Kyle Biondich

8/20/2023

[IT FDN 110 A](https://canvas.uw.edu/courses/1655585)

Assignment 07

[**https://kbiondo.github.io/IntroToProg-Python-Mod07/**](https://kbiondo.github.io/IntroToProg-Python-Mod07/)

Pickling with Errors

# Introduction

Week 7 of the course introduced pickling and error handling in python. The following paragraphs outline the examples I found on the internet and the methods that were used to create a game save state into a state.bin file as well as how the try, exception, and finally error handling block statements work.

# Intended Outcome

The intended outcome of this week is the example of pickling game save states and error handling.

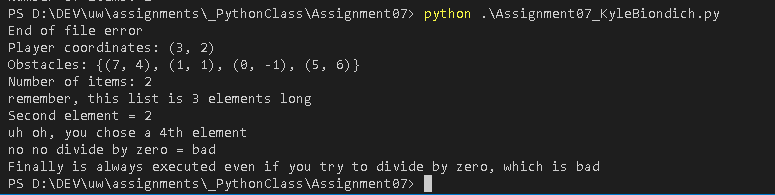


Figure 1: Intended Outcome: Game Information and Error Handling

**Learning to Pickle**

After a quick googling, I stumbled upon this website (The ultimate guide to Python pickle, n.d.). I used its example of how to pickle a game save state to a saveState.bin file, and then read that file back into the script. I chose this site because it appeared to have a short and easy example that I could use. I unfortunately ran into some issues with it and ended up having to use some of the try, except handling from the second part of the script. Technically my script runs fine, but it could use some work with the ‘EndOFFIle’ issue it seems to be creating. I’ll have to continue researching.

Initially, this example creates a class of GameItem and a second class of GameState. It gives a player, an obstacle, and items an initial state and then passes those to a state variable. From there, a saveState variable is created to hold the open function and an argument of ‘wb’ write binary is passed to it. Then pickle.dump reads the state object into a ‘saveState’ binary file.

Next, saveState is passed the open function with an argument of ‘rb’ read binary and the saveState.bin file is opened and passed into savedState. From here I get an EOFError and use the try and except handling to either open and read the file contents or to use the previous values of state.

I need to do a little more research on why the EOF error is occurring.

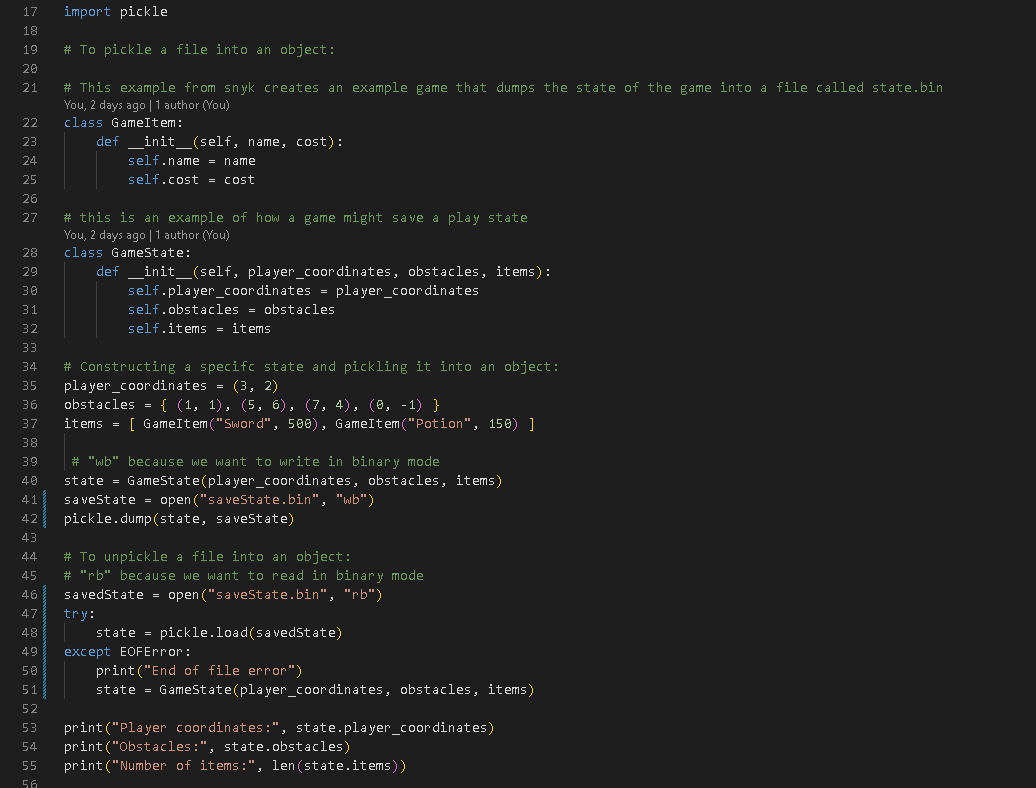


Figure 2: Learning to Pickle

## Learning to Handle Errors

I then googled a few sites about handling errors and came across this website (Python Exception Handling, n.d.). The example creates a three element list of numbers and then starts a try block and attempts to find an element at position 4. The except portion of the try block catches the error and explains that issue. Another example is the divide by zero error. This try block presents x = 10/0 and catches the error with the ‘ZeroDivisionError’ exception. This example also uses a ‘Finally’ block which executes regardless of the exceptions above.

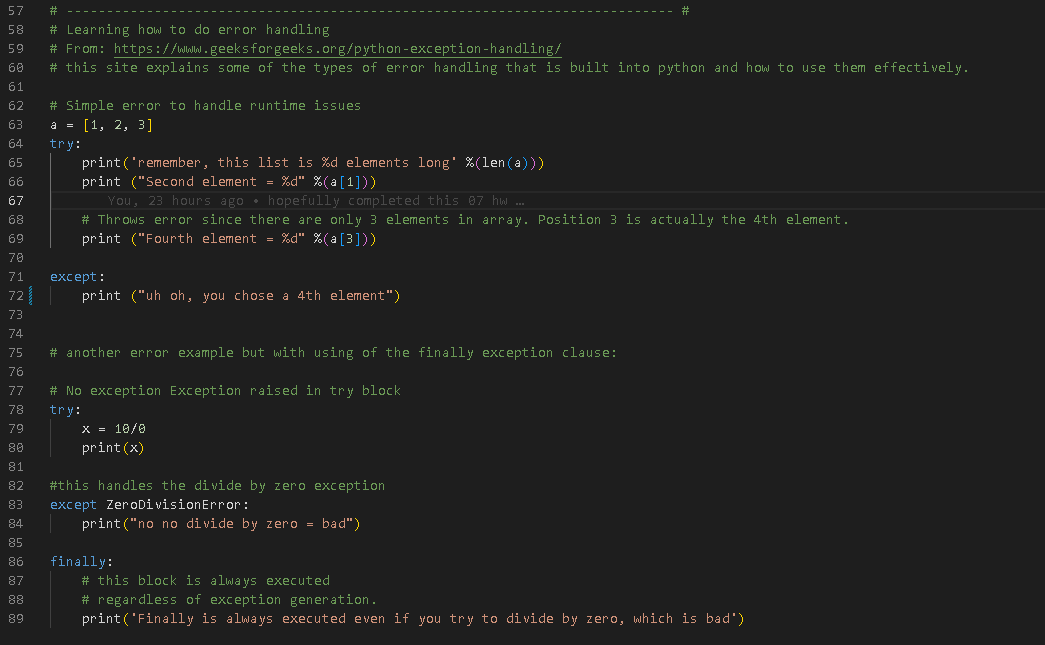


Figure 3: Handling Errors

# Observations

This week’s objective was interesting. Pickling can be a useful resource for saving obscured data which I may use at work. Although I haven’t quite figured out how to solve the End OF File error, I see this as a useful exercise. The error handling portion of the objective is also useful to understand, especially as I build applications for work. Another big understanding is the vast amount of examples around the web. I found that I could google and find examples for everything I was trying to accomplish.

# Summary

In summary, utilizing all the resources provided to the class and the online lecture, this paper outlines all the steps that were taken to create a python script that results in a successful execution of the intended outcome (Figure 1). Following the steps outlined above will allow for the audience to recreate the presented result.

# References

*Dictionaries*. (n.d.). Retrieved from AfterHours Programming: https://www.afterhoursprogramming.com/tutorial/

*Python Exception Handling*. (n.d.). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/python-exception-handling/

*The ultimate guide to Python pickle*. (n.d.). Retrieved from snyk.io: https://snyk.io/blog/guide-to-python-pickle/