**Name: Srirupa Guha**

**Date**: August 12, 2020

**Course**: IT FDN 110: Introduction to Programming (Python)

**Assignment**: 5

**Title**: Knowledge Document – More Strings, Tuples and Dictionaries

**Introduction** –

This week we learned more about lists and tuples, as well as a new variable type called the dictionary, as well as using GitHub (creating an account, setting it up, and finally using it for uploading files).

**Lists**

So lists are usually accessed by index from zero till the number which is 1 less than the length of the list. List items can also be addressed in a reverse format starting from -1, which is the last item in the list, and moving leftwards. Example, a list looking like this: mammals = [‘cat’,’bat’,’rat’] can be indexed as 0, 1 and 2, and also -1, -2 and -3 in reverse from right to left. So mammals[-3] is ‘cat’, mammals[2] is ‘bat’, for example. You can also use ranges as in: mammals[0:2], which will retrieve ‘cat’ and ‘bat’. You can also use functions for lists such as min(), max() and len(). Lists can carry items that can be of several types, such as integers, floats, strings, lists, tuples, dictionaries, and even hierarchies of lists, tuples and dictionaries. Several methods can be used for list manipulation, such as list.append(), list.insert(), list.extend(), list.remove(), list.pop() and others. Lists are totally mutable and dynamic that they can grow and shrink. The following site is really helpful in illustrating many characteristics and functionalities of lists: [Lists and Tuples](https://realpython.com/python-lists-tuples/) [[1]](#footnote-1)

**Tuples**

Tuples behave similar to lists in the sense that, like lists any data variable type can be contained in a tuple (strings, integers, floats, lists, tuples, dictionaries). Tuple items are accessed and indexed the same manner as lists. However, tuples are immutable and therefore they are not dynamic like lists. Tuples are better used for situations where the data change is not desired, such as some sensitive logs or set profiles. A good use for tuples is in keys of dictionaries due to its immutable nature.

**Dictionaries**

The third data variable type studies are dictionaries. Dictionaries are lists of items consisting of key:value pairs, such as client\_profile = {‘name’:’Joe’,’gender’:’male’,’age’:’37’,’pets’:[‘cat’,’dog’,’ferret’]}

The keys, like name, gender, etc. are always unique, like no more than one ‘age’ in the dictionary. When accessing data, the key of the item (key:value pair) is used, as in client\_profile[‘name’]. A dictionary is dynamic like the list in the sense that it can grow and shrink, can contain several data type in values (although not lists for keys). The website [Dictionaries](https://realpython.com/python-dicts/)[[2]](#footnote-2) is very useful in explaining the various features and functions of dictionaries.

**GitHub**:

We also learned about using GitHub, how to create an account, then setting up the user profile, creating a repository for code, uploading code and other files, and downloading from the GitHub repository for further editing/appending.

Ultimately, I put all this knowledge to test and wrote and executed my first Python program:

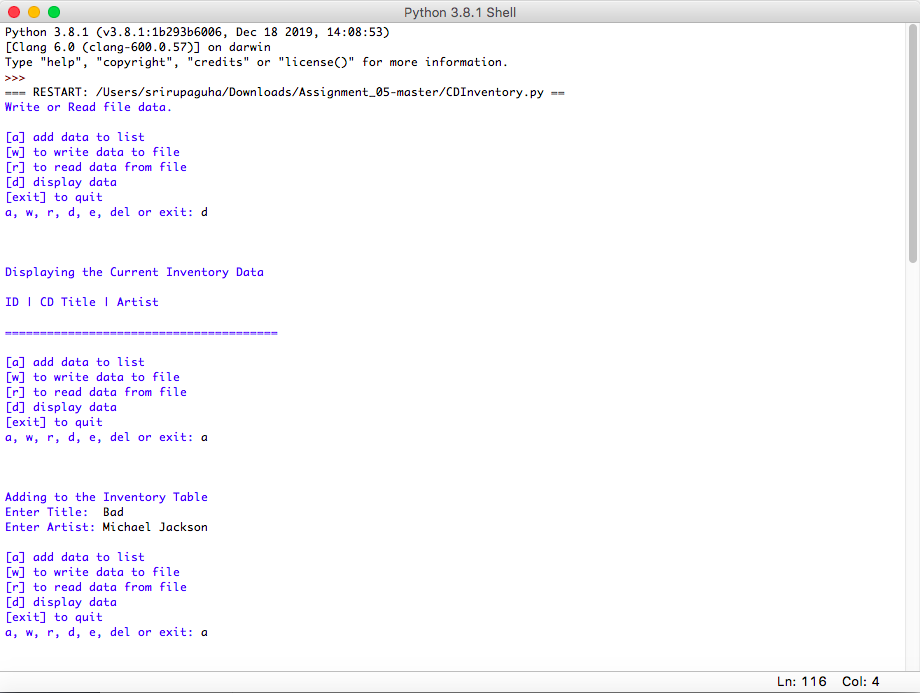


Figure 1. Running my Code from CDInventory.py

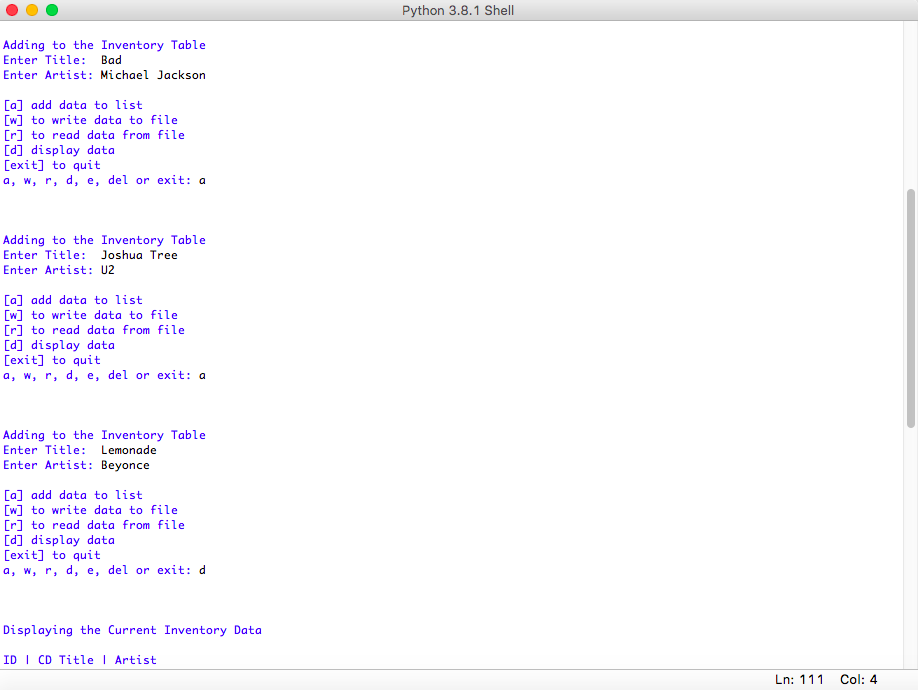


Figure 2 Continued - CDInventory.py

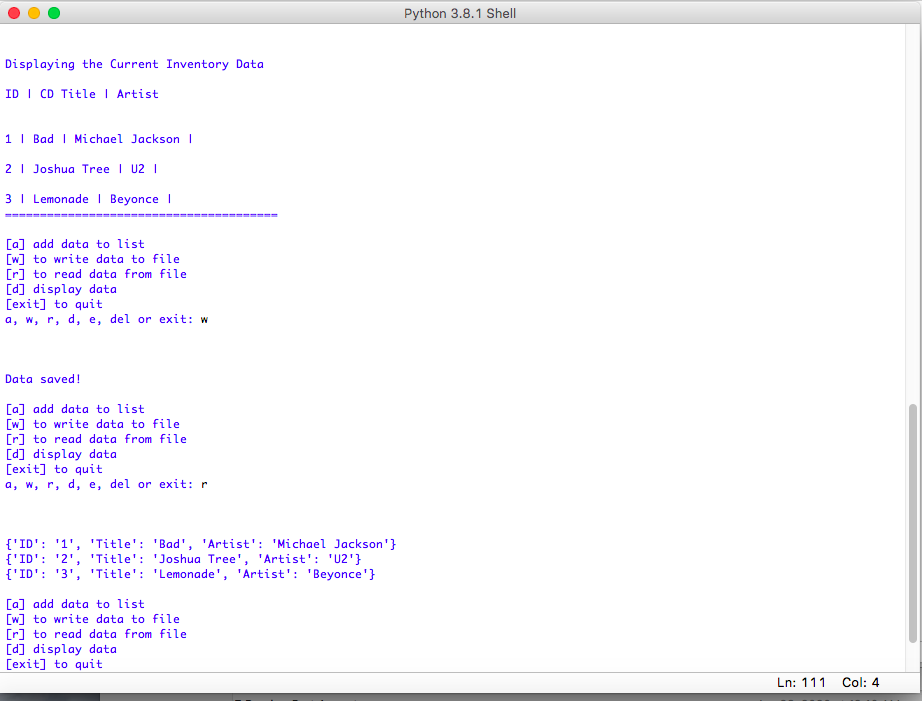


Figure 3 continued - CDInventory.py

**Summary**

This week we learned more about lists and tuples, as well as a new variable type called the dictionary, as well as using GitHub (creating an account, setting it up, and finally using it for uploading files).

**Appendix**

The Python assignment code for this week:

1. #------------------------------------------#
2. # Title: CDInventory.py
3. # Desc: CDInventory - Creating, Appending, Reading to and Writing from Dictionaries
4. # Change Log: (Who, When, What)
5. # Srirupa Guha, 2020-Aug-09, Created File
6. #------------------------------------------#
8. # Declare variabls
10. strChoice = '' # User input
11. lstTbl = []  # list of lists to hold data
12. dicRow = {}  # 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, e, del 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 reached if strChoice is not 'exit'
28. # Add data to list in memory
30. **print**("Adding to the Inventory Table")
31. cd\_Title = input("Enter Title:  ")
32. cd\_Artist = input("Enter Artist: ")
34. dicRow = {"ID":(len(lstTbl)+1),"Title":cd\_Title,"Artist":cd\_Artist}
35. lstTbl.append(dicRow)
37. #pass
39. **elif** strChoice == 'w':
41. # List to File
43. new\_line = ""
44. objFile = None
46. **for** row **in** lstTbl:
47. **for** item **in** row.values():
48. new\_line = new\_line + str(item) + ","
49. new\_line = new\_line[:-1] + "\n"
51. objFile = open(strFileName, "w")
52. objFile.write(new\_line)
53. objFile.close()
54. **print**("Data saved!")
56. #pass
58. **elif** strChoice == 'r':
60. # File to print
62. lstRow = []
63. dicRow = {}
64. objFile = None
66. objFile = open(strFileName, "r")
67. **try**:
68. **for** row **in** objFile:
69. lstRow = row.strip().split(",")
70. dicRow = {"ID":lstRow[0],"Title":lstRow[1],"Artist":lstRow[2]}
71. **print**(dicRow)
72. **except**:
73. **print**("Something went wrong - maybe there is nothing to read.")
75. objFile.close()
77. #pass
79. **elif** strChoice == 'd':
81. # Display data
83. **print**("Displaying the Current Inventory Data\n")
85. **print**("ID"+" | "+"CD Title"+" | "+"Artist")
87. **for** row **in** lstTbl:
88. **print**("\n")
89. **for** item **in** row.values():
90. **print**(item,end=" | ")
91. **print**("\n=======================================")
93. #pass
95. **elif** strChoice == 'e':
97. # Edit data
99. old\_data = input("What would you like to edit or replace? ")
101. **for** row **in** lstTbl:
102. **if** old\_data **in** row.values():
103. new\_data = input("What is the new value? ")
104. new\_data = old\_data
105. **break**
106. **else**:
107. **print**("This term does not exist!")
109. #pass
111. **elif** strChoice == 'del':
113. # delete data
115. del\_ID = str(input("What is the ID of the record to remove? "))
117. **for** row **in** lstTbl:
118. **if** del\_ID **in** row.values():
119. row.delete
120. **break**
121. **else**:
122. **print**("Row ID does not exist")
124. #pass
126. **else**:
127. **print**('Please choose either a, w, r or exit!')

1. <https://realpython.com/python-lists-tuples/> Retrieved August 12, 2020 [↑](#footnote-ref-1)
2. <https://realpython.com/python-dicts/> Retrieved on August 12, 2020 [↑](#footnote-ref-2)