Rebecca Ly

August 19, 2020

Foundations of Programming: Python

Assignment 07

<https://github.com/lyrebecca/IntroToProg-Python-Mod07>

**Error Handling and Pickling**

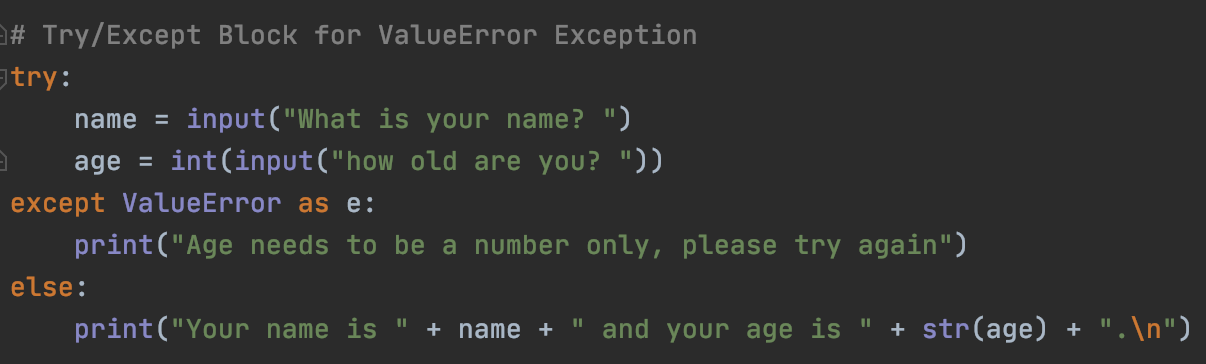
**Introduction**

This week’s assignment introduced me to two concepts within Python called error handling and pickling. Error handling allow programmers to intercept a script when it encounters an error within the code and control the message output and can allow the script to fully run without exiting out like a script would if there was no except clause to intercept. Pickling is used to convert python objects such as lists and dictionaries into byte streams for space efficiency. Pickling also allows for easier transfer of data from one system to another.

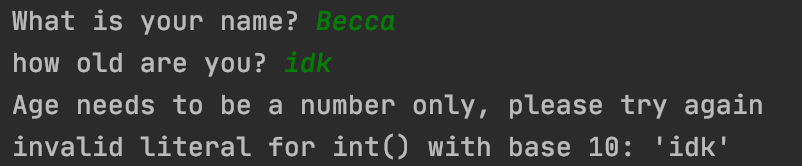
**Error Handling**

As mentioned previously, error handling allows programmers to intercept errors in order to provide their own error message or to allow the code to continue despite an error by putting the code in a try/except block. A try/except block has a segment of code in a “try” block which is the code you want to run and the “except” block will handle the error.

In Figure 1 below, in the “try” block, I’m asking for two pieces of information from the user which are the name and age. The age variable can only accept integers and will error out if anything else is entered. I used the “ValueError” exception because I know that’s the error that will pop up if the wrong value type is entered and cannot be converted. I can print my own error message plus use Python’s error message as well.



*Figure 1: Try/Except block asking for name and age from user*



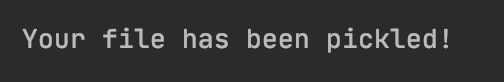
*Figure 2: Output from try/except block with an error*

**Pickling in Python**

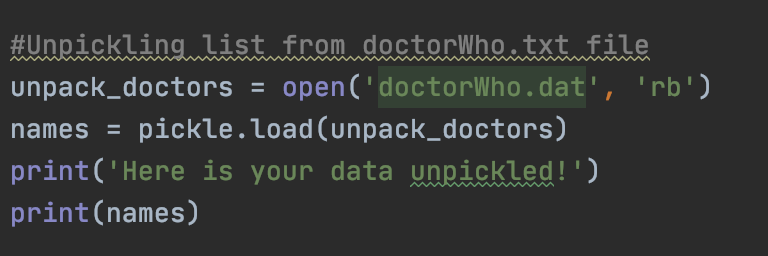
As shown in Figure 1 below, the “pickle” module is imported so we can use their methods such as “dump” and “load” which “dump” is shown below in Figure 3. In my pickling example, I have created a list of actor names from the show “Doctor Who” and writing them to a file called “doctorWho.dat”. Instead of writing to a plain text file like before, I’m writing it to a byte file which has a “.dat” extension instead of “.txt”



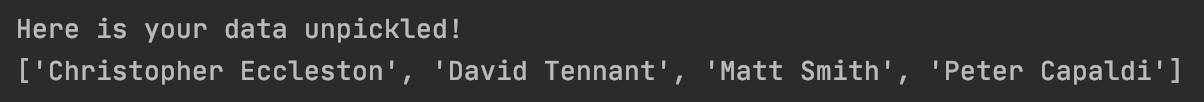
*Figure 3: Using pickling to write a list to a file*

**

*Figure 4: Print statement after pickling the list*

Now I want to read the list back from the file which I’ve done in Figure 3 below. I created a variable to open the text file and used the mode “rb” which means to read the bytes from the file and the “names” variable grabs the list using the “load” method from the file. The output is the same list that I provided in Figure 3, unpickled as shown in Figure 6. **

*Figure 5: Unpickling the list*

**

*Figure 6: Print statement calling variable to unpickle the list*

**Summary**

This week’s module gave me understanding around two topics that are very useful, specifically error handling. Error handling allows me to have more control of my code and prevent the code from exiting if it encounters an error. Pickling is a new concept that I understand most of it but am having trouble understanding when I would need to use it but I look forward to learning more about pickling in the future.