Jenn Wall

27 Feb, 2022

Foundations of Programming: Python

Assignment 05

Dictionaries

Introduction

In this assignment we discusses the differences between lists and dictionaries. We also covered how to use different functions and methods specific to dictionaries. We used this information to expand on our CD Inventory script from last week to use dictionaries in place of lists. We also explored what Git and GitHub are.

Dictionary vs List

A dictionary is similar to a list in that it is a set of data assigned to a single variable. The difference is that dictionaries are set up with a key:value construct. Dictionaries are also created using curly brackets instead of square brackets or parentheses. For example,

```
dicRow = {'a': 1, 'b': 200, 'c': 3000}
```

Dictionaries are also mutable. This means that if you change the value of one of the keys, it changes the original dictionary. It does not create a copy with the new value.

Read Data

For assignment 05 we went through how to read data from a text file into either a list or a dictionary. Reading data from a file is fairly straightforward. To do this, we call the file from which we wish to read (make sure to open in read mode). Then we iterate through each row in the file and assign it to a list or dictionary. Once all rows have been read, we close the file.

Separation of Concerns

In order to make our scripts more organized, we structure them so that related items are grouped together. This makes reading the code much easier. For our purposes, we will be organizing our sections by data, processing, and then presentation. In the data section we state any variables and/or functions. Then we use this data in the processing section to perform any actions or computations needed. Once we have processed our date, we present it to the user.

Another way to organize our code is to define functions. Instead of running computations when we ask for user input, we can define simple arithmetic functions in our data section, then later

call these functions to run in our presentation section. Doing it this way allows us to keep our code to one data section.

We can also set up a script template to add header data automatically to each new file we create. We do this for efficiency sake as it saves us the time of typing it out or copy/pasting from previous files.

Error Handling

Error handling allows us to anticipate failures we might encounter and handle them gracefully so that we are not exited out of our code. This allows us to continue running our program by displaying an error to the user and asking them to try again.

GitHub

GitHub is a massive source-code host. It allows users to store and collaborate on their code from anywhere with internet access. GitHub has features like versioning, bug tracking, and feature requests. It even has it's own mascot - the Octocat.

Create a Script

For assignment 05 we were tasked with expanding on our CD Inventory script from last week. We needed to convert our inner data types to dictionaries. We also needed to add functionality to read in data from a file and delete an entry from our table.

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

    # 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':
            objFile = open(strFileName, 'r')
            for row in objFile:
                lstRow = row.strip().split(',')
                dicRow = {'ID': int(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 = ', ')
            print('\n\n')

        elif strChoice == 'd':
            deleteID = int(input('Enter the ID of the CD you want to delete: '))
            for row in lstTbl:
                if row['ID'] == deleteID:
                    print(row)
                    del row

        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()

        else:
            print('Please choose either l, a, i, d, s or x!')
```

Figure 1. CDInventory script

Once the script is written, we run through the options.

```
[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: Bad

Enter the Artist's Name: Michael Jackson
```

Figure 2. Add data to 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: i

ID, CD Title, Artist
1, Bad, Michael Jackson
2, Thriller, Michael Jackson
```

Figure 3. Read data

```
[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 the ID of the CD you want to delete: 2
{'ID': 2, 'Title': 'Thriller', 'Artist': 'Michael Jackson'}
```

Figure 4. Delete data

CDInventory.txt

~/Library/Mobile Documents/com~apple~CloudDocs/Class/Mod_05/Assignment05/CDInventory

```
1,Bad,Michael Jackson
2,Thriller,Michael Jackson
```

Figure 5. Data saved to file

Summary

To summarize, we went through how and when to use dictionaries and list, how to read data from a file, how to use GitHub, and how to organize our code. I found the majority of this assignment to be straightforward, but I struggled with how to delete a dictionary from a list. I was unable to figure this out, and my script still does not delete an entry.