Christopher Manzuk

06 Mar 2022

Foundations of Programming Python

Assignment 6

Lesson Six – List and Dictionaries

Introduction

Lesson Six discusses Functions and Classes, variable scopes and DocStrings.

CD Inventory.py

The task is to upgrade a version of our CD Inventory program to a class/function structure. Most main procedure code was moved into functions within pre-defined classes, and full documentation ('docstrings') was added to each function. An operating code from Assignment 5 was supplied. The final script itself, with trimmed documentation, is shown below:

"CDInventory.py"

```
# -- DATA -- #
strChoice = '' # User input
lstTbl = []  # list of dicts to hold data
dicRow = {}  # dict of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object


# -- PROCESSING -- #
class DataProcessor:


    @staticmethod
    def InputCD_LineItem():
        """Function to collect new line item data

        Takes input strings and returns a list of the strings

        Args:
            None
        Returns:
            (list): List of line item input strings
            strID (string): line item ID number
            strTitle (string): CD title
            stArtist (string): CD artist
        """

        strID = input('Enter ID: ').strip()
        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        print()

        return[strID,strTitle,stArtist]     # return a List


    @staticmethod
    def addCD_LineItem (newRow, table):           # imput parameters as lists
        """Function to append new row of data to existing table

        Makes needed data type conversion, creates dict row from the input list,
        appends to existing table

        Args:
            newRow (list): line item elements as strings
            table (list of dict): 2D data structure that holds the data during runtime

        Returns:
            table (list of dict): (same)
        """

        dicRow = {'ID': int(newRow[0]), 'Title': newRow[1], 'Artist': newRow[2]}
        table.append(dicRow)

        return table


    @staticmethod
    def delCD_LineItem (IDselect, table):    # input parameters  int, list
        """Function to delete a selected row of data from existing table

        Takes user's ID selection, converts to list index number, identifies
        the corresponding row in the table, and deletes it.

        Args:
            IDselect (integer): user selection for row to be deleted
            table (list of dict): 2D data structure that holds the data during runtime

        Returns:
            table (list of dict): 2D data structure that holds the data during runtime
        """

        intRowNr = -1
        blnCDRemoved = False
        for row in table:
            intRowNr += 1
            if row['ID'] == IDselect:
                del table[intRowNr]
                blnCDRemoved = True
                break

        if blnCDRemoved:
            print('The CD was removed\n')
        else:
            print('Could not find this CD!')

        return table


class FileProcessor:
    """Processing the data to and from text file"""


    @staticmethod
    def read_file(file_name, table):
        """Function to manage data ingestion from file to a list of dictionaries

        Reads the data from file identified by file_name into a 2D table
        (list of dicts); one line in the file represents one dictionary row in table.

        Args:
            file_name (string): name of file used to read the data from
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None.
        """

        table.clear()  # this clears existing data and allows to load data from file

        objFile = open(file_name, 'r')
        for line in objFile:
            data = line.strip().split(',')
            dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
            table.append(dicRow)
        objFile.close()


    @staticmethod
    def write_file(file_name, table):
        """Function to manage data outout from the list of dictionaries to a file

        Writes the data from a 2D table (list of dicts) into file identified by file_name;
        one line in the file represents one dictionary row in table.
        Args:
            file_name (string): name of file used to read the data from
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None.
        """

        # DONE Add code here

        objFile = open(file_name, 'w')

        for row in table:
            lstValues = list(row.values() )
            lstValues[0] = str(lstValues[0])
            objFile.write(','.join(lstValues) + '\n')
        objFile.close()


# -- PRESENTATION (Input/Output) -- #

class IO:
    """Handling Input / Output"""


    @staticmethod
    def print_menu():
        """Displays a menu of choices to the user

        Args:
            None.

        Returns:
            None.
        """

        print('\n\nMenu\n\n[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
        print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit\n')


    @staticmethod
    def menu_choice():
        """Gets user input for menu selection

        Args:
            None.

        Returns:
            choice (string): a lower case sting of the users input out of the choices l, a, i, d, s or x
        """
        choice = ' '
        while choice not in ['l', 'a', 'i', 'd', 's', 'x']:
            choice = input('Which operation would you like to perform? [l, a, i, d, s or x]: ').lower().strip()
        print()  # Add extra space for layout
        return choice


    @staticmethod
    def show_inventory(table):
        """Displays current inventory table

        Args:
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime.

        Returns:
            None.
        """

        print('\n======= The Current Inventory: =======')
        print('ID\tCD Title (by: Artist)\n')
        for row in table:
            print('{}\t{} (by:{})'.format(*row.values() ) )
        print('======================================\n\n')


# 1. When program starts, read in the currently saved Inventory
FileProcessor.read_file(strFileName, lstTbl)

# 2. start main loop
while True:

    # 2.1 Display Menu to user and get choice
    IO.print_menu()
    strChoice = IO.menu_choice()


    # 3. Process menu selection

    # 3.1 process exit first
    if strChoice == 'x':
        break


    # 3.2 process load inventory
    if strChoice == 'l':
        print('WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from
file.')
        strYesNo = input('type \'yes\' to continue and reload from file. otherwise reload will be canceled :')

        if strYesNo.lower() == 'yes':
            print('\nreloading...\n')
            FileProcessor.read_file(strFileName, lstTbl)
            IO.show_inventory(lstTbl)
        else:
            input('\ncanceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.\n')
            IO.show_inventory(lstTbl)
        continue  # start loop back at top.


    # 3.3 process add a CD
    elif strChoice == 'a':

        # 3.3.1 Ask user for new ID, CD Title and Artist
        # DONE - move IO code into function

        lstNewRow=[]
        lstNewRow = DataProcessor.InputCD_LineItem()

        # 3.3.2 Add item to the table
        # DONE move processing code into function

        lstTbl = DataProcessor.addCD_LineItem(lstNewRow,lstTbl)
        print()
        IO.show_inventory(lstTbl)

        continue  # start loop back at top.


   # 3.4 process display current inventory
    elif strChoice == 'i':
        IO.show_inventory(lstTbl)
        continue  # start loop back at top.


    # 3.5 process delete a CD
    elif strChoice == 'd':

        # 3.5.1 get Userinput for which CD to delete
        # 3.5.1.1 display Inventory to user
        IO.show_inventory(lstTbl)

        # 3.5.1.2 ask user which ID to remove
        intIDDel = int(input('\nWhich ID would you like to delete? \n').strip())

        # 3.5.2 search thru table and delete CD
        # DONE move processing code into function

        lstTbl = DataProcessor.delCD_LineItem (intIDDel, lstTbl)
        IO.show_inventory(lstTbl)

        continue  # start loop back at top.
    # 3.6 process save inventory to file
    elif strChoice == 's':

        # 3.6.1 Display current inventory and ask user for confirmation to save
        IO.show_inventory(lstTbl)
        strYesNo = input('\nSave this inventory to file? [y/n] \n').strip().lower()

        # 3.6.2 Process choice
        if strYesNo == 'y':

            # 3.6.2.1 save data
            # DONE move processing code into function

            FileProcessor.write_file(strFileName, lstTbl)

        else:
            input('\nThe inventory was NOT saved to file. Press [ENTER] to return to the menu.\n')

        continue  # start loop back at top.


    # 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be save:
    else:
        print('\nGeneral Error')
```

==============================

Converting to the class/function structure was relatively straightforward. A minor stumbling block was getting the syntax correct for passing the table list to functions. Another was naming the table list consistently in the main procedure and functions to keep the global/local relationship correct.

The last challenge was identifying the correct location to clear the table list when adding or deleting an entry. For instance, at one point the add function was overwriting the existing CD list with only the new CD entry, despite using the <list>.append method.

Adding the docstrings to the functions was uncomplicated, but a bit time consuming, as one would expect.

CD Inventory – IDE Run

Running the code in the spyder IDE worked without incident, as shown in the console text stream below. Red text indicates comments describing which functions are being demonstrated. The final output file is shown in Figs 1 ("CDInventory.txt").

Console Output:

```
In [24]:         0  '/Users/CMM/Dropbox/Rcds_370--
Edu_UW_Py/ Assignment06/Assignment06.py'

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: i
```

## Program start automatically loads existing data file, shown with selection 'i'

```
======= The Current Inventory: =======

ID CDTitle(by:Artist)

1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)

======================================

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: d

======= The Current Inventory: =======

ID CDTitle(by:Artist)

1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)

======================================
```

## Before demonstrating the 'load' function, both existing entries are deleted, also testing the 'delete' function

```
Which ID would you like to delete?
1
The CD was removed

======= The Current Inventory: =======

ID CDTitle(by:Artist)

2 Bad (by:Michael Jackson)

======================================

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: d

======= The Current Inventory: =======

ID CDTitle(by:Artist)

2 Bad (by:Michael Jackson)

======================================

Which ID would you like to delete?
2
The CD was removed

======= The Current Inventory: =======

ID CDTitle(by:Artist)

======================================

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: l
```

## Initial file was reloaded. First tested the 'cancel load' option, leaving file still empty.

```
WARNING: If you continue, all unsaved data will be lost
and the Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise
reload will be canceled :
```

```
canceling... Inventory data NOT reloaded. Press [ENTER]
to continue to the menu.


======= The Current Inventory: =======


ID CDTitle(by:Artist)


======================================


Menu
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: l
```

## Load function reselected, tested the option to actually load file

```
WARNING: If you continue, all unsaved data will be lost
and the Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise
reload will be canceled :yes

reloading...
======= The Current Inventory: =======


ID CDTitle(by:Artist)


1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)


======================================


Menu


[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: a
```

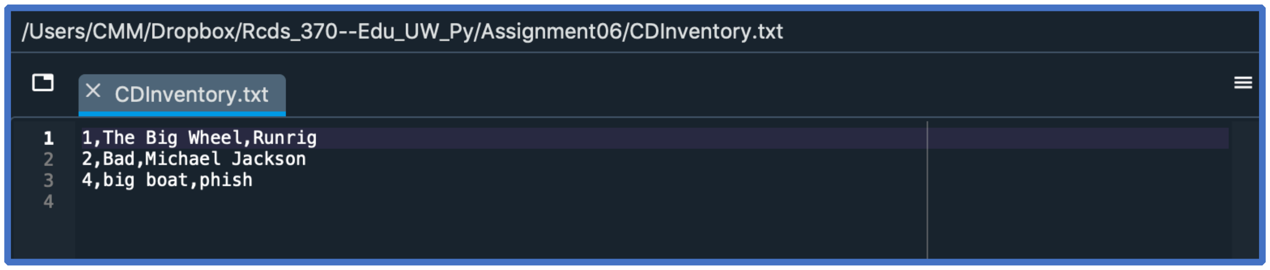
## 2 additional line items using ‘add’

```
Enter ID: 3
What is the CD's title? fuego
What is the Artist's name? phish


======= The Current Inventory: =======


ID CDTitle(by:Artist)


1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
3 fuego (by:phish)


======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d,
s or x]: a

Enter ID: 4
What is the CD's title? big boat
What is the Artist's name? phish


======= The Current Inventory: =======


ID CDTitle(by:Artist)


1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
3 fuego (by:phish)
4 big boat (by:phish)


======================================


Menu


[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: d
```

## Delete function demonstrated again, confirming that after deleting a middle row, the function will return the correct updated table. There had been problems with this during debugging.

```
======= The Current Inventory: =======


ID CDTitle(by:Artist)


1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
3 fuego (by:phish)
4 big boat (by:phish)


======================================
Which ID would you like to delete?
3
The CD was removed



======= The Current Inventory: =======


ID CDTitle(by:Artist)


1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
4 big boat (by:phish)


======================================
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: s
  ## tested 'save' function and its cancel option

======= The Current Inventory: =======

ID CDTitle(by:Artist)

1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
4 big boat (by:phish)

======================================

Save this inventory to file? [y/n]
x
The inventory was NOT saved to file. Press [ENTER] to
return to the menu.

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: i
  ## 'display' to confirm the table is unchanged

======= The Current Inventory: ======= ID
CDTitle(by:Artist)

1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
4 big boat (by:phish)
======================================
```

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: s
  ## tested 'save' function with an actual save

======= The Current Inventory: =======

ID CD Title(by:Artist)

1 The Big Wheel (by:Runrig)
2 Bad (by:Michael Jackson)
4 big boat (by:phish)
======================================

Save this inventory to file? [y/n]
y

Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
Which operation would you like to perform? [l, a, i, d,
s or x]: x

In [24]:

In [24]:
```

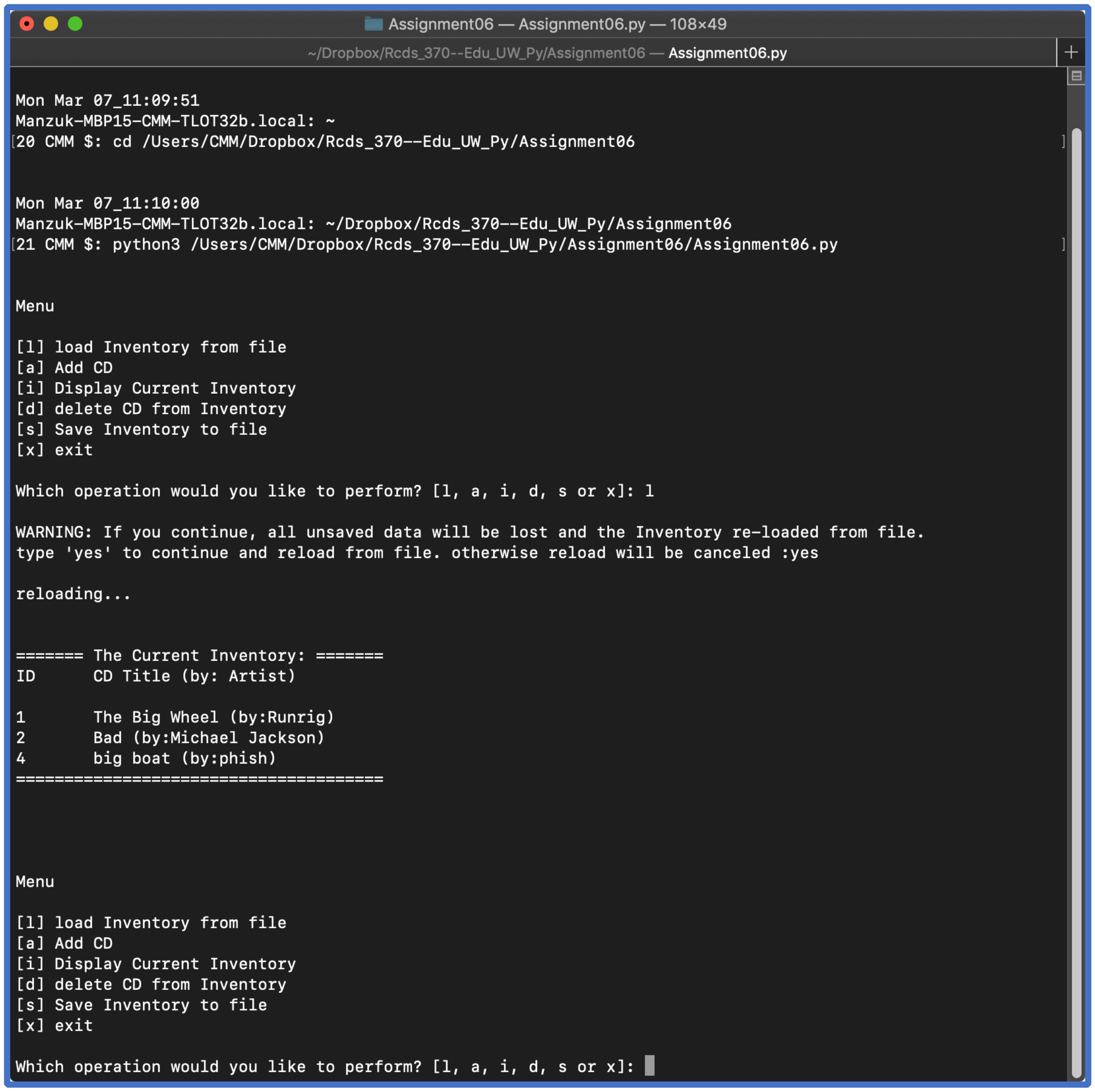

/Users/CMM/Dropbox/Rcds_370--Edu_UW_Py/Assignment06/CDInventory.txt

```
1,The Big Wheel,Runrig
2,Bad,Michael Jackson
4,big boat,phish
```

Figure 1 - Output File “CDInventory.txt” in Final State

CD Inventory – Terminal Run

The ongoing difficulty of running the scripts in the Macos Terminal was solved. Mac OS X uses python 2.x at the system level – therefore, if the python version is unspecified, it defaults to v2.x. When 'python3' is used to run scripts, they run as advertised.



Assignment06 — Assignment06.py — 108×49
~/Dropbox/Rcds_370--Edu_UW_Py/Assignment06 — Assignment06.py

```
Mon Mar 07_11:09:51
Manzuk-MBP15-CMM-TLOT32b.local: ~
[20 CMM $: cd /Users/CMM/Dropbox/Rcds_370--Edu_UW_Py/Assignment06


Mon Mar 07_11:10:00
Manzuk-MBP15-CMM-TLOT32b.local: ~/Dropbox/Rcds_370--Edu_UW_Py/Assignment06
[21 CMM $: python3 /Users/CMM/Dropbox/Rcds_370--Edu_UW_Py/Assignment06/Assignment06.py


Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.
type 'yes' to continue and reload from file. otherwise reload will be canceled :yes

reloading...


======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       The Big Wheel (by:Runrig)
2       Bad (by:Michael Jackson)
4       big boat (by:phish)
======================================




Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]:
```

Figure 2 - CDInventory.py run on Terminal Command Line

Summary

Assignment 6 required thinking through variable scopes as lists and strings were passed between the main procedure and class functions. After some initial difficulties with table clearing and overwriting behavior, the code ran successfully in both the IDE and Mac Terminal.