Mark Mathis
Feb. 14th 2021
Foundations of Programming, Python
Assignment 05

Introduction:

In this assignment I will attempt to outline the steps I used to create a program in the Python language that helps a user to create a data dictionary that organizes a CD (Compact Disk) library. There were several key components to this assignment that were completed in order to demonstrate my knowledge of Python, those components were:

1) Display a menu that will help the user perform a task.
2) Add user input Data to a Dictionary
3) Display user input data.
4) Write input text file to dictionary.
5) Delete input in dictionary

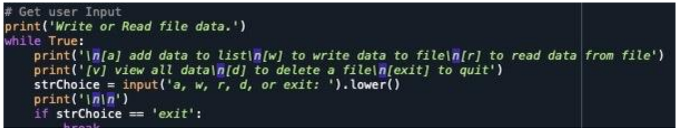
Drafting the Code:

OVERVIEW:

I began this assignment by reviewing the class materials, textbook, web links as well as, my own web research to attempt to grasp an understanding of how Python turns user input into a working dictionary. I build on previous lessons, adding code that performs the specific task. This exercise utilizes loops via the "if" "else" & "elif" commands that offer a way for a user to select different functions of the program to create a structured data dictionary from multiple data sources.

CREATING A LIST:

I begin this code with the basic directory suture provided in the class materials and modified the code to accommodate a few more options. The menu is established through print statement that lay out the menu options. The user makes a selection through the input command and loops through the if, elif and else statements.



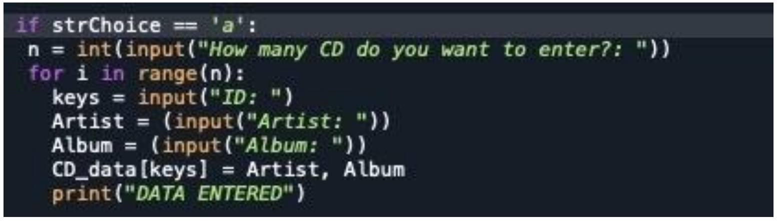
```
# Get user Input
print('Write or Read file data.')
while True:
    print('\n[a] add data to list\n[w] to write data to file\n[r] to read data from file')
    print('[v] view all data\n[d] to delete a file\n[exit] to quit')
    strChoice = input('a, w, r, d, or exit: ').lower()
    print('\n\n')
    if strChoice == 'exit':
        break
```

figure 1.1

The user input is constrained through the ".lower()" command which forces proper case of the selected menu letter.

Option ‘a’: User input

After several iteration of this code, I settled on the below option, for the main reason of its compactness. These lines of code accomplish several tasks and demonstrate the versatility of Python. One of the first problems I had when creating a dictionary was the problem of getting the script to iterate. I solved this problem by utilizing the “range()” function. Through range I was able to get the script to iterate through entry after entry, by asking the user to input how many CDs a user would like to enter. The range () function then loops the user entry for as many iterations as the user specifies. The other problem I had to overcome was how to assign the proper key to the dictionary entry. The simplest answer was through “dictionary.keys().”In two simple lines I was able to add a key as a variable I specified and enter all the user input data into the dictionary under that key.

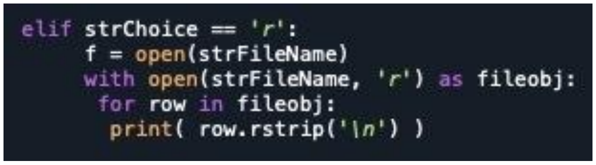


```
if strChoice == 'a':
 n = int(input("How many CD do you want to enter?: "))
 for i in range(n):
   keys = input("ID: ")
   Artist = (input("Artist: "))
   Album = (input("Album: "))
   CD_data[keys] = Artist, Album
   print("DATA ENTERED")
```

Figure 1.2

Option ‘r’: read data

The read data option is just a way of reading the text file created in the last assignment. The way that I understand this assignment is we are imagining a person who would want to add CDs to their collection individually but would also like to port their current collection from one app to another, I envisioned the read function as a way of simple checking that the data in the .txt file is correct. In order to make the data easily readable it was important to strip out the character that tells the computer to insert a new line. While necessary to perform certain functions, these characters make for a very messy presentation. By utilizing the “fileobj” command, this command allows a programmer to access the file and “strip” each row of the text data of the new line character.

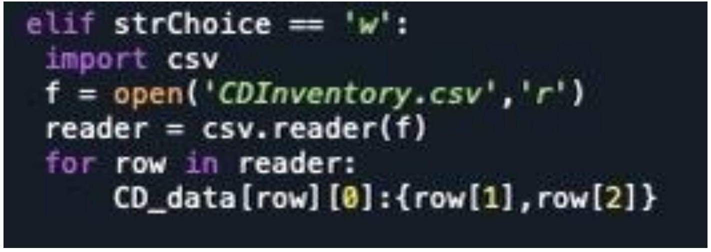


```
elif strChoice == 'r':
     f = open(strFileName)
     with open(strFileName, 'r') as fileobj:
      for row in fileobj:
       print( row.rstrip('\n') )
```

figure 1.3

Option ‘w’: write data

figure 1.4



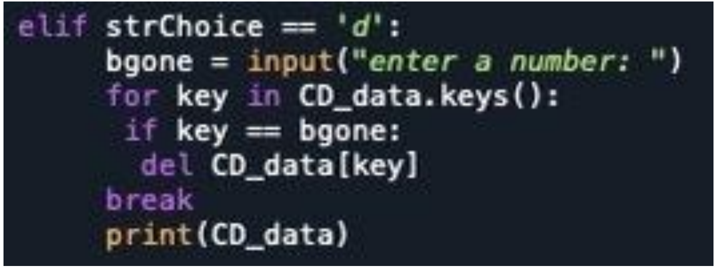
```
elif strChoice == 'w':
 import csv
 f = open('CDInventory.csv','r')
 reader = csv.reader(f)
 for row in reader:
     CD_data[row][0]:{row[1],row[2]}
```

The write option is to take the text file and write it to the dictionary, the trick being the format of the text file and the establishment of Identical format to the user entry and the establishment of a key. I attempted to accomplish this task by utilizing “import csv” and “reader”. The reader command allows the rows to be read and inserted into the dictionary with the key assignment that I wanted.

Option ‘d’: write data

The delete option is to delete a record by it’s key. The statement is relatively simple, asking the user to input an ID key to delete and executing the delete (del) command in the dictionary.

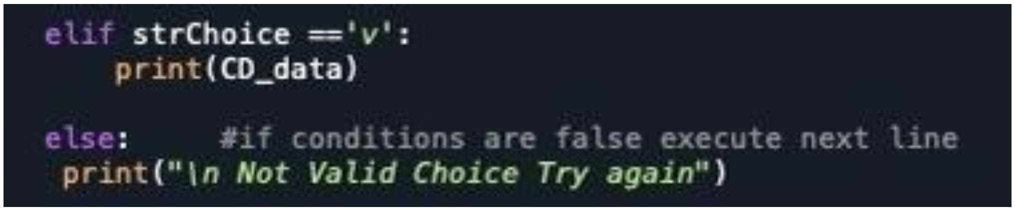
figure 1.5



```
elif strChoice == 'd':
     bgone = input("enter a number: ")
     for key in CD_data.keys():
      if key == bgone:
       del CD_data[key]
     break
     print(CD_data)
```

Option ‘v’: write data & else:

The ending functions of the code are simply the option to view the data in the dictionary and the invalid choice loop.



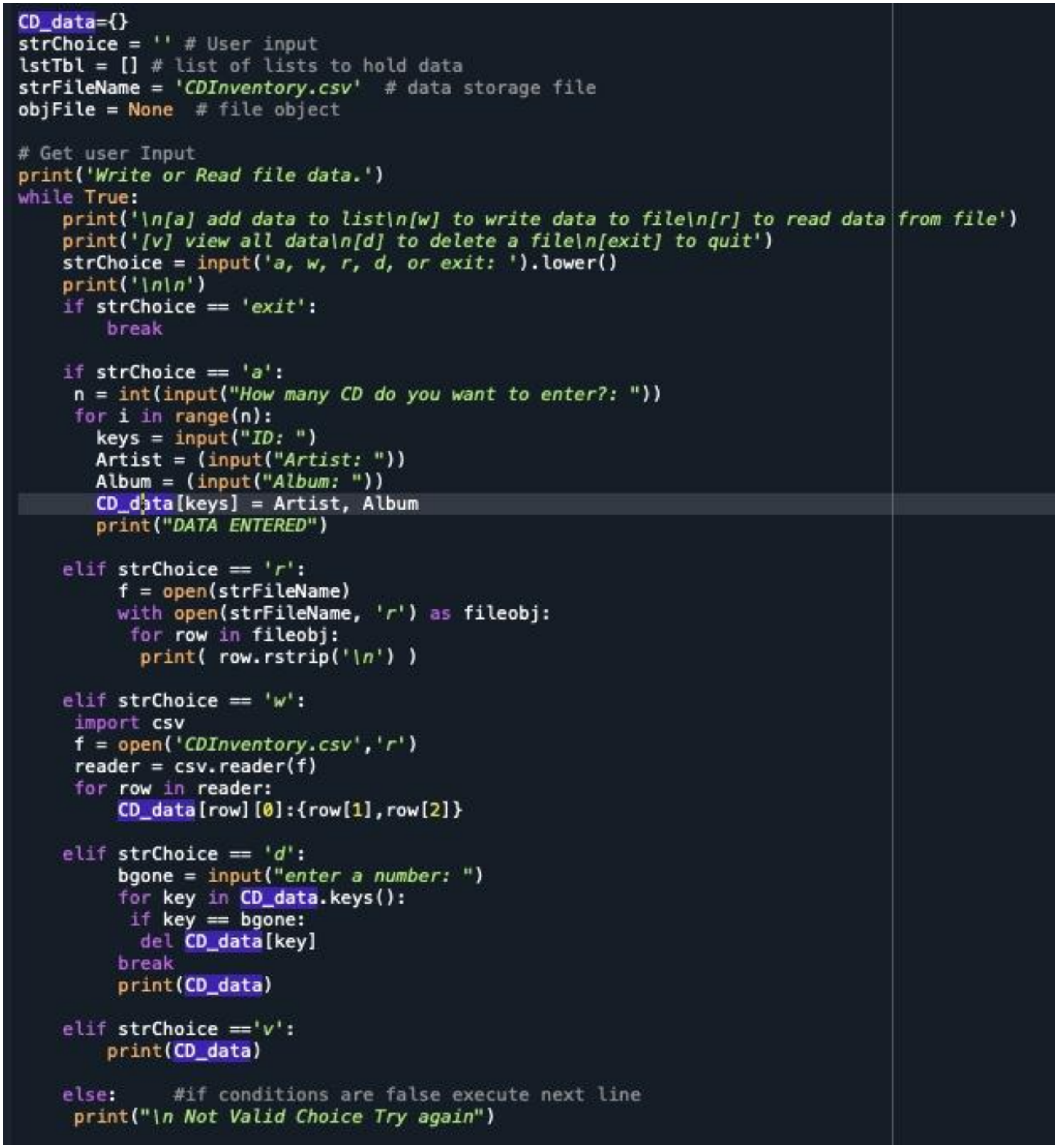
```
 elif strChoice =='v':
     print(CD_data)

 else:     #if conditions are false execute next line
  print("\n Not Valid Choice Try again")
```

figure 1.6

COMPLETED CODE:

Figure 2.1

The above code that completes this assignment was produced with help from the course materials and the web.



```
CD_data={}
strChoice = '' # User input
lstTbl = [] # list of lists to hold data
strFileName = 'CDInventory.csv'  # data storage file
objFile = None  # file object

# 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('[v] view all data\n[d] to delete a file\n[exit] to quit')
    strChoice = input('a, w, r, d, or exit: ').lower()
    print('\n\n')
    if strChoice == 'exit':
        break

    if strChoice == 'a':
     n = int(input("How many CD do you want to enter?: "))
     for i in range(n):
       keys = input("ID: ")
       Artist = (input("Artist: "))
       Album = (input("Album: "))
       CD_data[keys] = Artist, Album
       print("DATA ENTERED")

    elif strChoice == 'r':
         f = open(strFileName)
         with open(strFileName, 'r') as fileobj:
          for row in fileobj:
           print( row.rstrip('\n') )

    elif strChoice == 'w':
     import csv
     f = open('CDInventory.csv','r')
     reader = csv.reader(f)
     for row in reader:
         CD_data[row][0]:{row[1],row[2]}

    elif strChoice == 'd':
         bgone = input("enter a number: ")
         for key in CD_data.keys():
          if key == bgone:
           del CD_data[key]
         break
         print(CD_data)

    elif strChoice =='v':
        print(CD_data)

    else:     #if conditions are false execute next line
     print("\n Not Valid Choice Try again")
```