Muhammad Mannan

August 7 2021

IT FDN 110

Assignment05

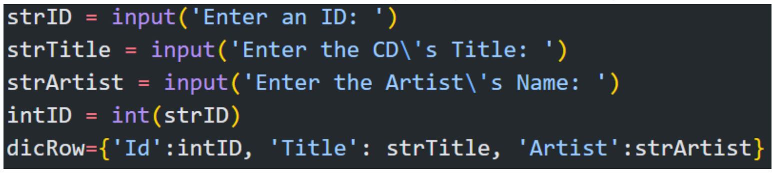
Module 05

Introduction:

The goal of this assignment was to learn about a new way of storing data through dictionaries as well as introduce us to the idea of working with other people's code as more often than not, when in the professional development environment people will have to work in teams which means learning to work on code created by other people and familiarizing yourself with being able to find solutions to a problem in ways that you do not normally use and instead pushing yourself out of your comfort zone to try and work around and with others.

Using dictionaries:

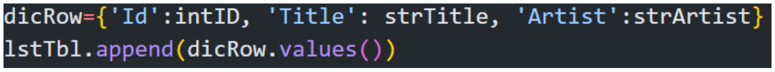
A new way of storing data that we learned about was through the use of dictionaries which replace the index and storing in sequence by storing key: value pairs. The goal of this assignment was to create a 2d table in which the inner "row" of data was a dictionary instead of a list. We can create a simple dictionary to store our values in by first creating a variable and then setting it to contain our data of keys and values (see figure 1).



```
strID = input('Enter an ID: ')
strTitle = input('Enter the CD\'s Title: ')
strArtist = input('Enter the Artist\'s Name: ')
intID = int(strID)
dicRow={'Id':intID, 'Title': strTitle, 'Artist':strArtist}
```

Figure 1-----------

However in order for us to store the necessary values into the list which will be our table to store our rows, we cannot just append a dictionary to the list. If we do this we would end up storing the key as well which we do not want. Instead we can use a function that will only store the values of our dictionary in the list (see figure 2)

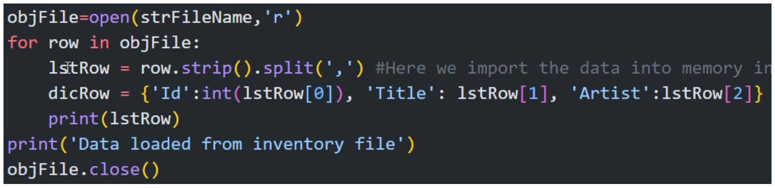


```
dicRow={'Id':intID, 'Title': strTitle, 'Artist':strArtist}
lstTbl.append(dicRow.values())
```

Figure 2-----------

Loading data:

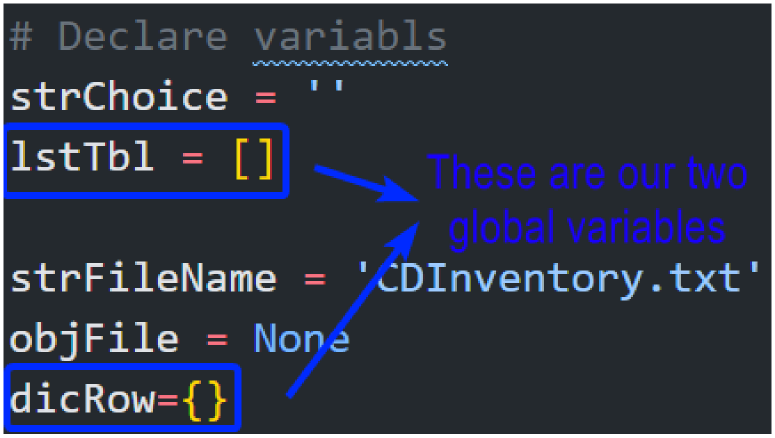
Another key component of this assignment was to be able to load data into our memory so that we could edit (or add to) previously existing data. However we cannot just simply read this data we need to be able to add it onto our list so that we can make it editable. In order to do so we can use a for loop to load our data into a list and then add the different indexes of that list to our main 2d table list (see figure 3).



```
objFile=open(strFileName,'r')
for row in objFile:
    lstRow = row.strip().split(',') #Here we import the data into memory in
    dicRow = {'Id':int(lstRow[0]), 'Title': lstRow[1], 'Artist':lstRow[2]}
    print(lstRow)
print('Data loaded from inventory file')
objFile.close()
```

Figure 3-----------

This will be the same list we manually add entries to which allows us to keep all our data in one place. However, because we are going to be loading data in, as well as manually adding data we need to create a global variable at the start of our code so that no matter where we are in our script we can always refer back to our original variable (see figure 4).

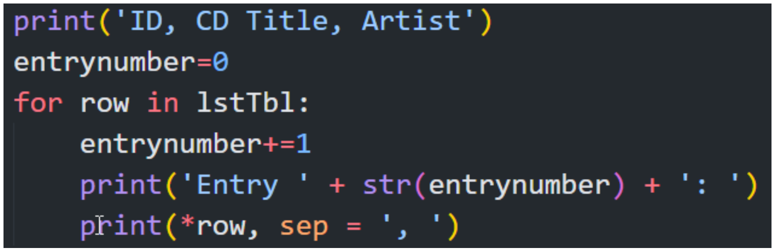


```
# Declare variabls
strChoice = ''
lstTbl = []

strFileName = 'CDInventory.txt'
objFile = None
dicRow={}
```

Figure 4-----------

Deleting an entry from memory:

The final component of this assignment was to be able to delete an entry from the data loaded into memory. This is rather simple as we can ask the user which entry they would like to remove from the inventory currently loaded into memory. When the user decides which entry they would like to remove we can delete that entry by referring to the list's index which is storing all our entries (see figure 5)

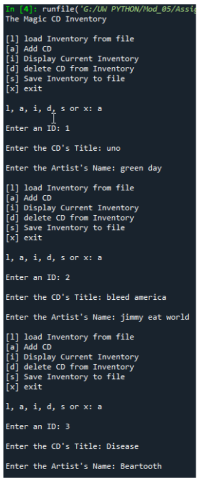


```
print('ID, CD Title, Artist')
entrynumber=0
for row in lstTbl:
    entrynumber+=1
    print('Entry ' + str(entrynumber) + ': ')
    print(*row, sep = ', ')
```

Figure 5-----------

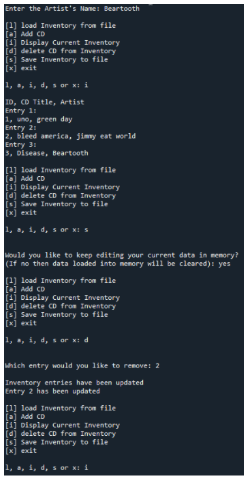
Summary:

After implementing these components here is what our program looks like running in spyder (see figure 6.1 to 6.4) and in the terminal (see figure 7.1 to 7.4)



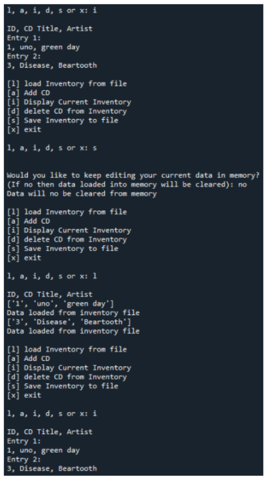
```
In [4]: runfile('G:/UW PYTHON/Mod_05/Assig
The Magic CD Inventory

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

l, a, i, d, s or x: a

Enter an ID: 1

Enter the CD's Title: uno

Enter the Artist's Name: green day

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

l, a, i, d, s or x: a

Enter an ID: 2

Enter the CD's Title: bleed america

Enter the Artist's Name: jimmy eat world

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

l, a, i, d, s or x: a

Enter an ID: 3

Enter the CD's Title: Disease

Enter the Artist's Name: Beartooth
```

Figure 6.1----------



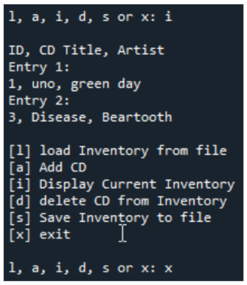
```
Enter the Artist's Name: Beartooth

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

l, a, i, d, s or x: i

ID, CD Title, Artist
Entry 1:
1, uno, green day
Entry 2:
2, bleed america, jimmy eat world
Entry 3:
3, Disease, Beartooth

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

l, a, i, d, s or x: s


Would you like to keep editing your current data in memory?
(If no then data loaded into memory will be cleared): yes

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

l, a, i, d, s or x: d


Which entry would you like to remove: 2

Inventory entries have been updated
Entry 2 has been updated

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

l, a, i, d, s or x: i
```

Figure 6.2-----------



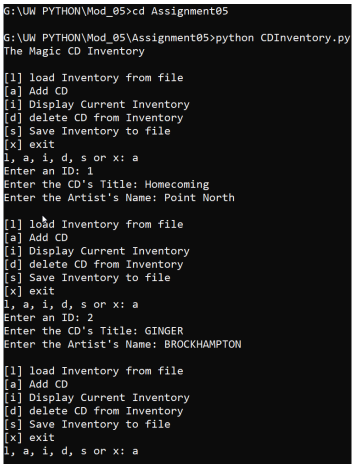
```
l, a, i, d, s or x: i

ID, CD Title, Artist
Entry 1:
1, uno, green day
Entry 2:
3, Disease, Beartooth

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

l, a, i, d, s or x: s


Would you like to keep editing your current data in memory?
(If no then data loaded into memory will be cleared): no
Data will no be cleared from memory

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

l, a, i, d, s or x: l

ID, CD Title, Artist
['1', 'uno', 'green day']
Data loaded from inventory file
['3', 'Disease', 'Beartooth']
Data loaded from inventory file

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

l, a, i, d, s or x: i

ID, CD Title, Artist
Entry 1:
1, uno, green day
Entry 2:
3, Disease, Beartooth
```

Figure 6.3-----------



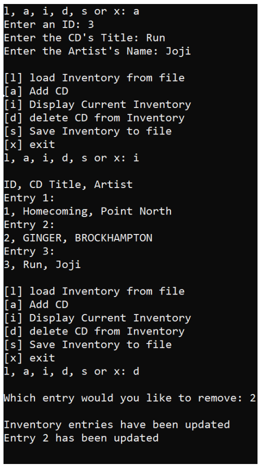
```
l, a, i, d, s or x: i

ID, CD Title, Artist
Entry 1:
1, uno, green day
Entry 2:
3, Disease, Beartooth

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

l, a, i, d, s or x: x
```

Figure 6.4-----------



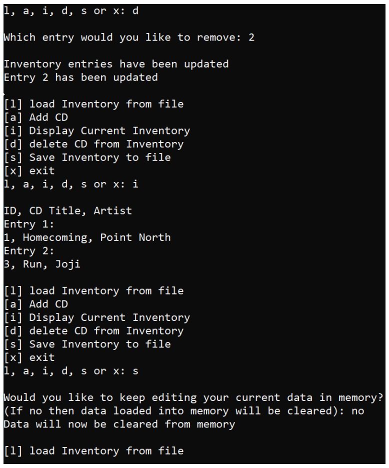
```
G:\UW PYTHON\Mod_05>cd Assignment05

G:\UW PYTHON\Mod_05\Assignment05>python CDInventory.py
The Magic CD Inventory

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a
Enter an ID: 1
Enter the CD's Title: Homecoming
Enter the Artist's Name: Point North

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a
Enter an ID: 2
Enter the CD's Title: GINGER
Enter the Artist's Name: BROCKHAMPTON

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

Figure 7.1----------



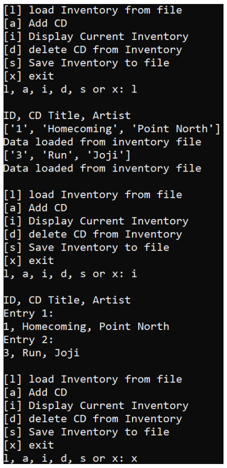
```
l, a, i, d, s or x: a
Enter an ID: 3
Enter the CD's Title: Run
Enter the Artist's Name: Joji

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

ID, CD Title, Artist
Entry 1:
1, Homecoming, Point North
Entry 2:
2, GINGER, BROCKHAMPTON
Entry 3:
3, Run, Joji

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

Which entry would you like to remove: 2

Inventory entries have been updated
Entry 2 has been updated
```

Figure 7.2-----------



```
l, a, i, d, s or x: d

Which entry would you like to remove: 2

Inventory entries have been updated
Entry 2 has been updated

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

ID, CD Title, Artist
Entry 1:
1, Homecoming, Point North
Entry 2:
3, Run, Joji

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

Would you like to keep editing your current data in memory?
(If no then data loaded into memory will be cleared): no
Data will now be cleared from memory

[l] load Inventory from file
```

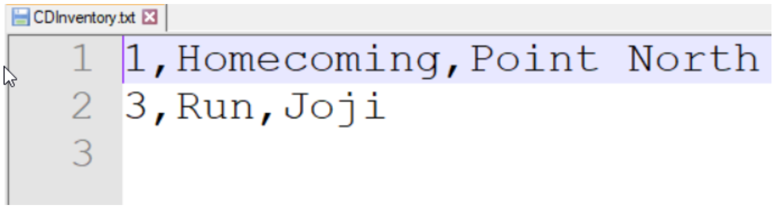
Figure 7.3-----------



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

ID, CD Title, Artist
['1', 'Homecoming', 'Point North']
Data loaded from inventory file
['3', 'Run', 'Joji']
Data loaded from inventory file

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

ID, CD Title, Artist
Entry 1:
1, Homecoming, Point North
Entry 2:
3, Run, Joji

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

Figure 7.4---------

and here is the output produced by running the script in the terminal as the text file (see figure 8).



```
1,Homecoming,Point North
3,Run,Joji

```

Figure 8-----------

After working on this project it is clear to see that while learning about dictionaries was the foundational focus on adding another tool to our skillset the bigger picture was learning and experience what it would be like to work in a more professional environment in which people have to work back and forth with each other, working around and adding on to each other's code. Sometimes we may receive a piece of code that someone may have worked on before that needs to be completed so it is important for us to sit down and understand what is happening in the script before we make any changes and learn to implement our code around that so that we do not break anything or ruin the code's integrity. This also allows us to code and problem solve in ways that we may not have thought of before and exposes us to situations which help us develop a more versatile skill set as programmers.