory.py
4. # Desc: Starter Script for Assignment 05
5. # Change Log: (Who, When, What)
6. # DBiesinger, 2030-Jan-01, Created File
7. #Kkauffman, 2020-Aug-09, Added code to TODOs
8. #------------------------------------------#
10. # Declare variabls
12. strChoice = '' # User input
13. lstTbl = []  # list of lists to hold data
14. # TODO replace list of lists with list of dicts DONE
15. lstRow = []  # dict of data row #Change to {}
16. strFileName = 'CDInventory.txt'  # data storage file
17. objFile = None  # file object
19. # Get user Input
20. **print**('The Magic CD Inventory\n')
21. **while** True:
22. # 1. Display menu allowing the user to choose:
23. **print**('[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
24. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
25. strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
26. **print**()
28. **if** strChoice == 'x':
29. # 5. Exit the program if the user chooses so
30. **break**
31. **if** strChoice == 'l':
32. # TODO Add the functionality of loading existing data (Am I loading from a file?)
33. #For now just printing
34. **for** row **in** lstTbl:
35. **print**(\*row, sep= '|')
36. **pass**
37. **elif** strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
38. # 2. Add data to the table (2d-list) each time the user wants to add data
39. strID = input('Enter an ID: ')
40. strTitle = input('Enter the CD\'s Title: ')
41. strArtist = input('Enter the Artist\'s Name: ')
42. intID = int(strID)
43. lstRow = [intID, strTitle, strArtist] #Can I have one key and two values? Nope
44. lstTbl.append(lstRow)
45. **elif** strChoice == 'i':
46. # 3. Display the current data to the user each time the user wants to display the data
47. **print**('ID, CD Title, Artist')
48. **for** row **in** lstTbl:
49. **print**(\*row, sep = ', ')
50. **elif** strChoice == 'd':
51. # TODO Add functionality of deleting an entry DONE
52. i = int(input('Enter the ID of the entry you wish to delete. '))
53. i = i - 1
54. lstTbl.pop[i]
55. **pass**
56. **elif** strChoice == 's':
57. # 4. Save the data to a text file CDInventory.txt if the user chooses so
58. objFile = open(strFileName, 'a')
59. **for** row **in** lstTbl:
60. strRow = ''
61. **for** item **in** row:
62. strRow += str(item) + ','
63. strRow = strRow[:-1] + '\n'
64. objFile.write(strRow)
65. objFile.close()
66. **else**:
67. **print**('Please choose either l, a, i, d, s or x!')

2. Assignment 05 First Draft

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. #Kkauffman, 2020-Aug-09, Added code to some TODOs
7. #KKauffman, 2020-Aug-10, Added reading file functionality
8. #------------------------------------------#
10. # Declare variabls
12. strChoice = '' # User input
13. lstTbl = []  # list of lists to hold data
14. # TODO replace list of lists with list of dicts DONE
15. lstRow = {}  # dict of data row #Change to {}
16. strFileName = 'CDInventoryp.txt'  # data storage file
17. objFile = None  # file object
19. # Get user Input
20. **print**('The Magic CD Inventory\n')
21. **while** True:
22. # 1. Display menu allowing the user to choose:
23. **print**('[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
24. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
25. strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
26. **print**()
28. **if** strChoice == 'x':
29. # 5. Exit the program if the user chooses so
30. **break**
31. **if** strChoice == 'l':
32. objFile = open(strFileName, 'r')
33. **print**(objFile.readline())
34. objFile.close()
35. **pass**
36. **elif** strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
37. # 2. Add data to the table (2d-list) each time the user wants to add data
38. strID = input('Enter an ID: ')
39. strTitle = input('Enter the CD\'s Title: ')
40. strArtist = input('Enter the Artist\'s Name: ')
41. intID = int(strID)
42. lstRow = {'ID':intID, 'Title':strTitle, 'Artist':strArtist}
43. lstTbl.append(lstRow)
44. **elif** strChoice == 'i':
45. # 3. Display the current data to the user each time the user wants to display the data
46. **print**('ID, CD Title, Artist')
47. **for** row **in** lstTbl:
48. **print**(row)
49. **print**(\*row.values(), sep= ', ')
50. **elif** strChoice == 'd':
51. # TODO Add functionality of deleting an entry DONE
52. i = int(input('Enter the position of the entry you wish to delete. '))
53. i = i - 1
54. lstTbl.pop(i)
55. **pass**
56. **elif** strChoice == 's':
57. # 4. Save the data to a text file CDInventory.txt if the user chooses so
58. objFile = open(strFileName, 'a')
59. **for** row **in** lstTbl:
60. strRow = ''
61. **for** item **in** row:
62. strRow += str(item) + ','
63. strRow = strRow[:-1] + '\n'
64. objFile.write(strRow)
65. objFile.close()
66. **else**:
67. **print**('Please choose either l, a, i, d, s or x!')

3. Assignment 05 Draft showing an attempt to load the data using code similar to the “save to a file” part of the code.

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. #Kkauffman, 2020-Aug-09, Added code to some TODOs
7. #KKauffman, 2020-Aug-10, Added reading file functionality
8. #KKauffman, 2020-Aug-11, Improved formatting
9. #------------------------------------------#
11. # Declare variabls
13. strChoice = '' # User input
14. lstTbl = []  # list of lists to hold data
15. # TODO replace list of lists with list of dicts DONE
16. lstRow = {}  # dict of data row #Change to {}
17. strFileName = 'CDInventoryp.txt'  # data storage file
18. objFile = None  # file object
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**('\n[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
25. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
26. strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
27. **print**()
29. **if** strChoice == 'x':
30. # 5. Exit the program if the user chooses so
31. **break**
32. **if** strChoice == 'l':
33. objFile = open(strFileName, 'r')
34. lines = objFile.readlines()
35. k = ''
36. **for** item **in** lines:
37. k = str(item).strip('\n')
38. k = k[:-1] + ', '
39. **print**(k)
40. objFile.close()
41. **pass**
42. **elif** strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
43. # 2. Add data to the table (2d-list) each time the user wants to add data
44. strID = input('Enter an ID: ')
45. strTitle = input('Enter the CD\'s Title: ')
46. strArtist = input('Enter the Artist\'s Name: ')
47. intID = int(strID)
48. lstRow = {'ID':intID, 'Title':strTitle, 'Artist':strArtist}
49. lstTbl.append(lstRow)
50. **elif** strChoice == 'i':
51. # 3. Display the current data to the user each time the user wants to display the data
52. **print**('ID, CD Title, Artist')
53. **for** row **in** lstTbl:
54. **print**(\*row.values(), sep= ', ')
55. **elif** strChoice == 'd':
56. # TODO Add functionality of deleting an entry DONE
57. i = int(input('Enter the position of the entry you wish to delete. '))
58. i = i - 1
59. lstTbl.pop(i)
60. **pass**
61. **elif** strChoice == 's':
62. # 4. Save the data to a text file CDInventory.txt if the user chooses so
63. objFile = open(strFileName, 'a')
64. **for** row **in** lstTbl:
65. strRow = ''
66. **for** item **in** row.values():
67. strRow += str(item) + ','
68. strRow = strRow[:-1] + '\n'
69. objFile.write(strRow)
70. objFile.close()
71. **else**:
72. **print**('Please choose either l, a, i, d, s or x!')

4. Final Draft of Assignment 05

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. #Kkauffman, 2020-Aug-09, Added code to some TODOs
7. #KKauffman, 2020-Aug-10, Added reading file functionality
8. #KKauffman, 2020-Aug-11, Improved formatting, improved menu choice 'l' and 'd'
9. #KKauffman, 2020-Aug-12, Improved menu choice 'd', added header for .txt file
10. #------------------------------------------#

13. # -- DATA -- #
15. # Declare variables
17. strChoice = '' # User input
18. lstTbl = []  # list of lists to hold data
19. lstRow = {}  # dict of data row
20. strFileName = 'CDInventory.txt'  # data storage file
21. objFile = None  # file object
23. # -- PRESENTATION --#
25. #Using this path library to write a header for the .txt file
26. **from** os **import** path
28. #Create heading for the file
29. cd\_header = 'ID,CD Title,Artist Name'
31. #Open file and save heading
32. **if** **not** path.exists(strFileName):
33. objFile = open(strFileName, 'w')
34. objFile.write(cd\_header + '\n')
35. objFile.close()
37. # Get user Input
38. **print**('The Magic CD Inventory\n')
39. **while** True:
40. # Display menu allowing the user to choose:
41. **print**('\n[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
42. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
43. strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
44. **print**()
46. **if** strChoice == 'x':
47. # Exit the program if the user chooses so
48. **break**
50. **if** strChoice == 'l':
51. #Display the data from the external .txt file
52. objFile = open(strFileName, 'r')
53. lines = objFile.readlines()
54. k = ''
55. **for** item **in** lines:
56. k = str(item).strip('\n') #Take out '\n' from when it was saved
57. **print**(k, sep= ',') #Separate items with a comma
58. objFile.close()
59. **pass**
61. # -- PROCESSING -- #
63. **elif** strChoice == 'a':
64. # Add data to the table (2d-list) each time the user wants to add data
65. strID = input('Enter an ID: ')
66. strTitle = input('Enter the CD\'s Title: ')
67. strArtist = input('Enter the Artist\'s Name: ')
68. intID = int(strID)
69. lstRow = {'ID':intID, 'Title':strTitle, 'Artist':strArtist}
70. lstTbl.append(lstRow)
71. **print**('\nYour data has been added to the CD inventory.')
73. # -- PRESENTATION -- #
75. **elif** strChoice == 'i':
76. # Display the current data to the user each time the user wants to display the data
77. **print**('ID, CD Title, Artist')
78. **for** row **in** lstTbl:
79. **print**(\*row.values(), sep= ', ')
81. **elif** strChoice == 'd':
82. # Deleting an entry the user inputted during the current run of the program
83. **print**('Row Number, ID, CD Title, Artist')
84. row\_num = 0 #Defining a variable to add a row number to each row in the list
85. **for** row **in** lstTbl:
86. row\_num += 1 #Increasing the row number each time the for loop runs
87. **print**(row\_num, \*row.values(), sep= ', ')
88. #Asking the user to provide the row number and then deleting it
89. i = int(input('Enter the row number of the entry you wish to delete. '))
90. i = i - 1
91. lstTbl.pop(i)
92. **print**('\nYour entry has been deleted.')
93. **pass**
95. # -- PROCESSING -- #
97. **elif** strChoice == 's':
98. # Save the data to a text file CDInventory.txt if the user chooses so
99. objFile = open(strFileName, 'a')
100. **for** row **in** lstTbl:
101. strRow = ''
102. **for** item **in** row.values():
103. strRow += str(item) + ','
104. strRow = strRow[:-1] + '\n'
105. objFile.write(strRow)
106. objFile.close()
107. **print**('\nYour data has been saved.')
109. # -- Presentation -- #
111. **else**:
112. **print**('Please choose either l, a, i, d, s or x!')

1. [https://youtu.be/m0o0CkYsDzI Accessed August 10](https://youtu.be/m0o0CkYsDzI%20Accessed%20August%2010), 2020 [↑](#footnote-ref-1)
2. <https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python> Accessed on August 11, 2020 [↑](#footnote-ref-2)