

Nicco Fortes

12/8/22

Foundations of Programming

Module 8

Module 8

Intro

This week was an introduction into Object Oriented Programming. We learned how to create our own objects and assign attributes, properties, and methods. We learned how we can use these objects to add another dimension of functionality to our scripts.

Module 8

The beginning of this module introduced us to creating classes in python. Classes are the umbrella under which we create and define the object we are making. In these classes we are able to assign attributes, which are spaces we can store data. We can then assign getters and setters, to control how we are able to interact with the data in the attributes. Then we covered methods, which are functions we can create and are called on the instance level. Then we brought it back to the static method, which we had previously used, which are functions we create to call on the class level.

Assignment 8

I found that object oriented programming was a concept I needed to sleep on a few days to internalize. Once I had I realized this assignment was essentially no different than the previous assignments, we just needed to convert the code to handle a custom CD data type.

I experienced two main problems in this assignment. The first and primary problem was the result of a misunderstanding of naming conventions within my CD class. Initially I had both an attribute and a property getter assigned to the same name, causing a stack over error. This error was conceptually hard for me to overcome until the explanation in class. I had solved the problem by then by setting the property as private but could not understand the reasoning behind it. The second lesser problem was walking through the idea of iterating through the cd objects with a for loop to convert the table into a list of tuples.

Lastly I set the script to auto generate ID's in the system. These IDs need to be unique to each instance of CD data in our list. As the number assigned to each entry is arbitrary as long as they are unique, I decided to take that input out of the hands of the user as a simple solution to this problem.

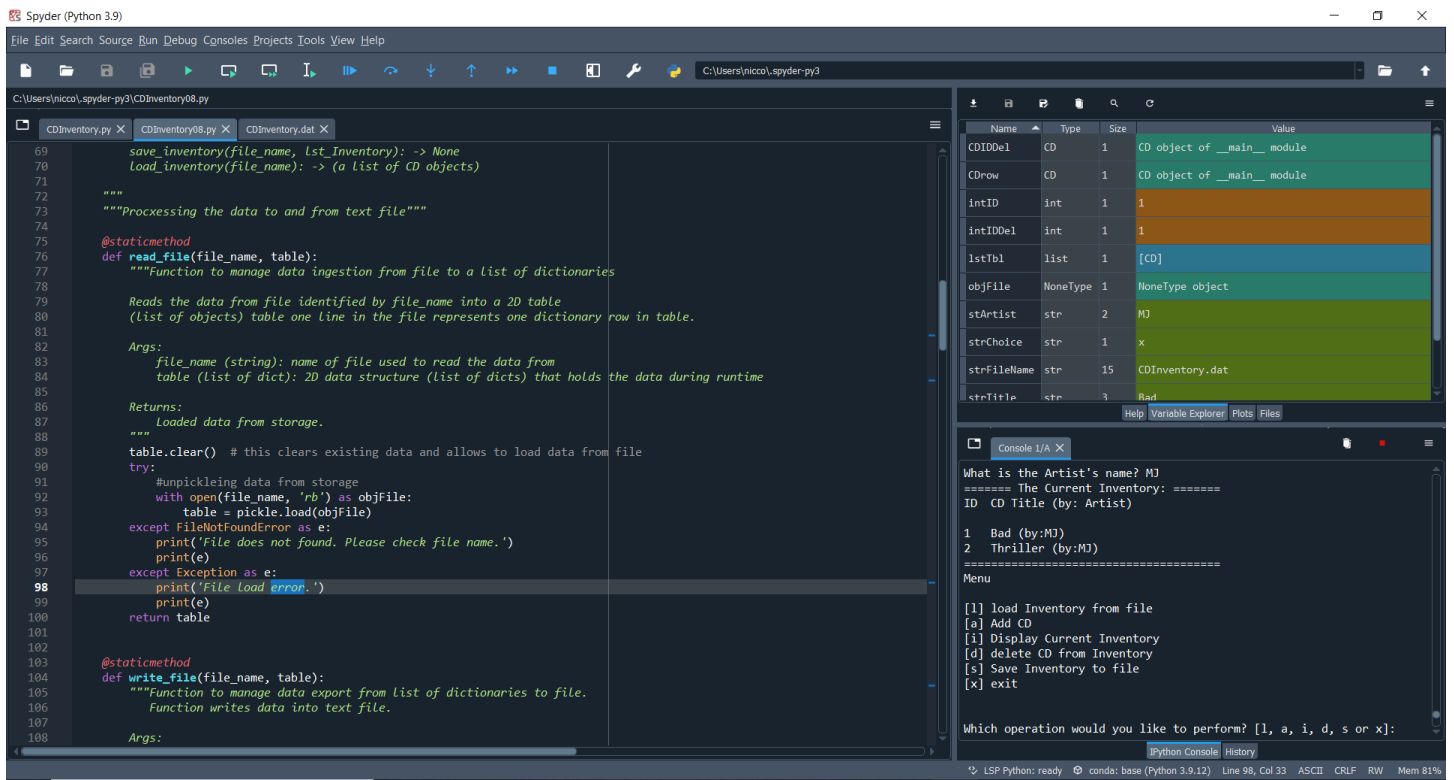


Figure 1: Script running in Spyder

```
Anaconda Prompt (anaconda3) - python CDInventory.py
[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        Bad (by:MJ)
2        Thriller (by:MJ)
=====
Which ID would you like to delete? 2
The CD was removed
===== The Current Inventory: =====
ID      CD Title (by: Artist)
1        Bad (by:MJ)
=====
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 canceledyes
reloading...
===== The Current Inventory: =====
ID      CD Title (by: Artist)
1        Bad (by:MJ)
2        Thriller (by:MJ)
=====
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]:
```

Figure 2: Script running in prompt

Summary

Module 8 was the most interesting module and the most conceptually challenging. After addressing some confusions I had around the creation of these objects and taking those learnings and applied them in the assignment I can see the vast potential this idea can have. I will say it make the structure of this class make a lot of sense as we have already been working with objects all along. It is almost as if we have just been introduced into the next dimension of programming as we are no longer constrained to the confines of preset objects in python. This feels like a big idea that we have just taken a glimpse of.

GitHub Link

Appendix/Search

Stack overflow error python

appending list python with custom class

moving syntax error python

Length of list function