Artem Aleshin
August 7th, 2022
IT FDN 110 A
Assignment 05

Assignment 05 Knowledge Document

Introduction

The Knowledge Document covers Points of Interest from Module 05, the work done as part of the fourth set of Laboratories and the completed Assignment. As part of working through the tasks, I learned more about Lists, learned about Dictionaries, GitHub, script templates, Separation of Concerns, and error handling.

Module 05 Points of Interest

A topic that I found interesting from Module 05 is error handling. It appears that Python handles error handling in functions in a very intuitive way. I also like how it paves the way for getting rid of complex error messages. I am also interested in how Python would go about handling n-dimensional tensors and arrays. Lastly, I think that the examples that the book is providing have started to become quite interesting as the games that are laid out have become more complex.

Module 05 Laboratories

LAB 05_A

1. The program must ask the user to enter an option. Then the program must be able to receive data, write data, read data, and display data.
2. The lab was saved appropriately.
3. The header was augmented.
4. The TODOs were replaced and the script is now as follows:

```python
#------------------------------------------#
# Title: Lab05_A.py
# Desc: Lab05-A script
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# aaleshin, 2022-Aug-07, Modified File to execute commands
#------------------------------------------#


# Declare variables

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
lstRow = []  # list of data row
readRow = []    # list of read data.
readTbl = []    # list of lists to hold read data.
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

file = open(strFileName, 'a+')  # Creates CDInventory.txt if it does not exit.
file.close()


# Get user Input
print('Write or Read file data.')
while True:
    print('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
    print('[d] display data\n[exit] to quit')
    print()
    strChoice = input('a, w, r, d, or exit: ').lower()  # convert choice to lower case at time of input
    print()

    if strChoice == 'exit':
        break
    if strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
        # Add data to list in memory
        Artist = input('Enter Artist Name: ')
        Title = input('Enter CD Title: ')
        lstRow = [Artist, Title]
        lstTbl.append(lstRow)
    elif strChoice == 'w':
        # List to File
        for row in lstTbl:
            objFile = open(strFileName, 'a')
            strRow = row[0] + ', ' + row[1]
            objFile.write(strRow + '\n')
            objFile.close()
    elif strChoice == 'r':
        # File to print
        objFile = open(strFileName, 'r')
        for row in objFile:
            readRow = row.strip().split(',')
            readTbl.append(readRow)
        objFile.close()
    elif strChoice == 'd':
        # Display data
        print('Artist, Title')
        for row in readTbl:
            print(*row, sep = ', ')
    else:
        print('Please choose either a, w, r or exit!')
```

Figure 1: Lab 05_A Script.

5. The code starts off with declaring the variables that will be used throughout the script. It then runs two lines of code that create a text file so that the code does not break while running if the user chooses to read the file and no file exists. The user is then presented with a list of options: a, w, r, d, and exit, which correspond to adding data, writing data, reading data, displaying data, or exiting the program. If the user selects exit, the program breaks. If the user chooses to append to the file, they are prompted to enter the artist's name and title of the album, which then get put into a list which gets appended to the 2D list. If the user chooses to write, the text file is opened and the rows of the 2D list get appended to the file until no rows remain. The file then closes. If the user chooses to read, the file is once again opened and read row by row in a loop. The individual rows get broken down into individual elements which are then saved to strings which get appended to a table for later use. The file is then closed. If the user chooses to display the data, every row in the table is printed, with the individual elements of a row being separated by a comma. Lastly, if the user enters something that does not match any of the options, they are prompted with entering an option that exists.
   The script running produces the following output:

```
In [1]: runfile('C:/_FDProgramming/Assignment_05/Lab05_A.py', wdir='C:/_FDProgramming/Assignment_05')
Write or Read file data.

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: a

Enter Artist Name: Runrig
Enter CD Title: The Big Wheel

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: a

Enter Artist Name: Michael Jackson
Enter CD Title: Bad

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit
```

```
a, w, r, d, or exit: w


[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: r


[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: d

Artist, Title
Runrig,  The Big Wheel
Michael Jackson,  Bad

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: exit
```

Figure 2: Lab 05-A Output.

LAB 05-B

1. A copy of LAB05-A was made.
2. Inner lists were replaced with dictionaries. The updated script is as follows:

```python
#------------------------------------------#
# Title: Lab05_B.py
# Desc: Lab05-B script
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# aaleshin, 2022-Aug-07, Modified File to execute commands
# aaleshin, 2022-Aug-07, Modified File to use dictionaries instead of lists
#------------------------------------------#


# Declare variables

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # dictionary of data row
readRow = []    # list of read data.
readTbl = []    # list of lists to hold read data.
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

file = open(strFileName, 'a+')  # Creates CDInventory.txt if it does not exit.
file.close()

# Get user Input
print('Write or Read file data.')
while True:
    print('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
    print('[d] display data\n[exit] to quit')
    print()
    strChoice = input('a, w, r, d, or exit: ').lower()  # convert choice to lower case at time of input
    print()

    if strChoice == 'exit':
        break
    if strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
        # Add data to list in memory
        Artist = input('Enter Artist Name: ')
        Title = input('Enter CD Title: ')
        dicRow = {'Artist': Artist, 'Title': Title}
        lstTbl.append(dicRow)
    elif strChoice == 'w':
        # List to File
        for row in lstTbl:
            objFile = open(strFileName, 'a')
            strRow = row['Artist'] + ', ' + row['Title']
            objFile.write(strRow + '\n')
            objFile.close()
    elif strChoice == 'r':
        # File to print
        objFile = open(strFileName, 'r')
        for row in objFile:
            readRow = row.strip().split(',')
            dicRead = {'Artst': readRow[0], 'Title': readRow[1]}
            readTbl.append(dicRead)
        objFile.close()
    elif strChoice == 'd':
        # Display data
        print('Artist, Title')
        for row in readTbl:
            print(*row.values(), sep = ', ')
    else:
        print('Please choose either a, w, r or exit!')
```

Figure 3: Lab 05-B Script.

3. The code works in a very similar fashion to code in LAB 05-A. However, there are several key differences. If the user selects to add data, the data gets saved to a dictionary instead of a list. The dictionary then gets appended to the 2D table. When the user chooses to write the data to a file, the rows call on the artist associated with the Artist key and the title associated with the Title key and separate them by a comma. When the user chooses to read the file, the read rows get added back to a dictionary as opposed to a list. Lastly, if the user decides to display the data, the data gets retrieved from the read table, and the values associated with the keys get displayed row by row. The output of the script is as follows:

```
In [3]: runfile('C:/_FDProgramming/Assignment_05/Lab05_B.py', wdir='C:/_FDProgramming/Assignment_05')
Write or Read file data.

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: a

Enter Artist Name: Runrig
Enter CD Title: The Big Wheel

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: a

Enter Artist Name: Michael Jackson
Enter CD Title: Bad

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: w


[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: r
```

```
[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit: d

Artist, Title
Runrig,  The Big Wheel
Michael Jackson,  Bad

[a] add data to list
[w] to write data to file
[r] to read data from file
[d] display data
[exit] to quit

a, w, r, d, or exit:
```

Figure 4: Lab 05-B Output.

4. It would be better if the output did not show the keys. This can be avoided by calling on the values of the rows and separating them by commas.

Assignment 05

1. Folder called Assignment05 was created.
2. The code was modified and is as follows:

```python
#------------------------------------------#
# Title: CDInventory.py
# Desc: Assignment 05 Script
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# aaleshin, 2022-Aug-07, Modified File to Use Dictionaries, Load Data and Delete Data
#------------------------------------------#

# Declare variables

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
dicRow = {}  # dictionary of data row
lstRow = []  # list of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

file = open(strFileName, 'a+')  # Creates CDInventory.txt if it does not exit.
file.close()

# Get user Input
print('The Magic CD Inventory\n')
while True:
    # 1. Display menu allowing the user to choose:
    print('[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')
    strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case at time of input
    print()

    if strChoice == 'x':
        # 5. Exit the program if the user chooses so
        break
    if strChoice == 'l':
        lstTbl.clear()
        objFile = open(strFileName, 'r')
        for row in objFile:
            lstRow = row.strip().split(',')
            dicRow = {'ID': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}
            lstTbl.append(dicRow)
        objFile.close()
    elif strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
        # 2. Add data to the table (2d-list) each time the user wants to add data
        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}
        lstTbl.append(dicRow)
    elif strChoice == 'i':
        # 3. Display the current data to the user each time the user wants to display the data
        print('ID, CD Title, Artist')
        for row in lstTbl:
            print(*row.values(), sep = ', ')
        lstTbl.clear()
    elif strChoice == 'd':
        delID = input('Enter an ID to delete: ')
        lstTbl.clear()
        objFile = open(strFileName, 'r')
        for row in objFile:
            lstRow = row.strip().split(',')
            dicRow = {'ID': lstRow[0], 'Title': lstRow[1], 'Artist': lstRow[2]}
            lstTbl.append(dicRow)
        objFile.close()
        for dic in lstTbl:
            if delID in dic['ID']:
                lstTbl.remove(dic)
        objFile = open(strFileName, 'w')
        for row in lstTbl:
            strRow = ''
            for item in row.values():
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
            objFile.write(strRow)
        objFile.close()
        lstTbl.clear()
    elif strChoice == 's':
        # 4. Save the data to a text file CDInventory.txt if the user chooses so
        objFile = open(strFileName, 'a')
        for row in lstTbl:
            strRow = ''
            for item in row.values():
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
            objFile.write(strRow)
        objFile.close()
        lstTbl.clear()
    else:
        print('Please choose either l, a, i, d, s or x!')
```

Figure 5: Assignment 05 Script.

3. The script running is as follows:

```
In [1]: runfile('C:/_FDProgramming/Assignment05/CDInventory.py', wdir='C:/_FDProgramming/Assignment05')
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: l

[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: The Big Wheel
Enter the Artist's Name: Runrig
[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: Bad
Enter the Artist's Name: Michael Jackson
[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

[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

[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
1, The Big Wheel, Runrig
2, Bad, Michael Jackson
```

```
[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

Enter an ID to delete: 2
[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

[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
1, The Big Wheel, Runrig
[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: Assignment 05 Spyder Output.

```
(base) C:\_FDProgramming\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: l

[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
1, The Big Wheel, Runrig
[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: Bad
Enter the Artist's Name: Michael Jackson
[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

[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

Enter an ID to delete: 1
[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

[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
2, Bad, Michael Jackson
[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


(base) C:\_FDProgramming\Assignment05>
```

Figure 7: Assignment 05 Anaconda Prompt Output.

4. All options were used and were verified to be working.
5. Like Lab 05-B, the assignment uses dictionaries instead of lists for the inner rows of the 2D List. The code first declares variables to be used later. A file is created if it does not exist by using the append option. The code prints out the title of the program as well as the user options. The user is allowed to enter an option. If the user chooses to exit, the program breaks. If the user chooses to load the file, the file is opened in read only mode, the rows get split and the individual elements of the rows get paired with dictionary keys. The rows then get appended to a 2D list and the file is then closed. If a user chooses to add another line to the file, they are prompted to type in an ID, CD Title, and Artist name, all three of which then get assigned to a dictionary which is then appended to the 2D list. If the user selects to display the information in the 2D list, the data is read line by line from the list and gets unpacked row by row where the elements are separated by a comma followed by a space. If the user selects to delete a row, they are prompted to enter the ID associated with the row they want to delete. The 2D list is cleared to remove any existing data, the file is opened for reading, and the contents get appended to the 2D list, like how when the file was read when it was loaded. A loop then searches for the ID the user selected and deletes the rows if the ID matches the dictionary value. The file then gets reopened and the updated 2D table then gets written to it. If the user chooses to save data to a file, the data that the user entered gets appended to the file similarly to how it was done for the Laboratories. Lastly, if the user enters an invalid option, they get reminded about what the valid inputs are and the options get presented to the user once more.

Summary

The document covers the Points of Interest, the Laboratories, and the Assignment from Module 05. By completing Module 05, I learned about dictionaries, lists, GitHub, error handling, script templates, and Separation of Concerns.