Muthulekshmi Sivathanu

02-13-2021

Foundations of programming python

Assignment-05

List of Dictionaries

# Introduction

This module explains in detail about Dictionaries and a collection of dictionaries in a List. The program of CD Inventory done in this module gives a practical experience of the maximum operations that can be done in a list of dictionaries.

# Dictionary

Dictionaries are used to store data values in pairs, which are called ‘keys’ and ‘values’. Items in dictionary cannot be referred using index. They can be referred using ‘key’ name. The values in dictionary item can be of any data type. They can be of mixed datatype too. ([External reference](https://www.w3schools.com/python/python_dictionaries.asp)[[1]](#footnote-1) ) Dictionary does not allow duplicate entries.

# List of Dictionaries

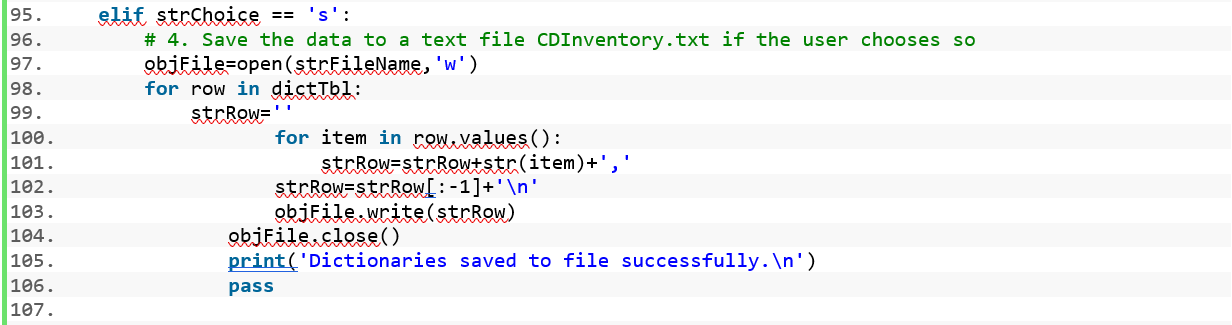
With python, we can have list of dictionaries. In the CD Inventory program executed in this module, we have created a dictionary with three Keys ‘CD ID, CD TITLE and Artist’. The corresponding data for these keys act as the ‘values’ in the dictionary. Multiple entries with the same key values can be made with having multiple dictionaries. All these dictionaries are enclosed in a list. Even though we create multiple entries similar to the first dictionary, it is required to give the key name again and again each time we create another dictionary. Each dictionary can be accessed separately through the indexing feature of list as they are entries of a List.

# Getting input from user and storing it in dictionary

3 Inputs are obtained from user. While entering these 3 data to a dictionary they are entered along with their respective keys. Since we must add more such dictionaries, we have appended each dictionary to a list. Multiple dictionaries in the list form a list of dictionaries. Append function is used to add each row to the list.

# Saving data to file

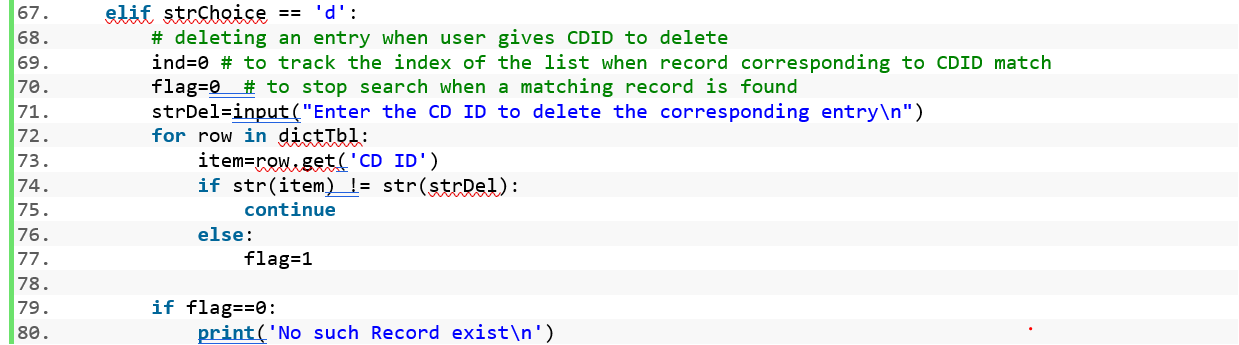
A file is opened in the write mode. Each value in the list of dictionaries (2D table) is written to file one by one. Keys are not written to the file. For each dictionary in the list, we must convert all the elements into strings. We can separate two values of the same dictionary with a comma. This can be done by concatenation as we are performing this operation on the string. After we finish writing one dictionary in the file, we can use a new line character to jump to the next line and then start inserting the next dictionary. ( [External Reference](https://pythonspot.com/save-a-dictionary-to-a-file/)[[2]](#footnote-2) )

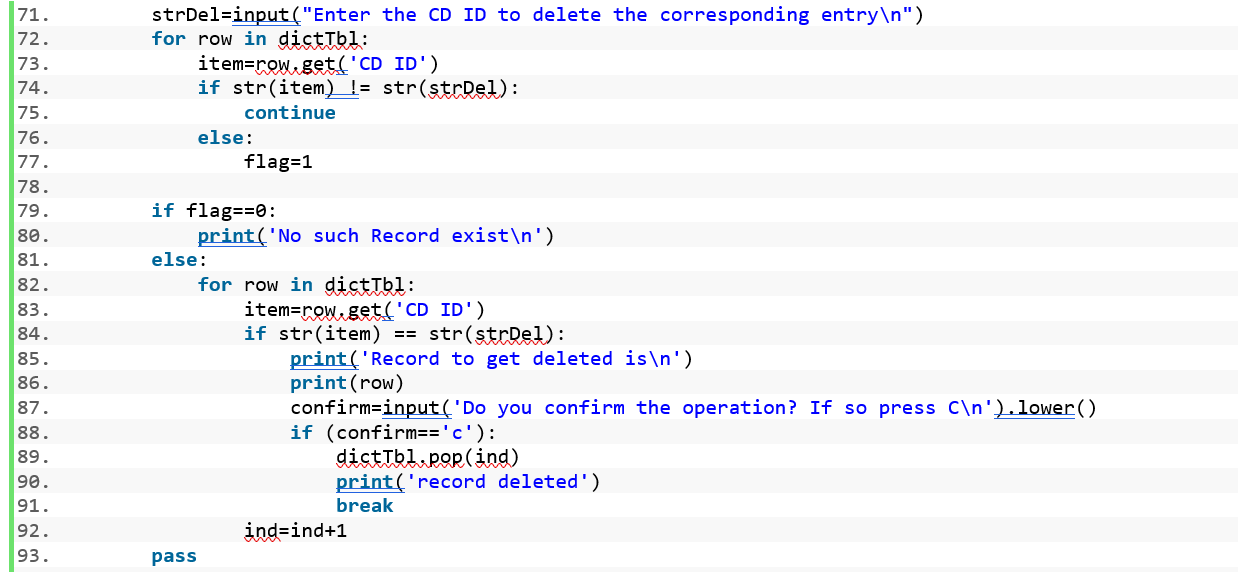


Listing 1 Saving Inventory data to file

# Deleting an entry

We get the CD ID from the user to delete the corresponding entry (dictionary) in the list. The variable row in the first for loop will contain the first dictionary entry. The variable item in the second for loop with contain the value corresponding to CD ID in the first dictionary. When we compare the value in item to the CD ID the user has entered, we must convert both to strings. When we find a match, we delete. Deletion is done using pop method. Among three values in the dictionary, a particular value is obtained through get method, with the respective key. We are not deleting the entry from the file unless the user confirms the deletion operation. Once he confirms we are updating the file with the new list, which has one dictionary deleted.

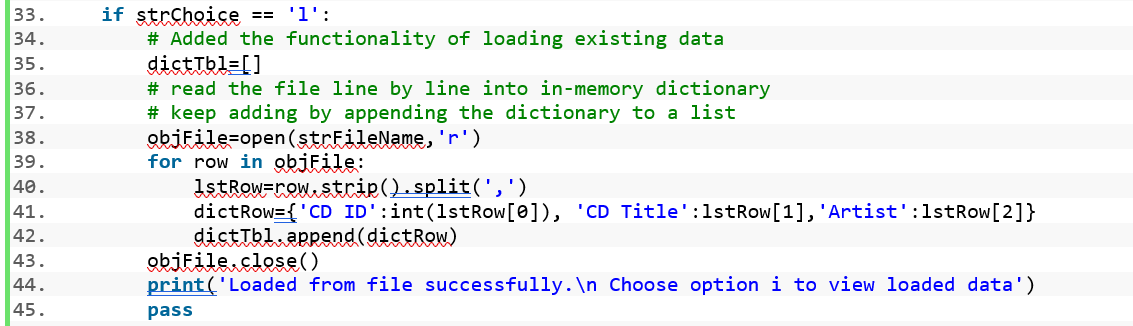




Listing 2 Delete entry from Inventory

# Load inventory from file

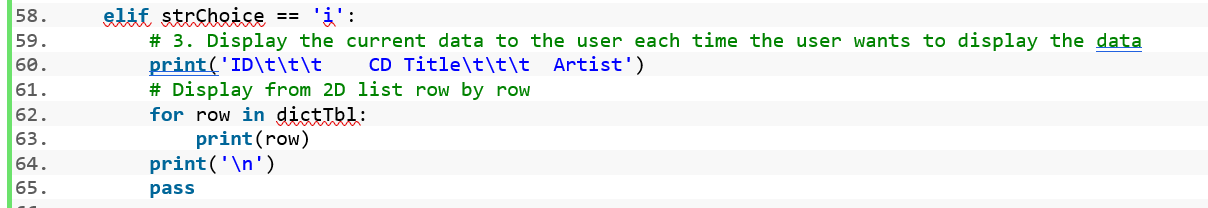
File contains only the value and not the keys. When we load entries from file, we must make sure that we insert the respective key entries. This is done through creating a dictionary with keys entered manually by us and values fetched from file. For consecutive rows in the file, the same operation is done, and the list of dictionaries is made.



Listing 3 Load Inventory

# Display current inventory.

Display of the entries is done from the table which is the 2D list of dictionaries. We extract row by row from the table and display as it exists in the table.



Listing 4 Display current inventory.

# Summary

In this module we learnt about dictionaries and list of dictionaries. Various operations were made on the list such as writing the entries to the file and loading the data back from the file, inserting an element, deleting an element, and displaying it. The coding was easier in one aspect, since the structure was already given, but it was more complex as the coding was not a result of free flow of my thought process of approaching the problem. It made me think at a point that, it would be better to write the program by myself from the scratch. The final version of the program with the document is uploaded at https://github.com/muthulekshmis/Assignment\_05[[3]](#footnote-3)

# Appendix

## CDInventory.py

1. #------------------------------------------#
2. # Title: CDInventory.py
3. # Desc: ADD,DEL,SAVE,LOAD CD DATA - Assignment 05
4. # Change Log: (Who, When, What)
5. # DBiesinger, 2030-Jan-01, Created File
6. # Muthu, 2021-Feb-12, Added Fuctionality to save data to file
7. # Muthu, 2021-Feb-13, Added Functionality  to load data from file to a table
8. # Muthu, 2021-Feb-14, Added Functionality to delete a record
9. #------------------------------------------#
11. # Declare variables

14. strChoice = '' # User input
15. dictTbl = []  # list of dictionaries to hold data
16. dictRow = {} # list of data row
17. strFileName='CDInvent.txt'  # data storage file
18. objFile = None  # file object
20. # Get user Input
21. **print**('\nThe 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**
33. **if** strChoice == 'l':
34. # Added the functionality of loading existing data
35. dictTbl=[]
36. # read the file line by line into in-memory dictionary
37. # keep adding by appending the dictionary to a list
38. objFile=open(strFileName,'r')
39. **for** row **in** objFile:
40. lstRow=row.strip().split(',')
41. dictRow={'CD ID':int(lstRow[0]), 'CD Title':lstRow[1],'Artist':lstRow[2]}
42. dictTbl.append(dictRow)
43. objFile.close()
44. **print**('Loaded from file successfully.\n Choose option i to view loaded data')
45. **pass**
47. **if** strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
48. # 2. Add data to the table (2d-list) each time the user wants to add data
49. strID = input('Enter an ID: ')
50. strTitle = input('Enter the CD\'s Title: ')
51. strArtist = input('Enter the Artist\'s Name: ')
52. intID = int(strID)
53. lstRow = [intID, strTitle, strArtist]
54. dictRow={'CD ID':intID,'CD Title':strTitle,'Artist':strArtist}
55. dictTbl.append(dictRow)
56. **pass**
58. **elif** strChoice == 'i':
59. # 3. Display the current data to the user each time the user wants to display the data
60. **print**('ID\t\t\t    CD Title\t\t\t  Artist')
61. # Display from 2D list row by row
62. **for** row **in** dictTbl:
63. **print**(row)
64. **print**('\n')
65. **pass**
67. **elif** strChoice == 'd':
68. # deleting an entry when user gives CDID to delete
69. ind=0 # to track the index of the list when record corresponding to CDID match
70. flag=0  # to stop search when a matching record is found
71. strDel=input("Enter the CD ID to delete the corresponding entry\n")
72. **for** row **in** dictTbl:
73. item=row.get('CD ID')
74. **if** str(item) != str(strDel):
75. **continue**
76. **else**:
77. flag=1
79. **if** flag==0:
80. **print**('No such Record exist\n')
81. **else**:
82. **for** row **in** dictTbl:
83. item=row.get('CD ID')
84. **if** str(item) == str(strDel):
85. **print**('Record to get deleted is\n')
86. **print**(row)
87. confirm=input('Do you confirm the operation? If so press C\n').lower()
88. **if** (confirm=='c'):
89. dictTbl.pop(ind)
90. **print**('record deleted')
91. **break**
92. ind=ind+1
93. **pass**
95. **elif** strChoice == 's':
96. # 4. Save the data to a text file CDInventory.txt if the user chooses so
97. objFile=open(strFileName,'w')
98. **for** row **in** dictTbl:
99. strRow=''
100. **for** item **in** row.values():
101. strRow=strRow+str(item)+','
102. strRow=strRow[:-1]+'\n'
103. objFile.write(strRow)
104. objFile.close()
105. **print**('Dictionaries saved to file successfully.\n')
106. **pass**
108. **else**:
109. **print**('Please choose either l, a, i, d, s or x!')

Text

Description automatically generated

Figure Terminal output of CDInventory program

Text

Description automatically generated

Figure 2 Terminal Output of CDInventory program

Text

Description automatically generated

Figure Spyder output of CDInventory program

Text

Description automatically generated

Figure 4 Spyder Output of CDInventory program

Text

Description automatically generated

Figure 5 Spyder output of CDInventory program

1. Referred on 02-14-2021. [↑](#footnote-ref-1)
2. Referred on 02-13-2021 [↑](#footnote-ref-2)
3. Referred on 14-02-2021 [↑](#footnote-ref-3)