Ryan Shipman

Dec 1, 2020

IT FDN 110 A

Assignment07

https://github.com/RyShip/IntroToProg-Python-Mod07.git

Pickles and Error Handling

**Introduction.**

This module was a welcome sight in a busy week. The module covered pickling/binary code and error handling. At the basic level it is pretty straightforward and easy to understand the concepts and logic behind it. In this document I will quickly go over how I implemented these concepts into some code. Essentially I repurposed the mod 6 program to ask the end user for some data and give the end user the ability to pickle or un-pickle the data. I then created a small script to demonstrate error handling. Lets jump in.

**Pickling and Unpickling**

Pickling in python is a process where a python object is converted into a byte stream. This character stream contains all the information necessary to reconstruct the object in another python script. Now, unpickling (if that's even what you call it) is the inverse of pickling, where a byte stream is converted back into an object within python.

To pickle an object, you first need to use the code 'import pickle’ this will import the pickle module which has two main methods within it. The first one is ‘dump’, which dumps or sends an object to a file object. The second is load, which loads an object from a file object. Below I will show you two functions in my code in which I used both the dump method as well as the load method. See Figures 1 and 2 for reference.

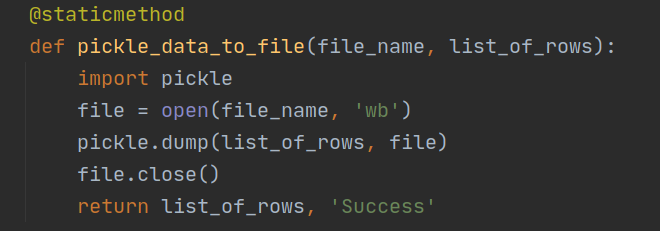


Figure 1. Dump method.

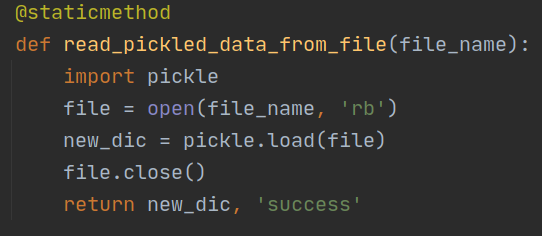


Figure 2. Load method

Now if the object in python is successfully pickled you should see the byte stream show up in the file you send it to and it should be very obscure to the user. See figure 3 for reference.



Figure 3. Pickled data.

If the data is successfully unpickled it should show up in the script. See figure 4 for reference.

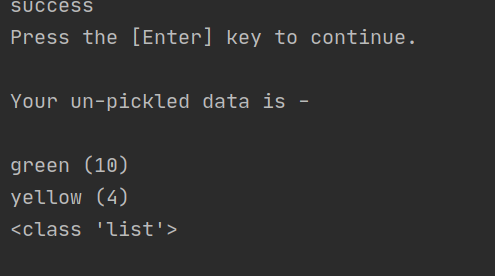


Figure 4. Unpickled data.

**Error Handling**

Next I wrote a very simple script from scratch that essentially asks the user to enter a file name for the program to open. However, since file names are case sensitive if the user forgets to capitalize the T and F in ‘TestFile.txt’ the program will run into an error. So I made a try and except block in the function that processes opening the file and ‘raised’ the exception to the main body of the script so the user would then be alerted of the error and could then enter the correct file name in and get the data within the file. To reference my code please see figure 5 and figure 6.

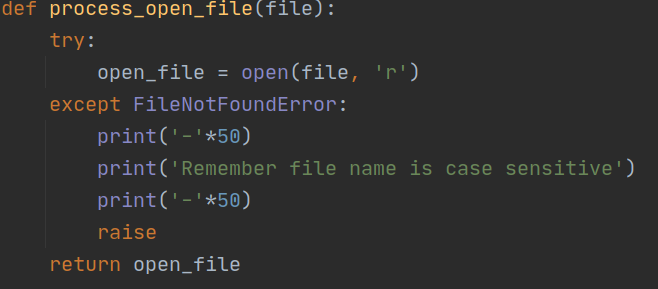


Figure 5. FileNotFoundError exception

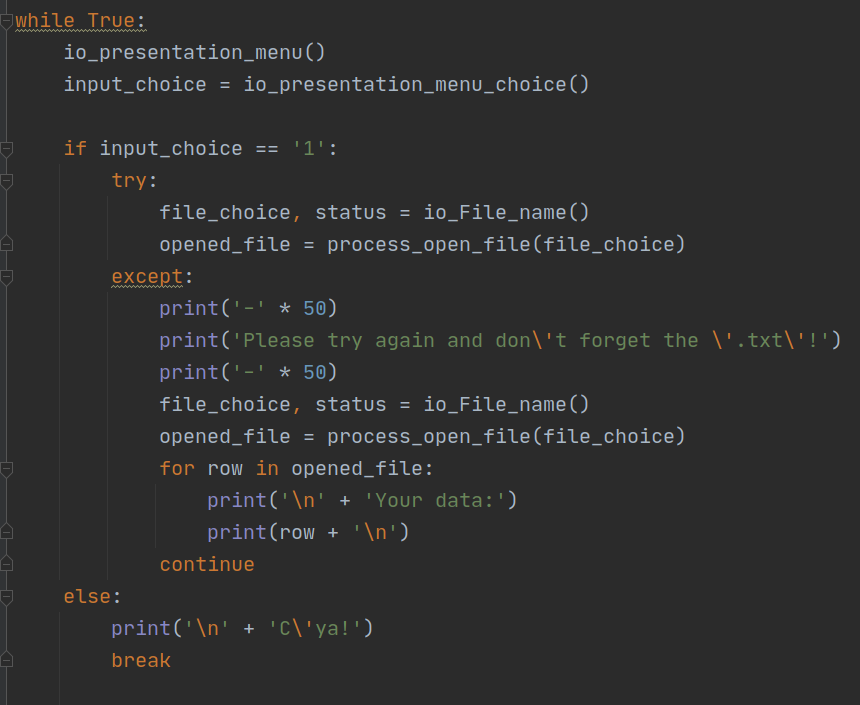


Figure 6. Try and except block in the main body of the script.

Notice that the try and except block was ‘raised’ from the function into the main body of the script which allowed me to re-run the while loop until the user eventually put in the correct file name.

**Summary**

In summary, I really enjoyed this module. It really allowed me to put everything I have learned thus far into two scripts that I essentially wrote myself or re-purposed from mod 6. I am really enjoying programming and wish I would have started sooner. Until next time.