Graham Byron

8 March 2022

IT FDN 110: Introduction to Programming (Python)

Assignment 06

Assignment 06: Functions and Classes

Intro

Assignment 06 centered around the use of functions and classes, and their different use within a script. Further, variable scope and docstring were introduced as well. Lastly, we again furthered our familiarization with GitHub by posting another repository. The emphasis of this section revolved around the use of functions and classes. Several examples were given of functions and how they can be used to help keep code clean. Further, many cases were given to illustrate how functions can be used in many different ways. This included parameters, which allow the coder to pass in values for processing. This continued with classes, which provide a way of grouping functions together, further cleaning up a script. Afterwards, I was tasked with altering code, following several TODO's which involved moving chunks of code into specific areas and then calling upon them later on in the script.

Topic

Functions made up the majority of this module, and many different aspects were discussed. Functions can be used in conjunction with parameters, which allows passing values in for processing. There are no limits on the amount of parameters that can be passed into a function! Continuing on, return values work in conjunction with functions and can either be explicitly printed or assigned to a variable. Further, there doesn't have to be only one return value. If there are multiple, however, it is necessary to group the values into a collection and return the collection.

Going further with functions, arguments could either be positional or named. This means that the writer could either explicitly fill the named parameter with the argument or intentionally leave the parameter unnamed. This assigns the parameter in the same sequence of how the parameters are defined in the function. Further, the writer could set a default value for a parameter, this means that the default value will be used if there is not a different parameter explicitly stated in the function.

Another important topic that was discussed was the difference between local and global variables as well as variable scope. The difference is whether or not the variable can affect the entire script or just the function it is set within. If the variable is local, then it only has effect within the set function that it is within. If it's global, then it's scope goes beyond just the specific function.

Another formatting tip that was discussed was Doc Strings. These are used to include an explanation at the beginning of the function and are common practice. Further, this string is able to be saved within the 'Help' section in Spyder and can be useful references.

Finally, classes are discussed and their ability to clean up code. They are a way of grouping functions together, allowing for more than one action to be called upon and done under the same umbrella. After this was discussed, I was tasked with altering some CDInventory code to group functions under classes. These classes were then called upon to execute the desired outcome based on user input. Below is the output within Spyder:

Spyder Output of Assignment

```
In [14]: runfile('/Users/grahambyron/Desktop/CDInventory.py', wdir='/Users/
grahambyron/Desktop')
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]: a


Enter ID: 1

Enter the CD's title. Shadows

Enter the Artist's name. Cannons
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Shadows (by:Cannons)
===================================
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]: a


Enter ID: 2

Enter the CD's title. Talk Talk

Enter the Artist's name. Bleachers
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Shadows (by:Cannons)
2   Talk Talk (by:Bleachers)
===================================
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]: i

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

1   Shadows (by:Cannons)
2   Talk Talk (by:Bleachers)
===================================
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]: s

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

1   Shadows (by:Cannons)
2   Talk Talk (by:Bleachers)
===================================

Save this inventory to file? [y/n] y
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]: d

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

1   Shadows (by:Cannons)
2   Talk Talk (by:Bleachers)
===================================

Which ID would you like to delete? 1
The CD was removed
======= The Current Inventory: =======
ID  CD Title (by: Artist)

2   Talk Talk (by:Bleachers)
===================================
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]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loade
from file.

type 'yes' to continue and reload from file. otherwise reload will be canceledyes
reloading...
======= The Current Inventory: =======
ID  CD Title (by: Artist)

1   Shadows (by:Cannons)
2   Talk Talk (by:Bleachers)
===================================
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]: x


In [15]:
```

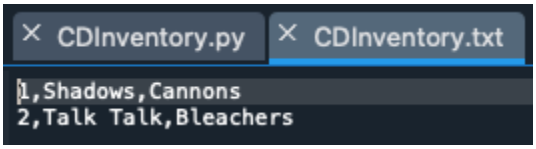
Terminal Output

```
(base) grahambyron@FVFF8CE1Q6L4 Python Scripts % python /Users/grahambyro
n/Desktop/Python\ Scripts/Assignment06/CDInventory.py
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]: a

Enter ID: 3
Enter the CD's title. Dreamland
Enter the Artist's name. Glass Animals
======= The Current Inventory: =======
ID      CD Title (by: Artist)

3       Dreamland (by:Glass Animals)
=======================================
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]: a

Enter ID: 2
Enter the CD's title. Melodrama
Enter the Artist's name. Lorde
======= The Current Inventory: =======
ID      CD Title (by: Artist)

3       Dreamland (by:Glass Animals)
2       Melodrama (by:Lorde)
=======================================
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]: s
```

```
Which operation would you like to perform? [l, a, i, d, s or x]: s

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

3       Dreamland (by:Glass Animals)
2       Melodrama (by:Lorde)
=======================================
Save this inventory to file? [y/n] y
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]: d

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

3       Dreamland (by:Glass Animals)
2       Melodrama (by:Lorde)
=======================================
Which ID would you like to delete? 2
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

3       Dreamland (by:Glass Animals)
=======================================
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]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory
 re-loaded from file.
type 'yes' to continue and reload from file. otherwise reload will be can
celedyes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

3       Dreamland (by:Glass Animals)
2       Melodrama (by:Lorde)
=======================================
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]: x

(base) grahambyron@FVFF8CE1Q6L4 Python Scripts %
```

While working with so many lines of code that was already written, it was incredibly difficult to ensure that all functions and variable names matched up with each other. Because of this, it took a long time to ensure that everything matched up (with some failures of course). After this, we can see that these did save within the correct file:

CDInventory Save



Conclusion

Assignment 06 revolved around the use of functions and classes. Further, the additional emphasis of how classes are used to help organize and parse code in a manner that is logical. This was used to build upon the list and dictionary knowledge. This was tied together in altering and moving around code to include additional functionalities as well as group together specific commands that could be requested by the user.

GitHub Link

https://github.com/gwiby123/Assignment_06

Appendix

Knowledge Test Code

```
#------------------------------------#
# Title: Assignment06_Starter.py
# Desc: Working with classes and functions.
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# GByron, 2022-Mar-09, Modified/ File and Completed Alterations
#------------------------------------#

# -- DATA -- #
strChoice = ''
lstTbl = []
dicRow = {}
strFileName = 'CDInventory.txt'
objFile = None


# -- PROCESSING -- #
class DataProcessor:
    """ Searches for ID in table, if it exists entry is deleted.
     if not user recieves 'Could not find this CD!' """

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

    """"""THis saves the data within the system"""""

    def save_data(strFileName):
        if strYesNo == 'y':
            objFile = open(strFileName, 'w')
            for row in lstTbl:
                lstValues = list(row.values())
                lstValues[0] = str(lstValues[0])
                objFile.write(','.join(lstValues) + '\n')
            objFile.close()
        else:
            input('The inventory was NOT saved to file. Press [ENTER] to return to
the menu.')

    def adding_cd():
        cd_lst= IO.add_cd()
        dicRow = {'ID': cd_lst[0], 'Title': cd_lst[1], 'Artist': cd_lst[2]}
        lstTbl.append(dicRow)

```

```
55          class FileProcessor:
56              """Processing the data to and from text file"""
57              @staticmethod
58              def read_file(file_name, table):
59                  """Function to manage data ingestion from file to a list of dictionaries
60
61                  Reads the data from file identified by file_name into a 2D table
62                  (list of dicts) table one line in the file represents one dictionary row in
63          table.
64
65                  Args:
66                      file_name (string): name of file used to read the data from
67                      table (list of dict): 2D data structure (list of dicts) that holds the data
68          during runtime
69
70                  Returns:
71                      None.
72                  """
73                  table.clear()
74                  objFile = open(file_name, 'r')
75                  for line in objFile:
76                      data = line.strip().split(',')
77                      dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
78                      table.append(dicRow)
79                  objFile.close()
80
81          # -- PRESENTATION (Input/Output) -- #
82
83          class IO:
84              """Handling Input / Output"""
85
86              @staticmethod
87              def print_menu():
88                  """Displays a menu of choices to the user
89
90                  Args:
91                      None.
92
93                  Returns:
94                      None.
95                  """
96
97                  print('Menu\n\n[l] load Inventory from file\n[a] Add CD\n[i] Display
98          Current Inventory')
99                  print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit\n')
100
101             @staticmethod
102             def menu_choice():
103                 """Gets user input for menu selection
104
105                 Args:
106                     None.
107
108                 Returns:
```

```
            choice (string): a lower case sting of the users input out of the choices
        I, a, i, d, s or x

            """
            choice = ' '
            while choice not in ['I', 'a', 'i', 'd', 's', 'x']:
                choice = input('Which operation would you like to perform? [I, a, i, d, s
        or x]: ').lower().strip()
            print()
            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('======================================')


            """Adds CD of user choice to the list, does not save the CD to memory"""

        def add_cd():
            strID = input('Enter ID: ').strip()
            strTitle = input('Enter the CD\'s title. ').strip()
            stArtist = input('Enter the Artist\'s name. ').strip()
            intID = int(strID)
            return [intID, strTitle, stArtist]

    # 2. start main loop
    while True:
        IO.print_menu()
        strChoice = IO.menu_choice()

        if strChoice == 'x':
            break
        if strChoice == 'I':
            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...')
```

```
163                 FileProcessor.read_file(strFileName, lstTbl)
164                 IO.show_inventory(lstTbl)
165             else:
166                 input('canceling... Inventory data NOT reloaded. Press [ENTER] to
167         continue to the menu.')
168                 IO.show_inventory(lstTbl)
169             continue
170         elif strChoice == 'a':
171             DataProcessor.adding_cd()
172             IO.show_inventory(lstTbl)
173             continue
174         elif strChoice == 'i':
175             IO.show_inventory(lstTbl)
176             continue
177         elif strChoice == 'd':
178             IO.show_inventory(lstTbl)
179             intIDDel = int(input('Which ID would you like to delete? ').strip())
180             DataProcessor.delete_cd_from_table(intIDDel)
                IO.show_inventory(lstTbl)
                continue
            elif strChoice == 's':
                IO.show_inventory(lstTbl)
                strYesNo = input('Save this inventory to file? [y/n] ').strip().lower()
                DataProcessor.save_data(strFileName)

                continue
            else:
                print('General Error')
```