Alicia Young

February 28, 2022

IT FDN 110 A

Assignment 7

<https://github.com/aliciay22/IntroToProg-Python-Mod07>

**Pickling and Error Handling Demos**

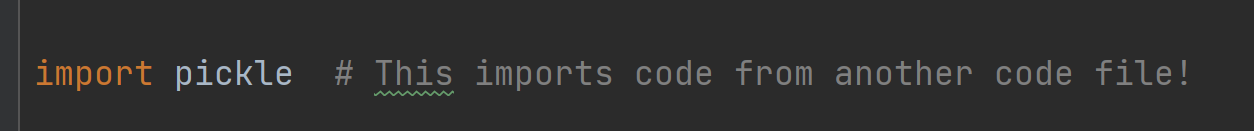
**Introduction**

The goal of this lesson is to demonstrate how pickling and error handling work in Python. Pickling refers to converting a python object to a binary file, and unpickling refers to converting the object back to its original form. Error handling refers to telling python how to handle errors, rather than having the program crash when they are encountered.

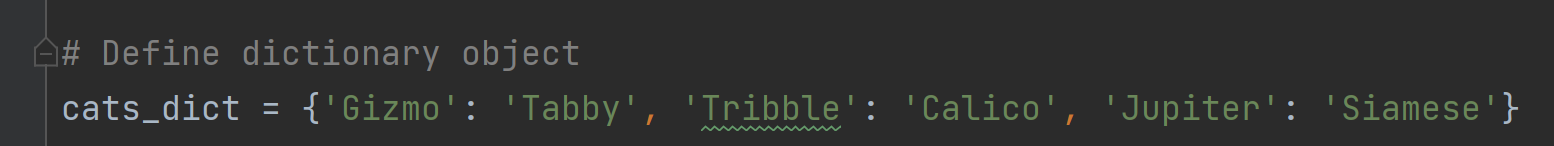
**Process**

**Pickling/Unpickling**

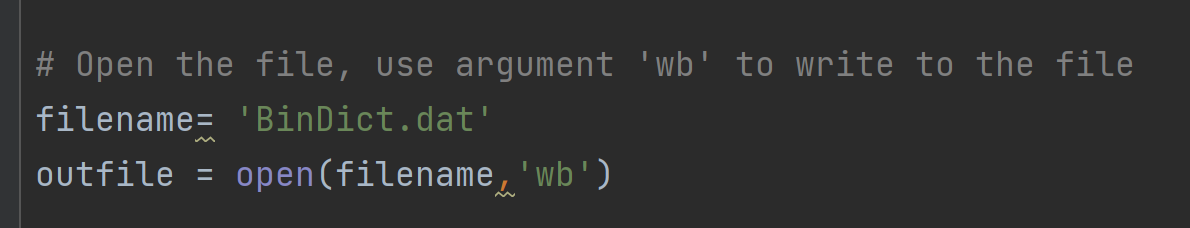
First the script demonstrates how pickling and unpicking work. To begin the pickling process, the code must import the pickle module.

****

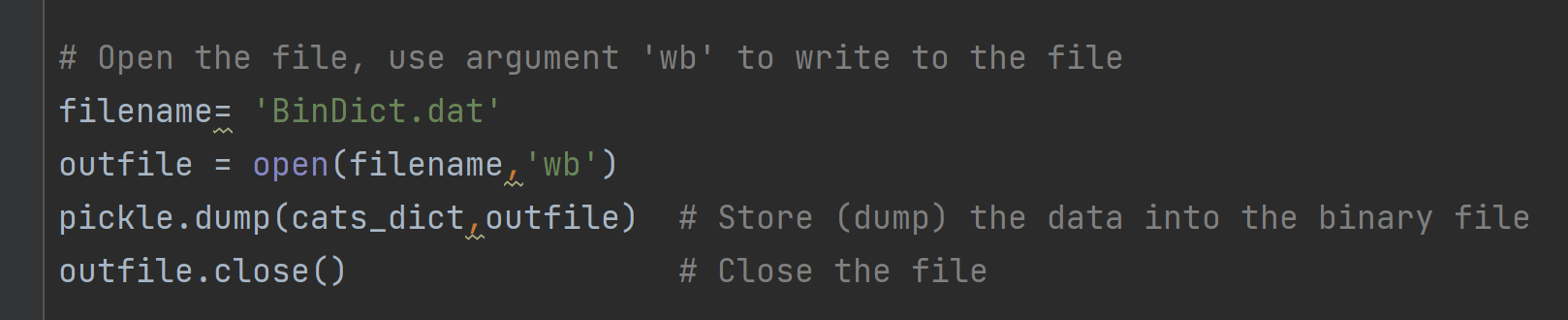
This example pickles a python dictionary object into a binary file. The dictionary object is defined.

****

Next, the file is opened, specifying the ‘wb’ option, which refers to write binary.

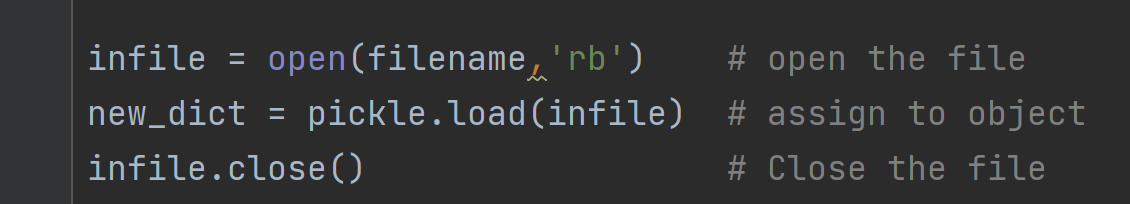


Then we use the pickle.dump() function to store (dump) the data into the file. The functions takes two argument, the object to be pickled and the file to which the object will be saved. After pickle.dump, we close the file.

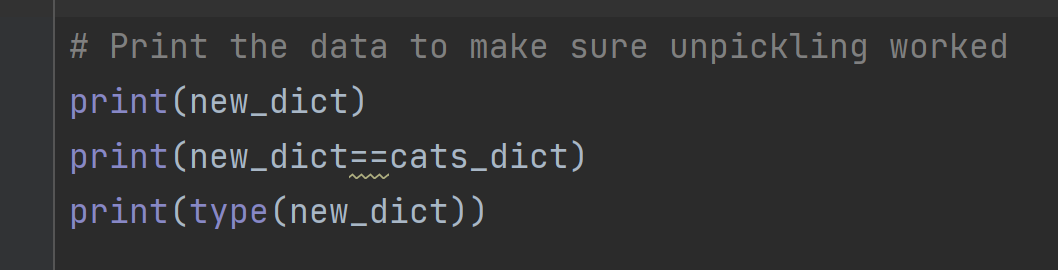


The file, BinDict.dat in this example, will be saved in the working directory.

To then unpickle a file, we open the file and then use the pickle.load() function with the ‘rb’ option (read binary) to load the data back into a python object. Then close the file.



We can print the data the new object, dictionary in this case, to make sure the results are as expected.

****

I found the following websites helpful, and the first one inspired this example.

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

<https://sites.pitt.edu/~naraehan/python3/pickling.html>

<https://towardsdatascience.com/do-not-use-python-pickle-unless-you-know-all-these-facts-d9e8695b7d43>

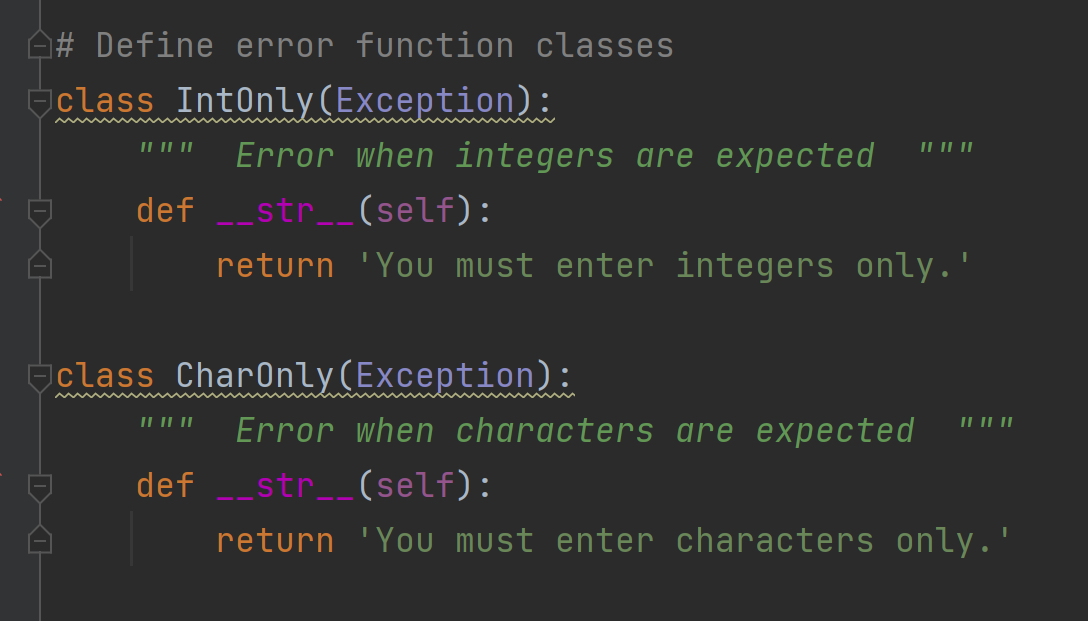
**Exceptions and Error Handling**

Error handling in Python makes the code more robust and can prevent the code from crashing by diverting to an alternate path when errors are identified or prompting the user for correct input when invalid data are supplied.

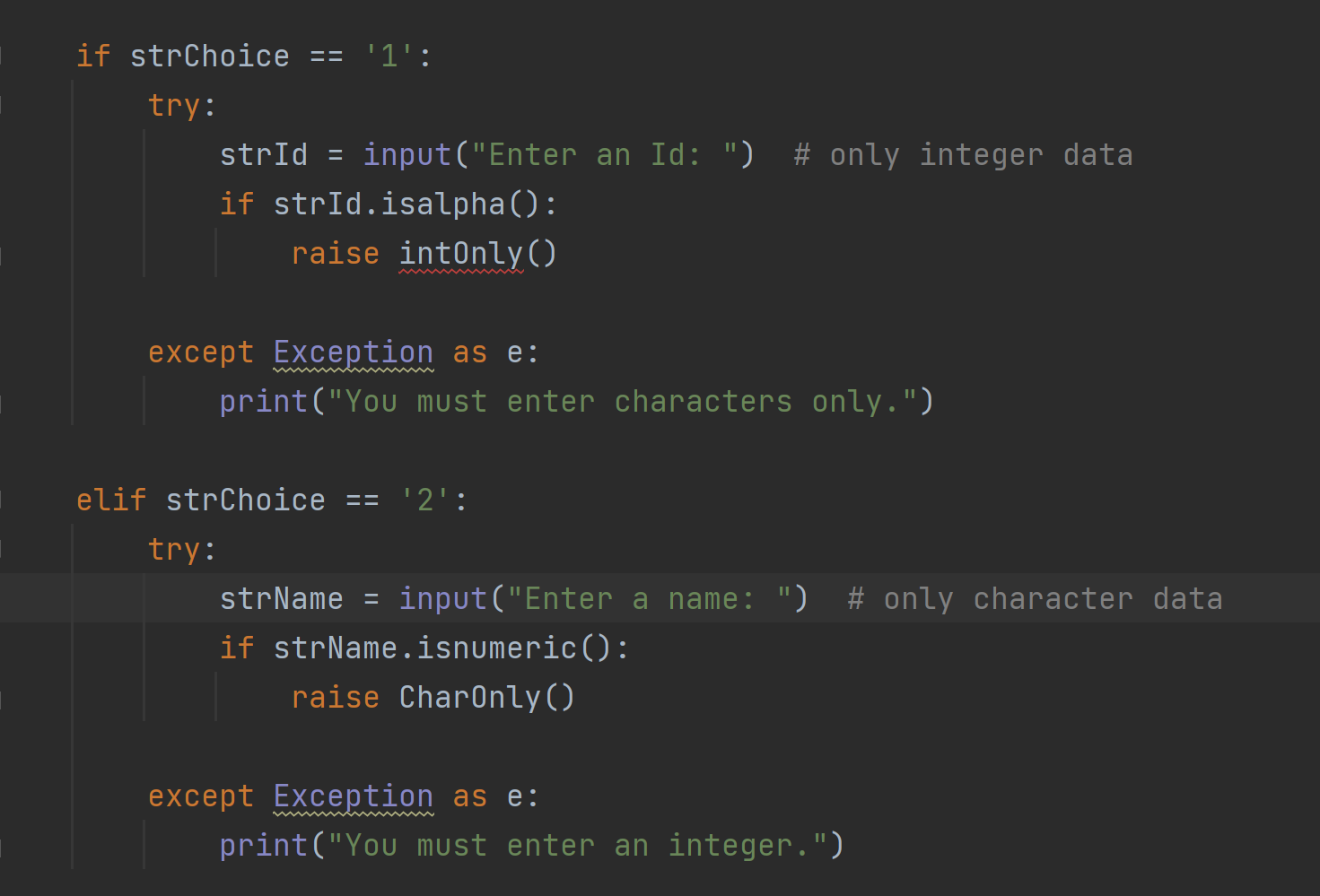
To demonstrate error handling, I created functions that checked that the user input either numeric or character data using a try/except block with a raise condition.

The code under the try block executes unless the condition under raise is met. Meeting this condition forces an exception.

In this example, I defined two error function classes, one to check for numeric data and one to check for character data.



In the main script, the user has a choice to enter either an ID (numeric data) or a name (character data). If the user enters character data when integer data is expected or numeric data when character data is expected an exception is raised and a message is printed to the user. The code continues to run then even when the user enters something unexpected.



I found the following websites helpful for error handling:

<https://www.w3schools.com/python/gloss_python_error_handling.asp>

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

<https://rollbar.com/blog/throwing-exceptions-in-python/>

**Summary**

This script provides simple demonstrations of pickling/unpickling in python and how error handling can be utilized. Pickling may be useful in help code run faster, and error handling is important for catching input and other errors.