John Stevens

March 02, 2020

IT FDN 100 B

Assignment 06

CD Inventory(w/functions) Script

# Introduction

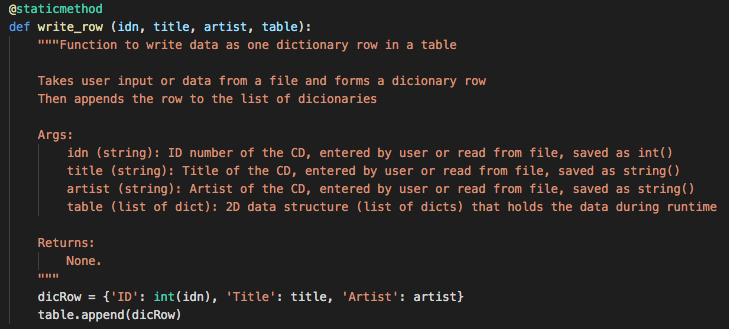
This document will show the modification of a pre coded script in Python to use functions that can load an existing list or create a new one, ask a user to input an inventory of CDs, display the inventory, delete any inputted CDs and then save the inventory list to a file.

# Script Creation

I began by declaring one variable not included in the original script, set the variable ‘saveFlag” to 0 for tracking unsaved changes.

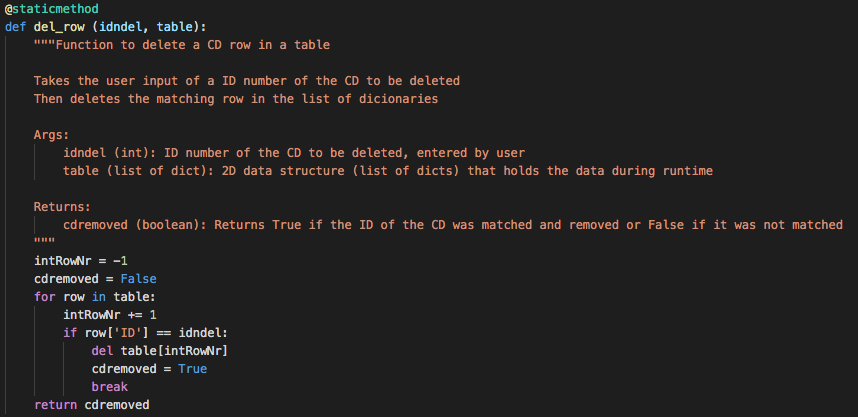
Then I began by adding functions (PYTHON PROGRAMMING FOR THE ABSOLUTE BEGINNER 3RD ADDITION, MICHAEL DAWSON, CENGAGE LEARNING 2010, P. 157-188), to the provided Class DataProcessor which contains function to process the data during runtime.

The first function write\_row,



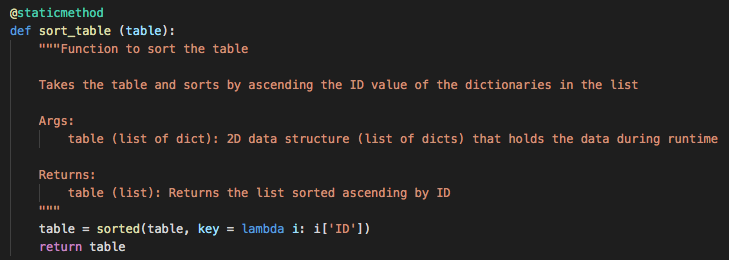
This function takes the user input or data from a file in the passed variables “idn, title, artist” and forms a dictionary row in “dicRow{}, then appends the row to the list of dictionaries defined in “table”.

The next function del\_row,



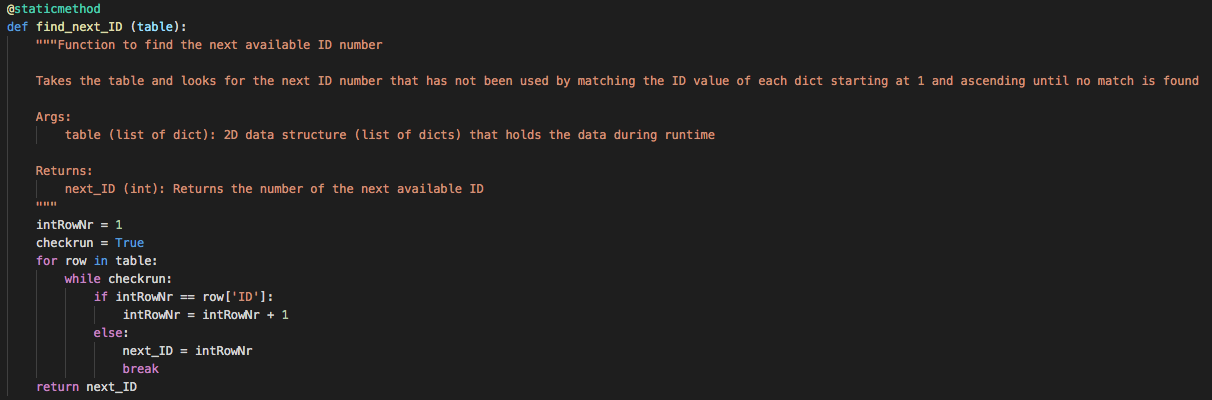
This function takes the user input of a ID number, “idndel”, of the CD to be deleted then deletes the matching row in the list of dictionaries defined in “table”. The function tries to match the variable in “intRowNr” to each row[ID] in “table”, if found that row dict is deleted and the variable “cdremoved” is set to true. If there is no match then cdremoved” is set to false. Then “cdremoved” is returned.

The next function is sort\_table,



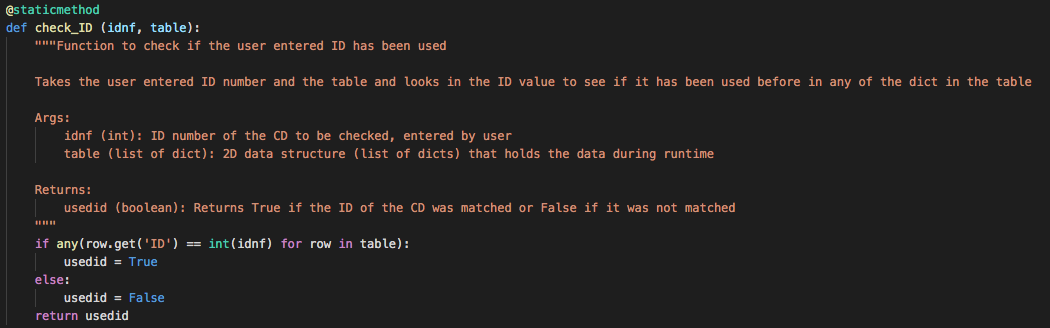
Takes the table and sorts by ascending the ID value of the dictionaries in the list. The “table” is sorted by the sorted() function https://www.geeksforgeeks.org/sorted-function-python/[[1]](#footnote-1)1, by the ID of each dictionary in the list “table”. Then “table” is returned.

The next function is find\_next\_ID,



This function takes “table” and looks for the next ID number that has not been used by matching the row[ID] value of each dictionary in the list “table” starting at 1 and ascending until no match is found. Once no match is found that int() is saved into the variable “next\_ID” and that is returned.

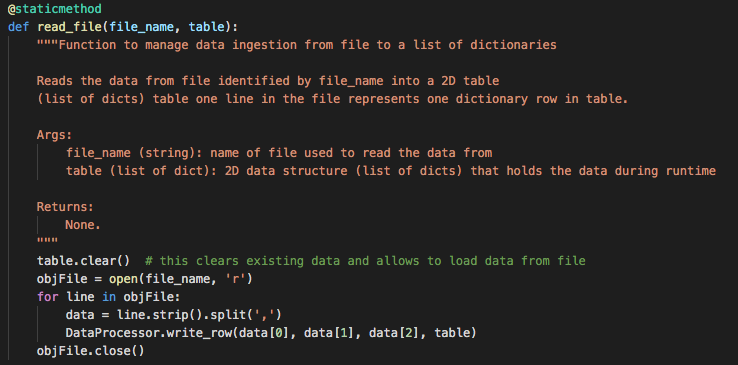
The final function in Class DataProcessor is check\_ID,



Takes the user entered ID number in “indf” and the table in “table” and looks in the ID value using list comprehension to see if it has been used before in any of the dictionaries in the table. If found the variable “usedid” is set to True, otherwise is set to False, ”usedid” is then returned.

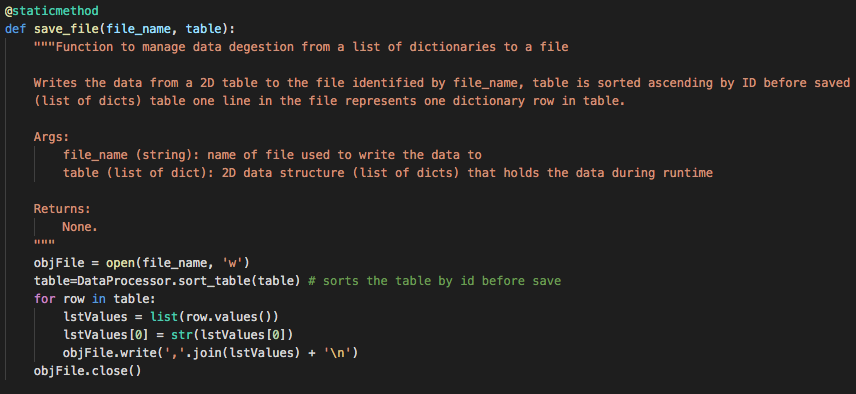
Next up is the Class FileProcessor that has functions that handle the processing the data to and from a text file.

The first function is read\_file,



This function reads the data from the file identified by file\_name into a 2D table. It first clears any data in the “table” variable then opens the file in the “file\_name” variable with the read flag. Each row in the file is then loaded into “data” and that is passed by array number in the correct order to the DataProcessor.write\_row function along with the table to write to. The file is then closed.

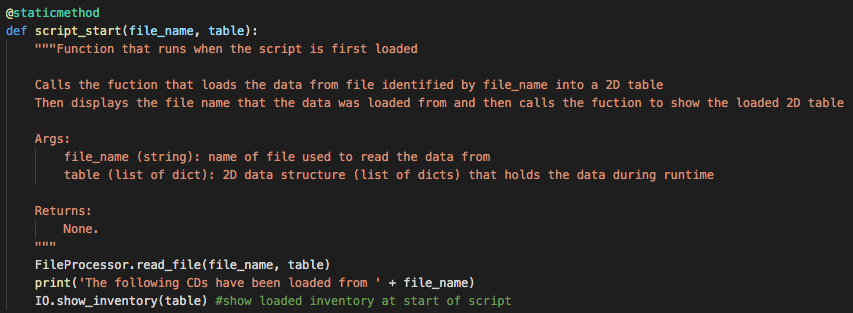
The next function is save\_file,



This function writes the data from a 2D table in “table” to the file identified by “file\_name. First the file is opened using the file in the “table” variable with the write flag. Next the table is sorted ascending by ID before being saved by calling the DataProcessor.sort\_table function. Then each dictionary in the list table is saved with the write method. Finally the file is closed.

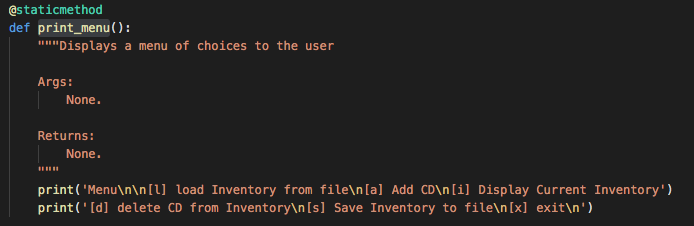
The final class IO handles Input / Output in the script.

The first function is script\_start,



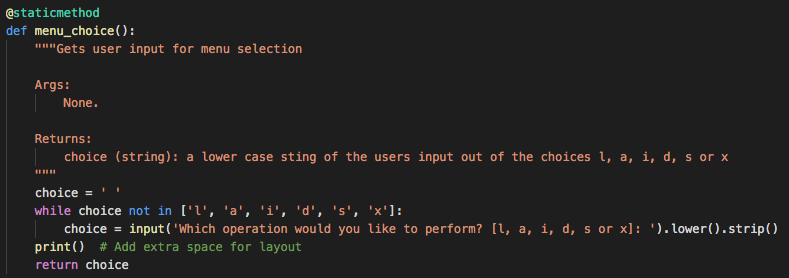
This function runs when the script is started. It calls the function FileProcessor.read\_file with the file\_name and table passed on to load the inventory into the table. Then the file loaded is presented to the user followed by IO.show\_inventory being called to show the inventory to the user.

The next function is print\_menu,



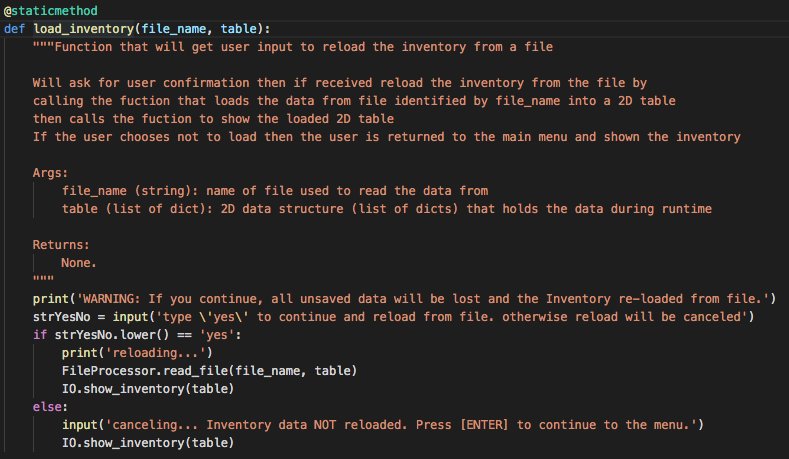
This function prints the menu choices to the user for the main menu of the script.

The next function is menu\_choice,



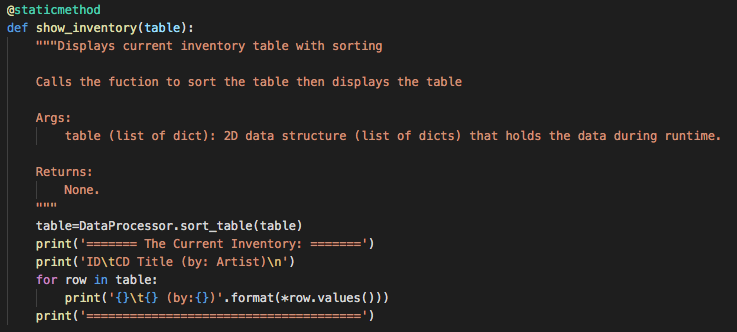
This function asks for user input for the main menu that must be either 'l', 'a', 'i', 'd', 's', 'x' and puts it into the variable “choice”, “choice” is then retuned.

The next function is load\_inventory,



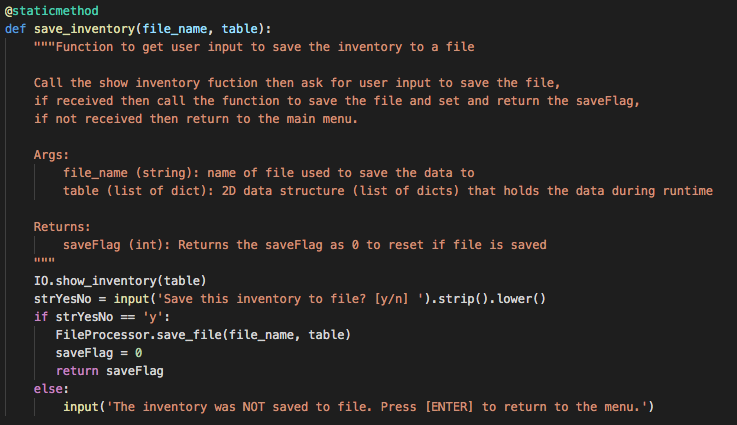
This function asks if the users wants to reload the inventory from the file. If so the function will call the FileProcessor.read\_file with “file\_name” and “table” passed on to load the file and then call IO.show\_inventory with “table”to display the current inventory to the user. Else the function will wait for a keystroke and then display the inventory back to the user.

The next function is show\_inventory,



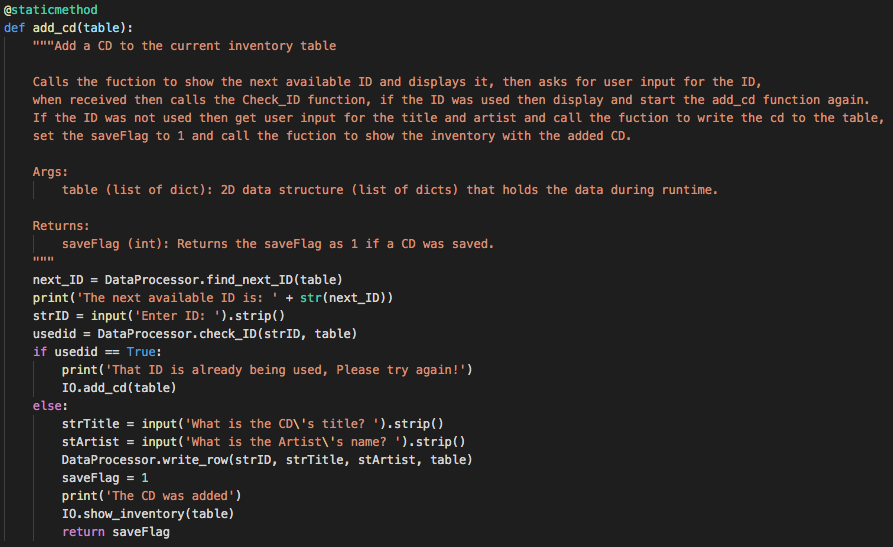
This function will display the current inventory to the user. It starts by calling the DataProcessor.sort\_table function with the passed “table” variable to sort the table first then will take the sorted “table” and display each CD as a row to the user.

The next function is save\_inventory,



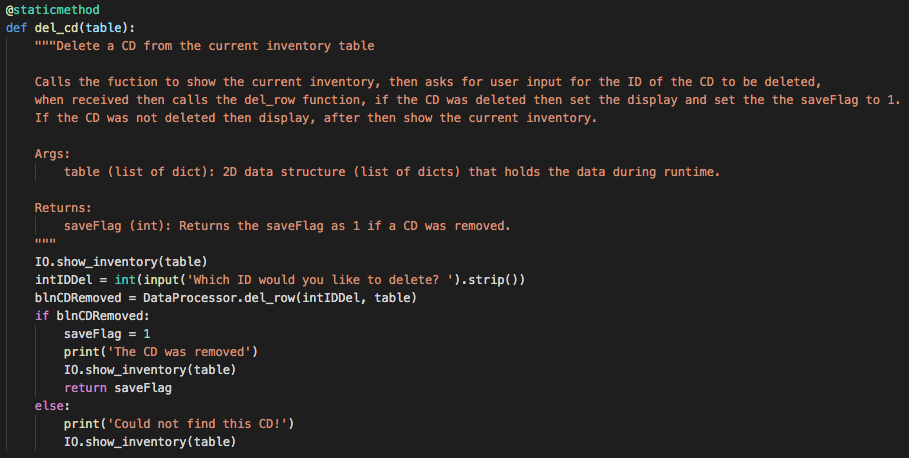
This function asks for user input to save the inventory to the file. The function starts by calling the IO.show\_inventory function to show the current inventory to the user then asks for user input if the inventory should be saved. If so then the function calls the function FileProcessor.save\_file passing the “file\_name” and “table” variables to save the file and the saveFlag is set to 0 and returned. Else the script returns to main menu without saving.

The next function is add\_cd,



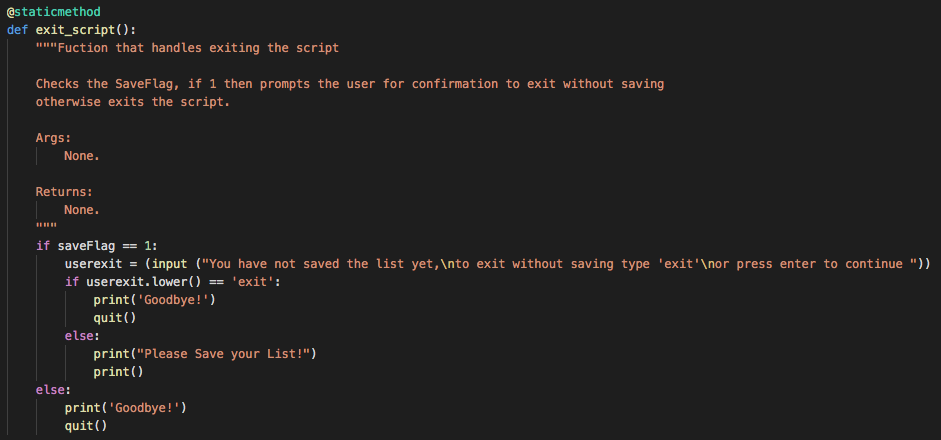
This function asks for user input to add a CD to the inventory. It starts by calling the DataProcessor.find\_next\_ID function and displaying it to show the next available cd not used in the inventory. Then the user is prompted to enter the ID to be used, this is checked by the DataProcessor.check\_ID function to see if the ID entered currently being used in the inventory. If so the users is told this and the IO.add\_cd function is restarted. If not then the user is prompted for the Title and Artist of the CD, this is then passed to the DataProcessor.write\_row function to save the CD into the table, the inventory is then shown again with the new cd added and the saveFlag is set to 1 and returned.

The next function del\_cd,



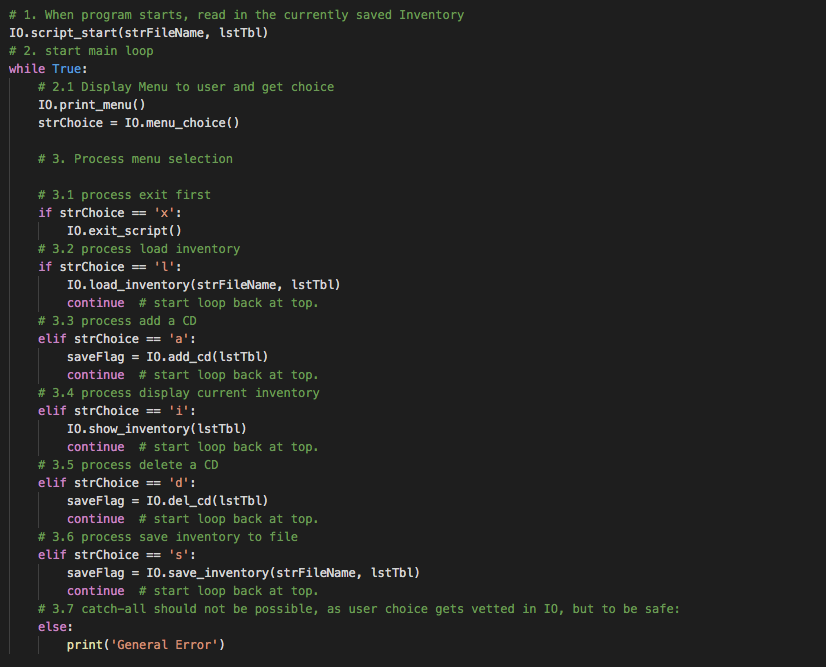
This function deletes an CD from user input. First the inventory is shown to the user and asks the user for the ID of the CD they wish to delete. This is passed in the variable “intIDDEL” with “table” to the function DataProcessor.del\_row which tries to delete the CD. If successful then the inventory is then shown again without the deleted CD and the saveFlag is set to 1 and returned. If not then the inventory is shown again and the user is returned to the main menu.

The final function is exit\_script,



This script will prompt the user for confirmation if the inventory has not been saved if they try and exit the script. The function looks to see if the saveFlag is set to 1, if so the user will be prompted to enter exit to leave the script without saving. If not then the user is returned to the main menu. If the saveFlag is set to 0, the script exits.

The main body of the script,



starts by calling the IO.script\_start function with the variables “strFileName” and “lstTbl” to load and display the inventory from the file. Then the IO.print\_menu is displayed and user input is requested with the IO.Menu\_choice function. The users input is stored into “strChoice”. Then depending on the users input the script will do the following, If the user enters ‘x’ the function IO.exit\_script will be called. If ‘l’ is entered then the function IO.load\_inventory is called with the variables “strFileName” and “lstTbl” passed on. If ‘a’ is entered then the function IO.add\_cd is called with the variable “lstTbl” passed on. If ‘I’ is entered then the function IO.show\_inventory is called with the variable “lstTbl” passed on. If ‘d’ is entered then the function IO.del\_cd is called with the variable “lstTbl” passed on. If ‘s’ is entered then the function IO.save\_inventory is called with the variables “strFileName” and “lstTbl” passed on.

# Performing the Script

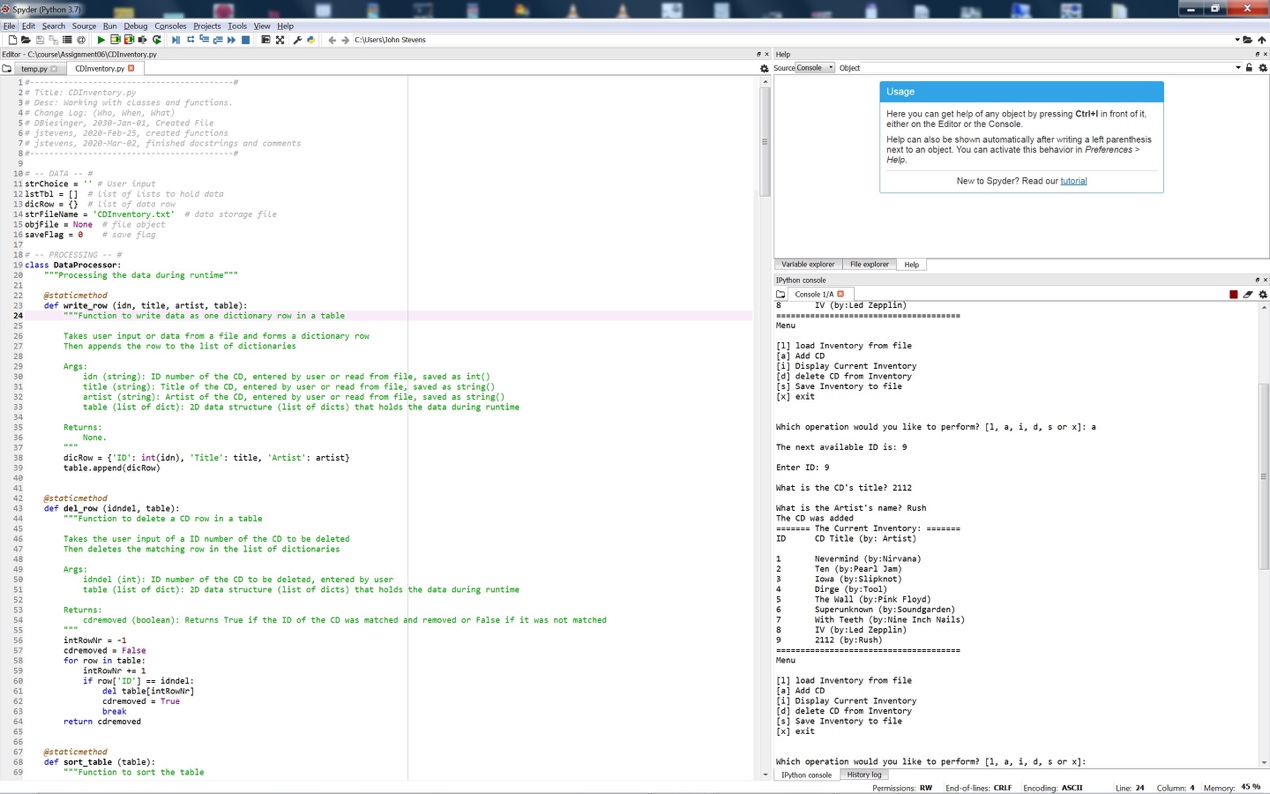


Figure 2 - Script performing in Spyder.

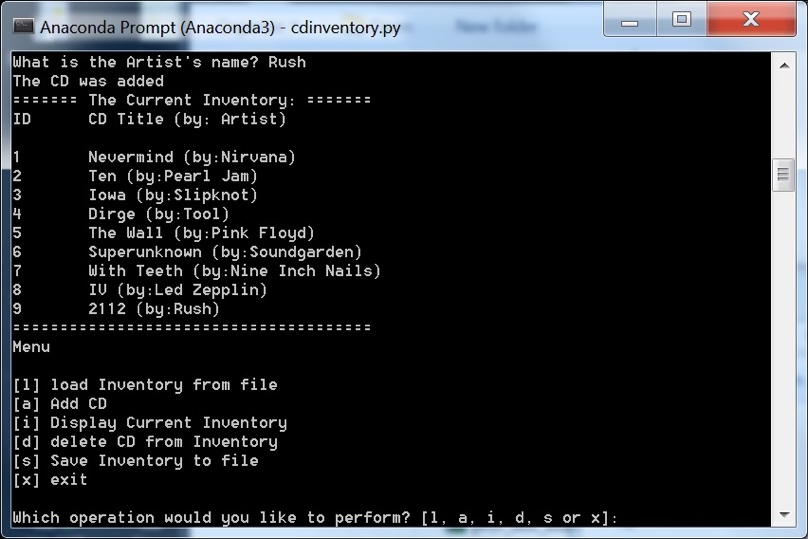


Figure 3 - Script performing in terminal

As seen in Figure 2 and Figure 3 I ran the script in Spyder and Terminal respectively. I loaded 8 test CDs from the file CDInventory.txt, then added 9th CD,displayed the list and saved the list to a file. When I checked the file CDInventory.txt I see that the input was saved correctly as seen in Figure 4.

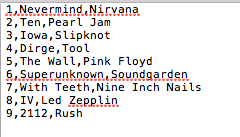


Figure 4 – CD input saved in CDInventory.txt file.

# Summary

In this assignment I was able to modify an existing script to use functions and classes to provide expanded functionality to load, add, delete, show and save CDs to a list. This script and document and posted at https://github.com/pjfan73/Assignment06

1. 1 Retrieved 2020-02-25 [↑](#footnote-ref-1)