Justin Roe

08/12/2020

IT FDN 110 B

Assignment 04

Assignment 05

# Introduction

This week, the “CDInventory” file was modified to utilize dictionaries and lists. Reading data from a file, saving data into a file, and modifying data in 2D lists saved to memory was discussed / utilized. Useful tips for code organization and error handling, as well as Github use were also outlined. I primarily struggled with the overall structure – namely the use of the while loop and its nested dictionaries / lists (really, though, all my issues boiled down to concatenation!). I also struggled initially with the error handling that provides feedback to the user when they enter an incorrect input. The delete functionality was also an issue for me, but the structure discussed in class quickly solved my issue (defining an “entry” in the list table).

# Knowledge Application / Details

See Figures 1-5 for screenshots of the code execution.

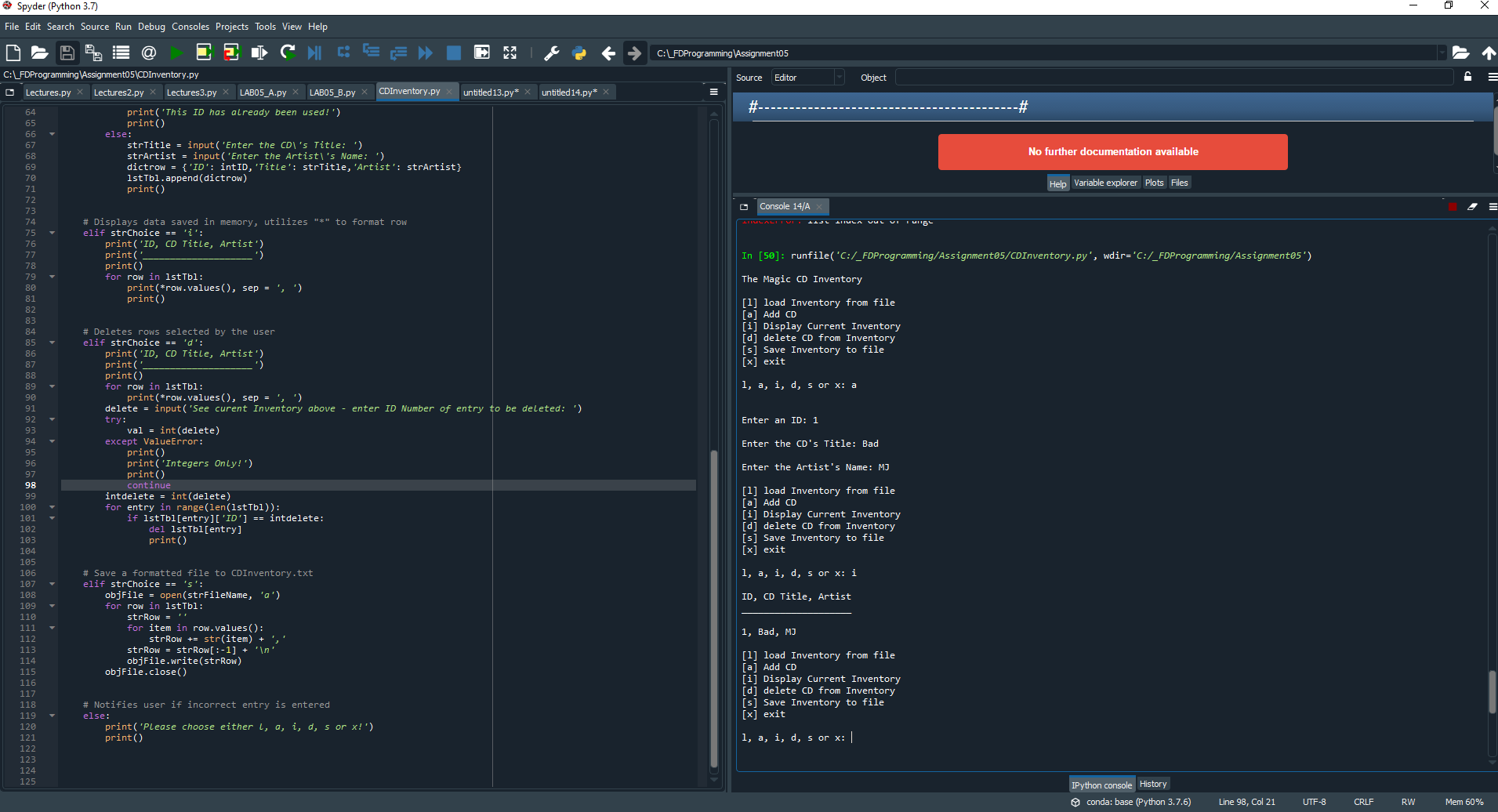


Figure 1 – Spyder run of CDInventory.py script

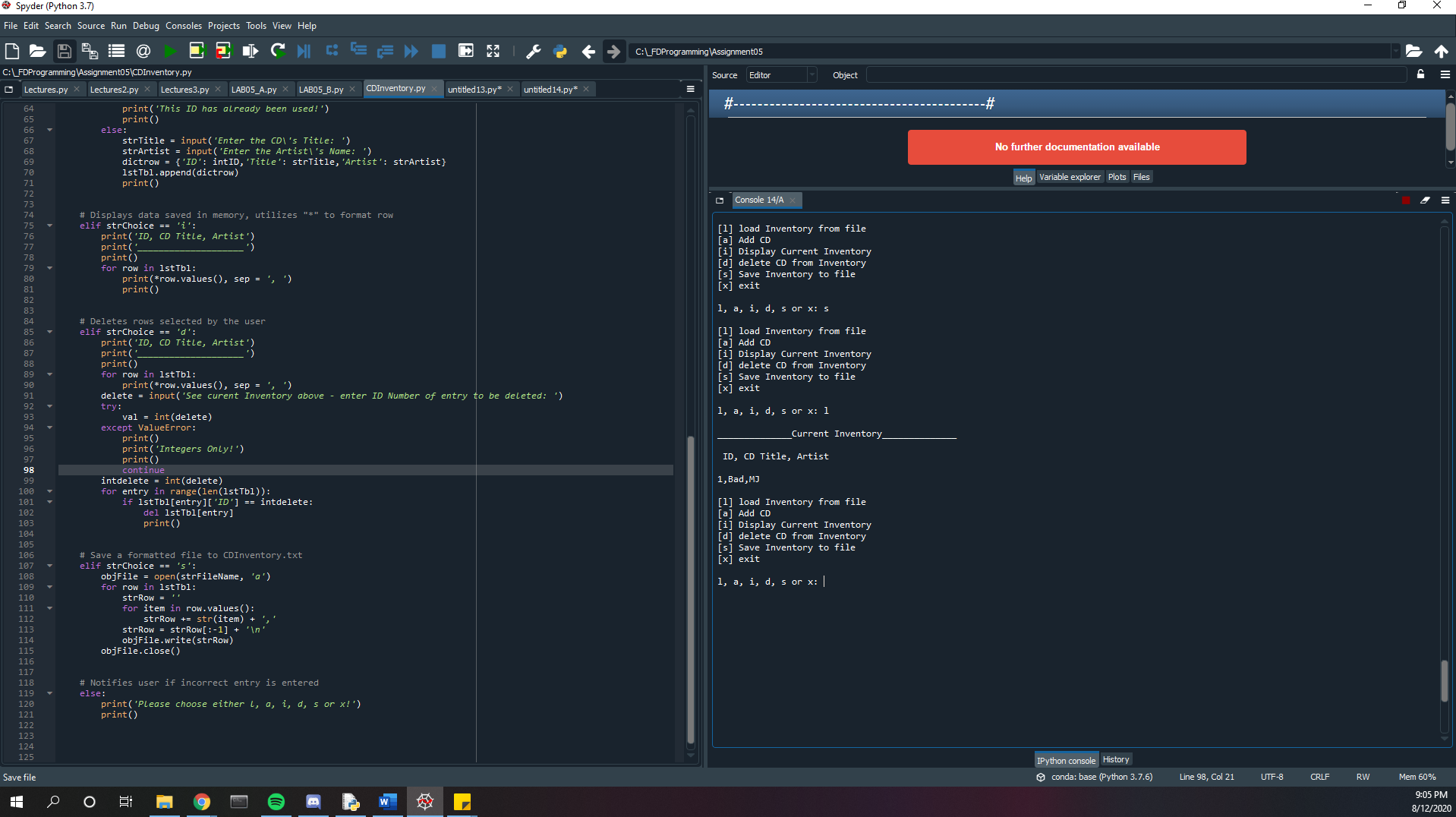


Figure 2 – Spyder run of CDInventory.py script

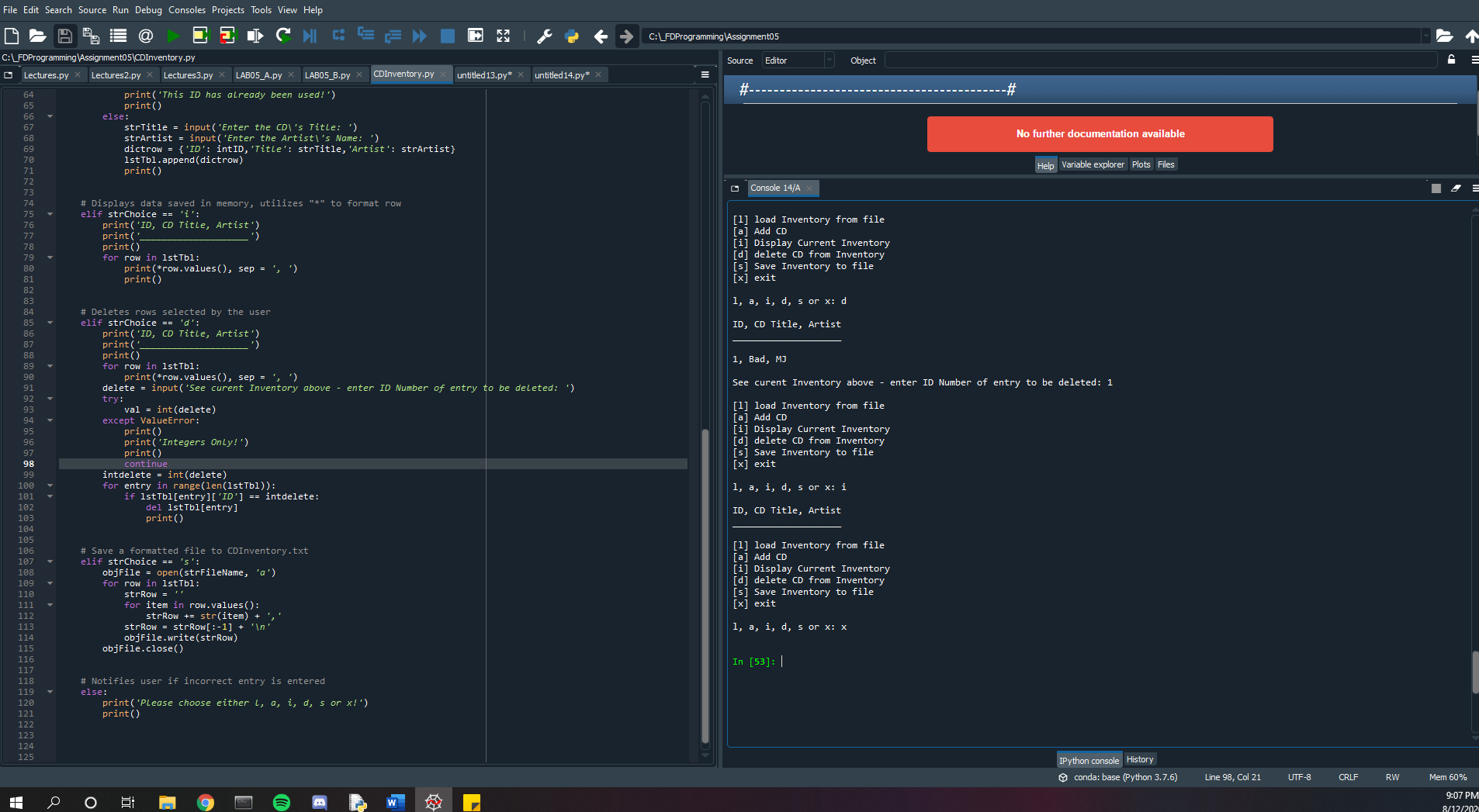


Figure - Spyder run of CDInventory.py script

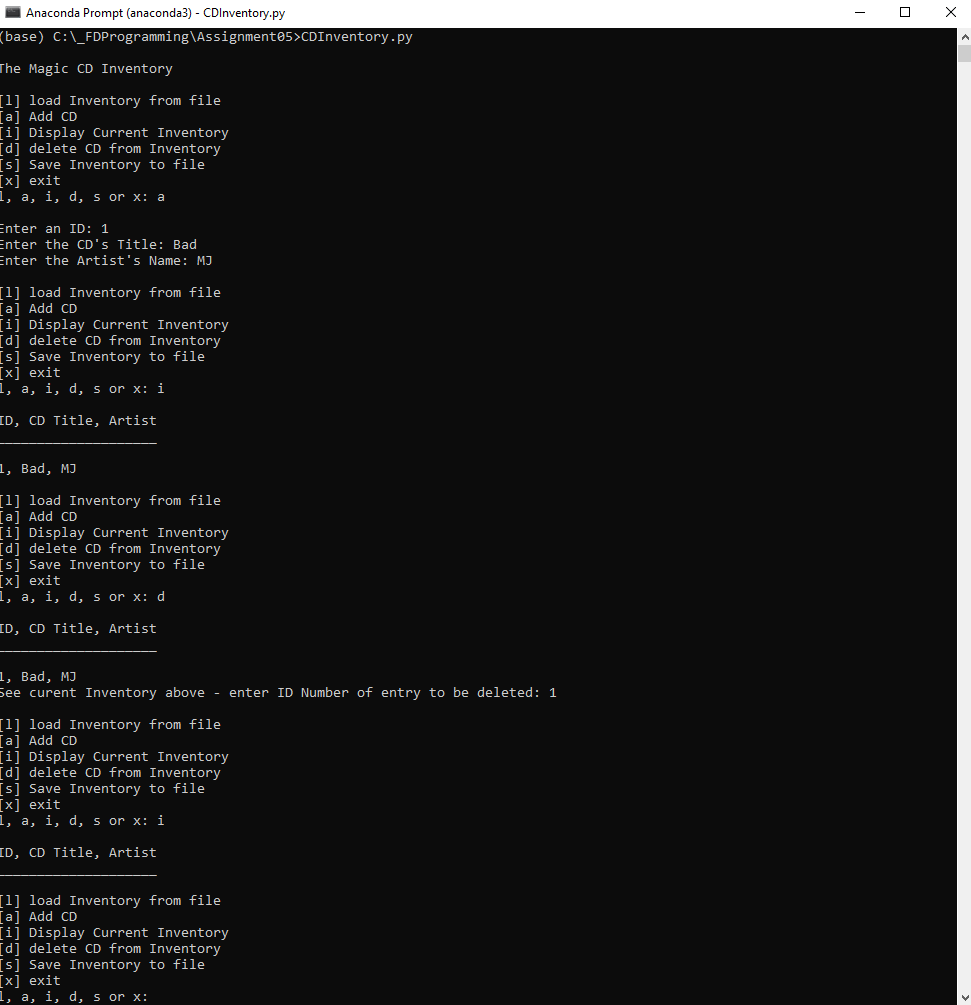


Figure - CDInventory.py running in console

The three primary areas I struggled with / had difficulty grasping were while loops with nested if statements (primarily resolved by the delete example shown in class… defining an “entry” than searching for it with “lstTbl[entry]['ID'] == intID” was just a format I hadn’t caught onto yet), the error handling (solved in part by this [PYnative webpage](https://pynative.com/python-check-user-input-is-number-or-string/) (external reference)[[1]](#footnote-1)), and general concatenation in loops (solved in part by the lectures and this [stack overflow webpage](https://stackoverflow.com/questions/5424716/how-to-check-if-string-input-is-a-number) (external reference)[[2]](#footnote-2)). My Github link is [here](https://github.com/jusroe/Assignment_05).

# Summary

Overall, the issues I had with this assignment were solved by websearches in conjunction with the course material. In my very brief programming experience in the past, I was 100% dependent on websearches to complete anything… you can really get far by using this method (although, its very nice to learn about the topics in a structured way in the course materials)! I trust that Github will aid with this as well!.

# Appendix A – Syntax

Generated using [planetb’s webpage](http://planetb.ca/syntax-highlight-word) (external reference)[[3]](#footnote-3) web page

1. #------------------------------------------#
2. # Title: CDInventory.py (For Assignment05)
3. # Desc: Script CDINventory to store CD Inventory data and allow user to make changes
4. # Change Log: (Who, When, What)
5. # JRoe, 2020-Aug-10, Created file
6. # JRoe, 2020-Aug-11, Modified code, added annotations
7. # JRoe, 2020-Aug-12, Modified code, added annotations
8. #------------------------------------------#
10. strChoice = '' # User input
11. lstTbl = []  # list of lists to hold data
12. dicrow = {}  # Create dictonary
13. strFileName = 'CDInventory.txt'  # data storage file
14. objFile = None  # file object

17. # Start script, initiate while loop
18. **print**()
19. **print**('The Magic CD Inventory\n')
20. **while** True:
21. #Display menu allowing the user to choose:
22. **print**('[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
23. **print**('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
24. strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
25. **print**()

28. # Exit script
29. **if** strChoice == 'x':
30. **break**
32. # use 'r' to read data from CDInventory.txt
33. **if** strChoice == 'l':
34. **print**('\_\_\_\_\_\_\_\_\_\_\_\_\_\_Current Inventory\_\_\_\_\_\_\_\_\_\_\_\_\_\_', '\n\n', 'ID, CD Title, Artist', '\n')
35. objFile = open(strFileName, 'r')
36. **for** row **in** objFile:
37. listrow = row.strip().split(',')
38. dicrow = {'ID': int(listrow[0]), 'CD Title': listrow[1], 'Artist': listrow[2]}
39. lstTbl.append(dicrow)
40. strRow = ''
41. **for** item **in** row:
42. strRow += str(item)
43. strRow = strRow[:-1] + '\n'
44. **print**(strRow)
45. objFile.close()

48. # Adds entries every time user asks - prevents user from entering ID values that already exist
49. **elif** strChoice == 'a':
50. strID = input('Enter an ID: ')
51. **try**:
52. val = int(strID)
53. **except** ValueError:
54. **print**()
55. **print**('Integers Only!')
56. **print**()
57. **continue**
58. intID = int(strID)
59. **for** entry **in** range(len(lstTbl)):
60. **if** lstTbl[entry]['ID'] == intID:
61. intID = 'new'
62. **break**
63. **if** intID == 'new':
64. **print**()
65. **print**('This ID has already been used!')
66. **print**()
67. **else**:
68. strTitle = input('Enter the CD\'s Title: ')
69. strArtist = input('Enter the Artist\'s Name: ')
70. dictrow = {'ID': intID,'Title': strTitle,'Artist': strArtist}
71. lstTbl.append(dictrow)
72. **print**()

75. # Displays data saved in memory, utilizes "\*" to format row
76. **elif** strChoice == 'i':
77. **print**('ID, CD Title, Artist')
78. **print**('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')
79. **print**()
80. **for** row **in** lstTbl:
81. **print**(\*row.values(), sep = ', ')
82. **print**()

85. # Deletes rows selected by the user
86. **elif** strChoice == 'd':
87. **print**('ID, CD Title, Artist')
88. **print**('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')
89. **print**()
90. **for** row **in** lstTbl:
91. **print**(\*row.values(), sep = ', ')
92. delete = input('See curent Inventory above - enter ID Number of entry to be deleted: ')
93. **try**:
94. val = int(delete)
95. **except** ValueError:
96. **print**()
97. **print**('Integers Only!')
98. **print**()
99. **continue**
100. intdelete = int(delete)
101. **for** entry **in** range(len(lstTbl)):
102. **if** lstTbl[entry]['ID'] == intdelete:
103. **del** lstTbl[entry]
104. **print**()

107. # Save a formatted file to CDInventory.txt
108. **elif** strChoice == 's':
109. objFile = open(strFileName, 'a')
110. **for** row **in** lstTbl:
111. strRow = ''
112. **for** item **in** row.values():
113. strRow += str(item) + ','
114. strRow = strRow[:-1] + '\n'
115. objFile.write(strRow)
116. objFile.close()

119. # Notifies user if incorrect entry is entered
120. **else**:
121. **print**('Please choose either l, a, i, d, s or x!')
122. **print**()

1. Retrieved 2020-Aug-12 [↑](#footnote-ref-1)
2. Retrieved 2020-Aug-12 [↑](#footnote-ref-2)
3. Retrieved 2020-Aug-12 [↑](#footnote-ref-3)