Austin Biehl

August 24, 2020

Foundations of Programming: Python

Assignment 07

https://github.com/biehlag/IntroToProg-Python-Mod07

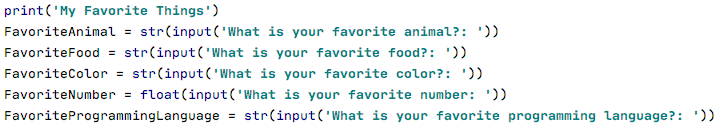
# Pickling and Error Handling in Python

## Introduction

This week our task was to write our own script to demonstrate pickling and error handling functionalities with the Python language. In this document, I will talk through the process that I went through, as well as some of the challenges that I faced. Overall, this document should help the reader understand the basics of storing and retrieving data using the pickle command.

## Writing the Pickling Script

I wanted to create a very simple script for the user so that I could focus on the new functionalities that we were learning this week. To do this, I wrote a simple script that simply asked the user to input information about their favorite things (Figure 1).



**Figure 1. The simple script asking for user input**

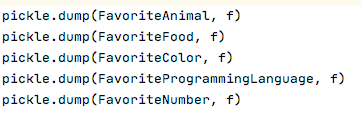
After this, I knew that I wanted to create a place to store the information that I had gathered from the user. To do this, I created a new file (Figure 2).



**Figure 2. Code to open a file to store the user’s information**

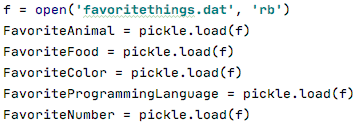
Note that the ‘wb’ specifies that the user will be writing to a binary file. This means that if that ‘favoritethings.dat’ file already existed, it would be overwritten. Since the file doesn’t exist, this command creates a new, fresh file.

After this, I added code to store the information I’d gathered from the user in the favoritethings.dat file. This is done using the pickle.dump command (Figure 3).



**Figure 3. Pickle.dump command, storing user data to the binary file**

Finally, I used the pickle.load command to get the data back out from the binary file I had created and stored the user’s data in.

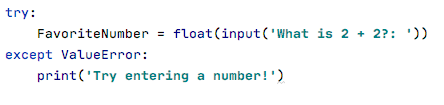


**Figure 4. Pickle.load command to retrieve data from the binary file**

Note that before using the pickle.load command, you need to tell the program to open the file and read it using the ‘rb’ command.

## Error Handling

In this program, I also created a short error handling demonstration. When there is an error in Python, sometimes the error message is confusing or unhelpful. However, by using error handling you can provide the user with a more detailed or helpful explanation of something that they weren’t doing right in the program. For example, I created an easy math problem for the user to complete. However, I used the try/except logic in Python to let the user know if they entered an answer that didn’t work (Figure 5).



**Figure 5. Error handling for a ValueError**

Note that within the try, the program specifically asks the user to input a float value. Should the user fail to do so, a custom error message ‘Try entering a number!’ will appear (Figure 6).



**Figure 6. Output of the Try/Except**

## Summary

This week’s assignment helped me to understand the basics of pickling and error handling. I can see the benefits of pickling, particular if I was working with larger, more complex sets of data. With the error handling, while I was able to display a custom error message to the user, I wanted to be able to do more with actually helping the user to resolve that error. For example, letting them try entering a new value, rather than just having the program end. I’m sure that further iterations upon this script would make that possible.