James

3/06/2022

IT FDN 110 B: Introduction to Programming Python

Assignment 06

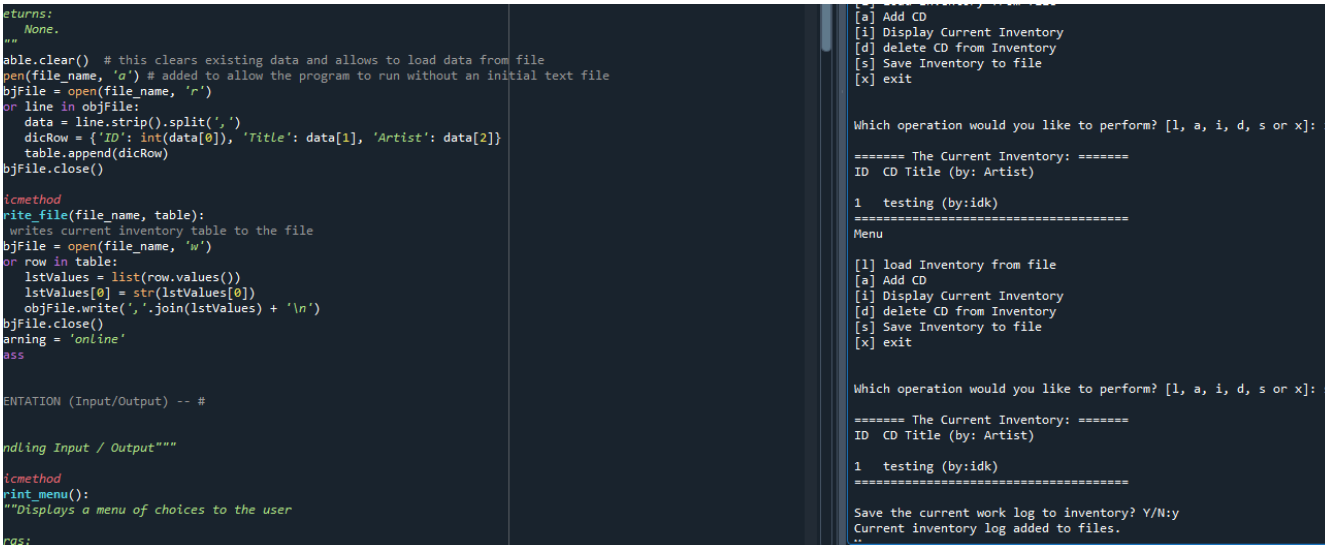
- Overview

This week, we shift our focus slightly to setting up classes and functions to handle the cd menu from last week.

- Creating Functions

Confusion with the prompt aside (I interpret adding to a script as adding additional lines, not replacing current lines) the functions themselves were far less painful than last week. In part because what changed was the formatting while the actions are all things we've done before, the main things to watch out for is keeping tabs on your input variables and making sure that our function is compatible. I once again messed up on using [] for calling items from a list, but this time I caught my mistake early instead of far, far, later.

[Image of Functions in Action]



```
eturns:
   None.
""
able.clear()  # this clears existing data and allows to load data from file
pen(file_name, 'a') # added to allow the program to run without an initial text file
bjFile = open(file_name, 'r')
or line in objFile:
   data = line.strip().split(',')
   dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
   table.append(dicRow)
bjFile.close()

icmethod
rite_file(file_name, table):
 writes current inventory table to the file
bjFile = open(file_name, 'w')
or row in table:
   lstValues = list(row.values())
   lstValues[0] = str(lstValues[0])
   objFile.write(','.join(lstValues) + '\n')
bjFile.close()
arning = 'online'
ass


ENTATION (Input/Output) -- #


ndling Input / Output"""

icmethod
rint_menu():
""Displays a menu of choices to the user

rgs:
```

```
[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]:

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

1   testing (by:idk)
======================================
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]:

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

1   testing (by:idk)
======================================

Save the current work log to inventory? Y/N:y
Current inventory log added to files.
```

- Testing the Program

Here’s where things got a little hairy. My old test text files were not compatible with the script handed to us to work on, as it for some reason had ID Song Artist for its first line causing issues with the int(data[0]). While I could have just deleted this line, I decided to delete the entire old text file, and found a critical error outside the todo sections of the file given to us. If there is no initial text file, it fails to run as it attempts to read the file at the beginning and fails to. I added an append file to create the text document before reading it.

- Summary

While there were still a few hiccups, compared to last week where it was a struggle to figure out how to even get everything to work, this week was a lot more relaxed, as the functionality was already cracked last week and instead we only had to really work on formatting. I do wish the instructions were clearer, as without clarification the assignment and todos taken literally would imply we should have numerous actions done twice over. Obviously massive amounts of duplicates wasn’t the intent, but this led to the actual intent being unclear before office hours.

- Appendix

*(If you have advice on how to format google docs such that the appendix table doesn’t sometimes wrap to the next page, I’d appreciate it)*

```python
#--------------------------------------#
# Title: Assignment06_Starter.py
# Desc: Working with classes and functions.
# Change Log: (James Miller, 3/4/22, Added additional IO/Processing functions)
# (also converted menu actions to function calls, and fixed issues with new txt files)
# DBiesinger, 2030-Jan-01, Created File
#--------------------------------------#

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


# -- PROCESSING -- #
class DataProcessor:
    def add_single(vlist):
        # takes the input song and adds it to the current work log
        dicRow = {'ID': vlist[0], 'Title': vlist[1], 'Artist': vlist[2]}
        lstTbl.append(dicRow)
        print('The song '+ vlist[1] +' has been added to the current log')
        return
        pass

    def eliminate(info):
        intRowNr = -1
        blnCDRemoved = False
        for row in lstTbl:
            intRowNr += 1
            if row['ID'] == info:
                del lstTbl[intRowNr]
                blnCDRemoved = True
                break
        if blnCDRemoved:
            print('The CD was removed')
        else:
            print('Could not find this CD!')
        pass

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) table 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
        open(file_name, 'a') # added to allow the program to run without an initial text 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):
        # writes current inventory table to the file
        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()
        warning = 'online'
        pass


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

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

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

        Args:
            None.

        Returns:
            None.
        """
```

```python

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

    def def_cd():
        strID = int(input('Enter ID: ').strip())
        strTitle = input('What is the CD\'s title? ').strip()
        stArtist = input('What is the Artist\'s name? ').strip()
        latestcd = [strID,strTitle,stArtist]
        return latestcd
    def mark():
        target = int(input('Enter the id number you wish to delete'))
        return target
    def confirm_save():
        flag = 'no'
        if warning == 'online':
            print('You have already saved to file this session,\ncontinuing may cause duplicate entrees')
        confirm = input('Save the current work log to inventory? Y/N:').strip().lower()
        if confirm == 'y':
            flag = 'yes'
        elif confirm == 'yes':
            flag = 'yes'
        return flag


# 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('reloading...')
            FileProcessor.read_file(strFileName, lstTbl)
            IO.show_inventory(lstTbl)
        else:
            input('canceling... Inventory data NOT reloaded. Press [ENTER] to continue to the menu.')
            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, adds it, then shows the new table
        thecd = IO.def_cd()
        DataProcessor.add_single(thecd)
        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':
            # 3.5.1.1 display Inventory to user
            IO.show_inventory(lstTbl)
            # 3.5.1.2 ask user which ID to remove
            target = IO.mark()
            DataProcessor.eliminate(target)
            # 3.5.2 search thru table and delete CD
            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 = IO.confirm_save()
            # 3.6.2 Process choice
            if strYesNo == 'yes':
                FileProcessor.write_file(strFileName, lstTbl)
                print('Current inventory log added to files.')
            else:
                input('The inventory was NOT saved to file. Press [ENTER] to return to the menu.')
            continue  # start loop back at top.
        # 3.7 catch-all should not be possible, as user choice gets vetted in IO, but to be safe:
        else:
            print('General Error')
```