Michael Clark

February 29, 2021

IT FDN 110 B

Assignment 07

<https://github.com/MClark89/IntroToProg-Python-Mod07>

Pickling and Error Handling

Introduction

This week we delved deeper into the python language and discussed more abstract concepts than just gathering and processing data. The two concepts that were focused on were pickling and error handling. Pickling in short is a way to encode information and store it on a hard drive. While error handling is a tool used to prevent user error.

Research

### Pickling

[https://www.youtube.com/watch?v=XzkhtWYYojg&ab\_channel=RealPython](https://www.reddit.com/r/Python/comments/4t6tah/what_is_pickling/)

<https://www.datacamp.com/community/tutorials/pickle-python-tutorial#what>

The above links provided good basic knowledge on the topic with somewhat of a deeper understanding. How I normally start researching is to google whatever topic I am interested in and type in “… for dummies” at the end. This normally leads to more simplified explanations of new or complex information. The first link is to a video that goes into depth about how to create a pickling script, since I am more of a visual learner, YouTube and other video media is more helpful in teaching me how to execute programs. However, the next link from datacamp goes into a deeper level of understanding that at first gives a general idea of what pickling is and then slowly expands on the knowledge.

### Error Handling

<https://www.datacamp.com/community/tutorials/exception-handling-python>

<https://www.youtube.com/watch?v=NIWwJbo-9_8&ab_channel=CoreySchafer>

Once again datacamp.com offers a more general explanation of what error handling is and builds a foundation of knowledge that can be expanded upon. While the video offers a walk through that give a visual of how the process looks and acts in a live python program.

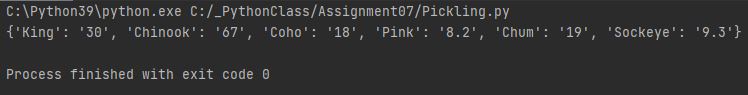
## Pickling Data

Pickling data is doing exactly that, taking data and storing it in a file that can be saved and easily accessed later. Pickling turns data that is understood by the human and converts it into bytes (Figure3.). These bytes reduce the size of the data and allow it to be stored more efficiently. An important note about pickling is that you never want to download a pickle file from an untrusted source, it can store malicious code that can damage your computer. When creating a pickle file in python you must first have a list of data points that need to be stored. In Figure 1 these are represented by dicSalmon and dicBear where each name corresponds to a species of animal and its weight. We specifically wanted to pickle dicSalmon and turn that dictionary into a byte. This was accomplished by first creating a .txt file and using the wb command which writes the dictionary in a binary format, ‘w’ = write and ‘b’ = binary. Then using pickle.dump which takes the information in the dictionary and encodes it as a binary.

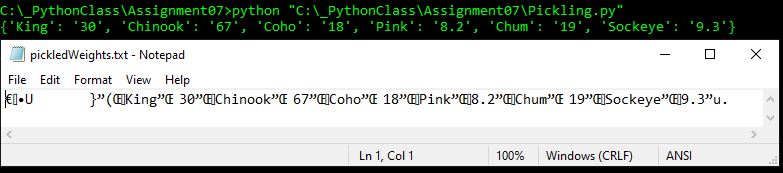


***Figure1. Pickle Script***

This step saves that data onto a .txt file in a binary form (Figure 3), which helps to efficiently save the data in a way the allows the computer to easily access it. But if you want to reupload the information, you will need to define what you want the program to do, which is to open the file and use the rb function, ‘r’ = read and ‘b’ = binary (Figure 1). Then you employ the pickle.load function which then converts the binary pickle data back into human read text as shown in Figure 1 and Figure 2.



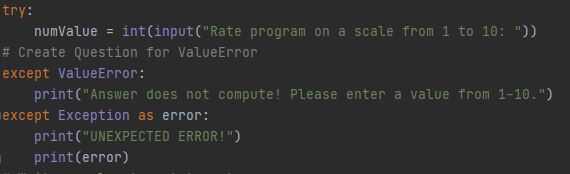
***Figure2. Pickle Check in PyCharm***



***Figure3. Pickle Check in Command Window Showing Byte Data in .txt file***

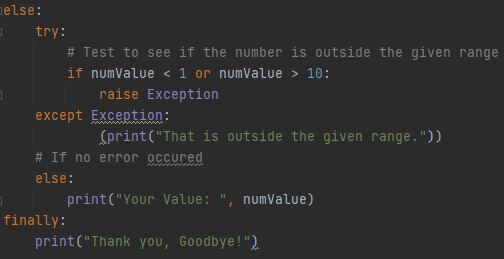
## Error Handling

When working with the general public, errors are bound to happen. Error handling is a way to help prevent such errors from ruining your script. There are a multitude of different errors that you can encounter while coding, but you can also predict these errors. In my example code we had the end user enter a numerical value, and a common error when asking for numbered data is the Value Error. The Value Error occurs whenever the end user enters an integer that is not within the range given. We first needed to identify that the value error was possible, we accomplished this by entering in the try function to utilize the input function. Then we needed to create a line of code that identified that the value error occurred and present that to the end user (Figure 4).



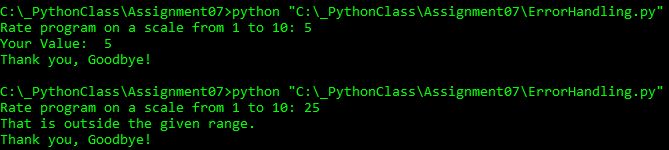
***Figure4. Creating a user-friendly Script***

After that is accomplished, the next goal is to set up the parameters that the code must follow. In Figure 5 we use an if statement to define the range of values that are considered acceptable. The else – try – raise – except – else – finally setup in Figure 5 is the common way to execute a Value Error handing sequence.



***Figure5. Value Error Handling for an integer***

In order to prove that this script worked we needed to first enter a value within the data range, which showed that the Value Error did not occur since the data was within range. However, when the number was outside of the acceptable range the error handling script kicked in and prevented the user from entering in a wrong value (Figure 6).



***Figure6. Error Handling test in command window***

Summary

As we develop a deeper understanding of python, we can now learn to prevent mistakes that used to cause errors in our script. By using error handling, we can predict where errors might occur for the end user and create code specifically to prevent this data from entering our database. In relation to data, we can use pickling to convert large datasets into compressed binary files.