# Dictionary versus List:

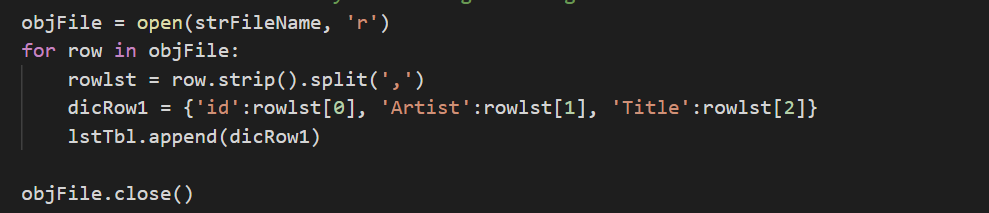
Lists can hold a collection of objects and come with a number of build-in functions. To introduce a dictionary, the braces operator is utilized. Dictionary keys are similar to named columns in a spread sheet or database, and the dictionary is a row of data. Dictionaries can be inserted into lists to create a list of dictionaries 2D table, which is like creating a list of list to form a 2D list for a 2D table. Using the append function, user input can be added to a dictionary. *(Biesinger Dirk, FDN\_Py\_Module\_05, retrieved from: https://canvas.uw.edu/courses/1424622/pages/module-05-overview?module\_item\_id=11567701)*

# Index versus Key:

All store data in a sequence and are accessible via an index (subscript). Dictionaries are similar to sequence types, but replace the index and storing in sequence by storing key: value pairs. Strings, Numbers and Tuples can always be used as keys. *(Biesinger Dirk, FDN\_Py\_Module\_05, retrieved from: https://canvas.uw.edu/courses/1424622/pages/module-05-overview?module\_item\_id=11567701)*

# Reading data from a file:

When reading data from a file, the ‘r’ in string objFile is utilized. With writing the data into the file, the ‘w’ in string objFile is used. Here is an example:



When reading data from a list, the print format can be utilized. Here is an example:

print('ID, Artist, Title')

for cd in cd\_inventory:

print ("{}, {}, {}".format(cd[0], cd[1], cd[2]))

# Functions

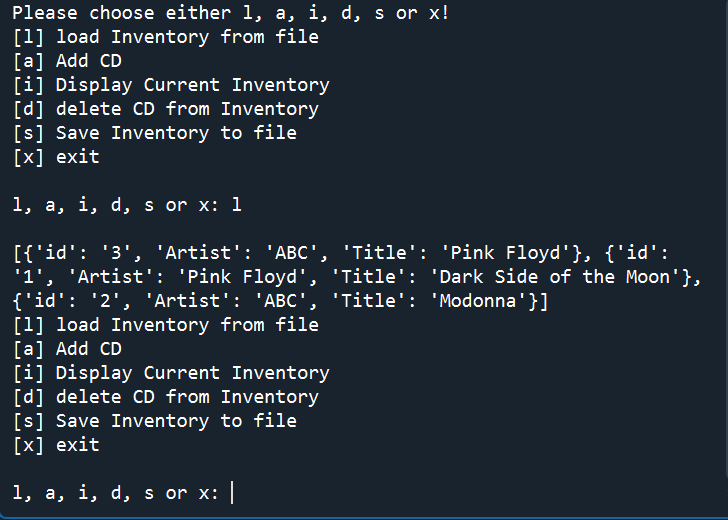
Functions allow you to group a set of statements and make them accessible via a given name. Functions must be defined in Python before they can be called. That way, the group of statements are executed when the function is called by its name. The declaration of a function loads the function into memory, but the code is not executed. For example, the keyword DEF can be utilized to define a function. An example is Def getsum() which is defining a sum function. *(Biesinger Dirk, FDN\_Py\_Module\_05, retrieved from: https://canvas.uw.edu/courses/1424622/pages/module-05-overview?module\_item\_id=11567701)*

# Assignment05

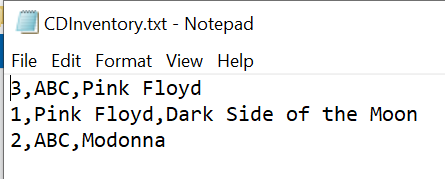
The started script was used and saved as CDInventory.py. The steps to perform this assignment was as follows:

1. The dictionary was defined. Uses braces, I declared the dictionary row.
2. The next step was to load existing data. I used the string ObjFile = open (strFilename, ‘r’) to achieve this. The strfilename was used at the beginning of the script to define the text in the CDInventory text file.
3. Deleting an entry required utilizing the remove.

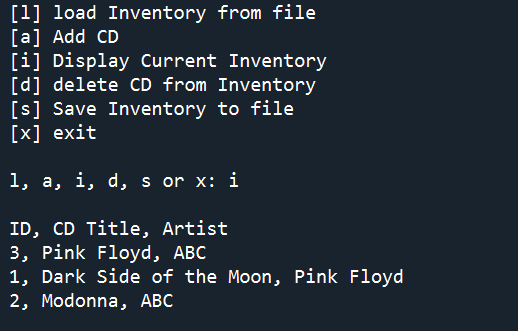
Running in Spyder:



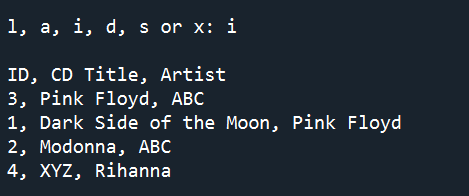
Choosing l allows us to load the data into the notepad file:



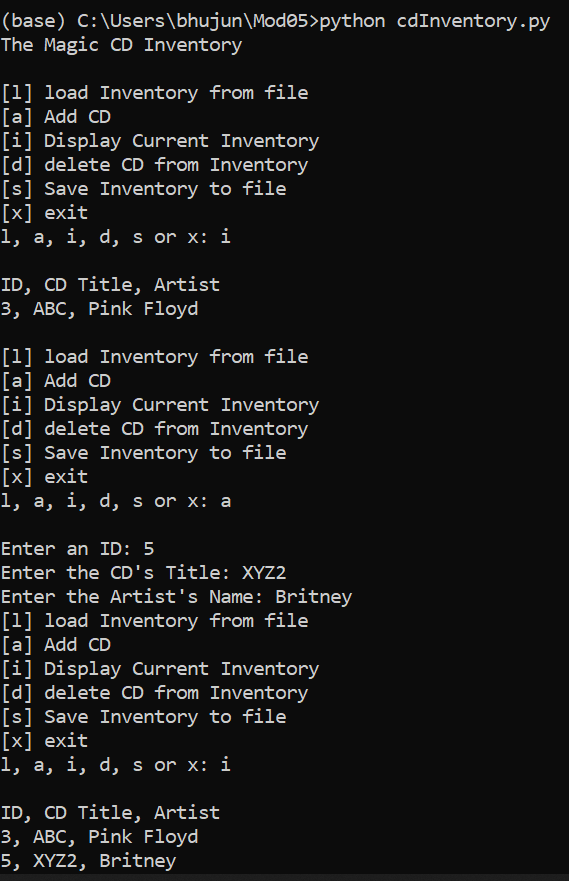
Displaying current inventory produced the following output:



Adding data (selection a) would allow me to add input and I then displayed the data:



Running in terminal:



Loading the inventory from file shows the following data in the text file:



# Summary:

In this module, we learned how to use dictionaries, and the difference between dictionaries and lists. We also incorporated what we have learned previously using the ‘for’ loops, and the ‘if and elif’ functions. This assignment helped us to bring together all the different functions we have used throughout the course. We used the string ‘w’ in objFile to write the data into the text file, and we also can use the string ‘r’ to read data from the text file.