Hyo Park

August 22, 2020

Foundations of Programming: Python

Assignment 07

<https://github.com/hyopark23/IntroToProg-Python-Mod07>

Exceptions Handling & Pickling in Python

# Introduction

For this assignment, we were asked to research on our own the two main topics of this module: exceptions handling and pickling. I will explain what the two Python features are, how they are used, and what makes them useful.

Exceptions Handling

*Exceptions* refers to unexpected error happening in Python that it cannot handle, and the program halting due to result. Suppose there is a simple program that asks the user to enter two numbers in which the first number is divided by the second number.

A screenshot of a cell phone

Description automatically generated

Figure 1: Simple program which divides first number by second number

If the user enters 0 for the second number, an exception will occur as division by 0 is mathematically impossible. As seen on the second last line of Figure 2, the program raised a *ZeroDivisonError* exception.

A screenshot of a cell phone

Description automatically generated

Figure 2: ZeroDivisionError due to division by 0.

Programs need to be written so that it prevents exceptions and accounts for different scenarios. One way to do this is through *exceptions handling*. Exceptions handlingis a feature in many modern programming languages. Python uses *try/except* statement for exceptions handling. First, *try* statement is declared after a colon, like an if statement. Under the try word, there are statements that are executed. Under the try is the *except* clause. If the code under try block raises an exception (e.g. divide by an int by a zero), it will jump to the code under exception. We can also implement an else statement that will run if no exception is raised. Figure 3 shows the revised program from Figure 1 with the try/except exception handler implemented.

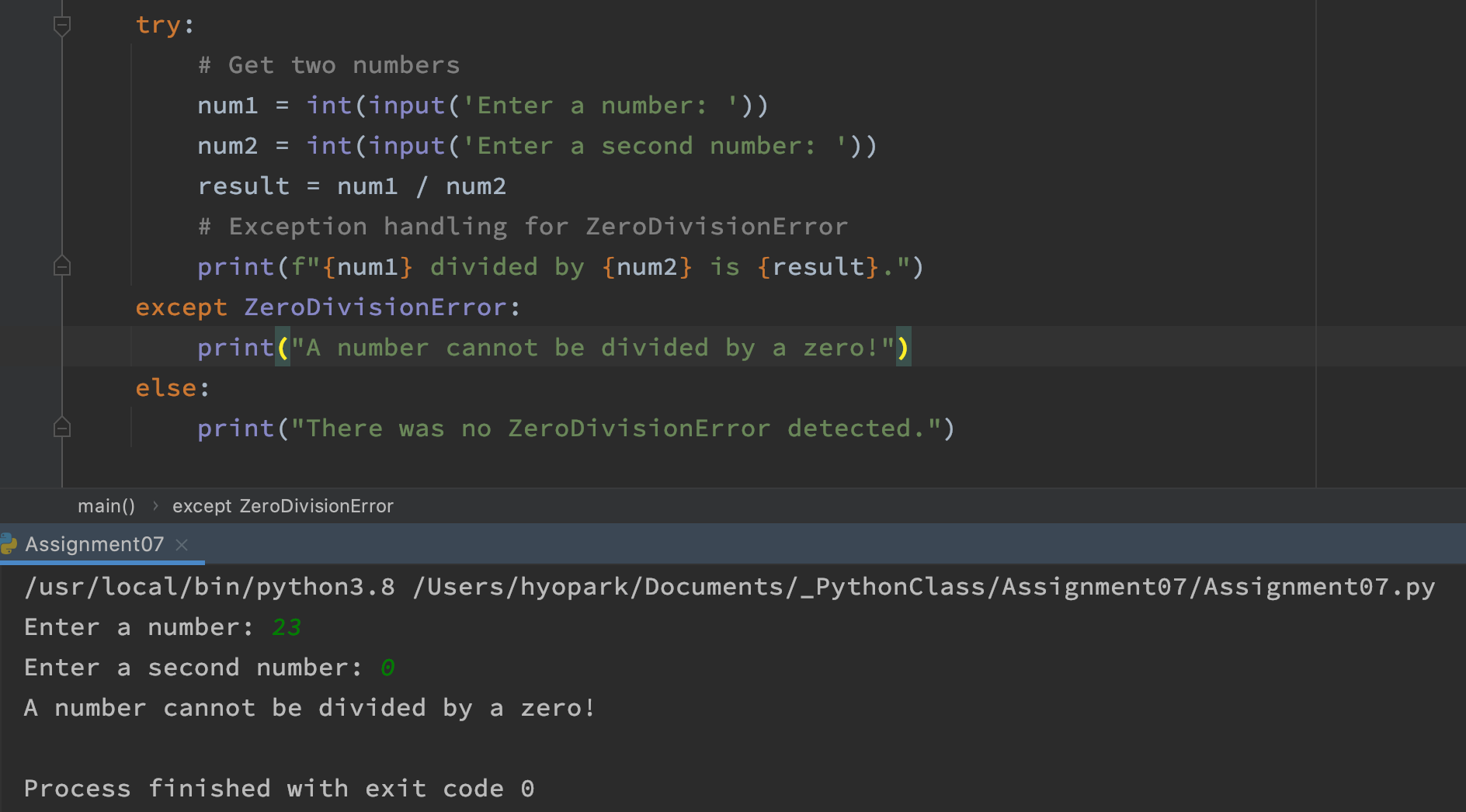


Figure 3: Handling a ZeroDivisionError exception.

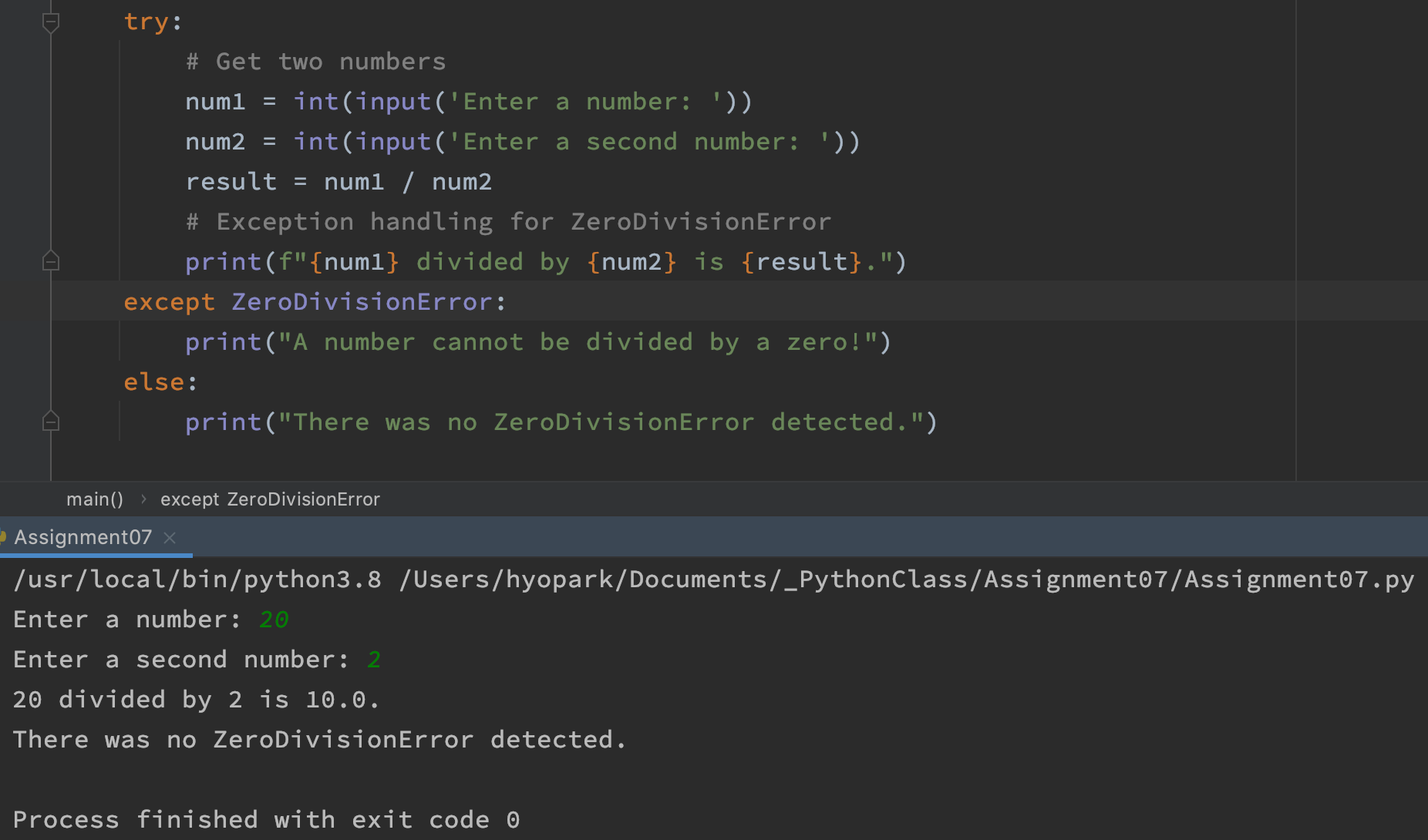


Figure 4: No exception is raised, and else statement is executed.

As you can see, the format and the idea of the try/except statement if similar to if/elif/else statement except for some minor differences. One thing to note is that there are other ways to handle exceptions. We can have multiple exceptions for specific type of exceptions. We can also have an exception without any error specified, which will execute if no other except clause is raised (catchall exception).

Pickling

We have so far learned how to read, write, and append simple string objects to a .txt file. For more complex objects such as dictionary or sets, Python has a module named *pickle*(“Introduction to Python Pickle”, [educba.com/python-pickle/](http://educba.com/python-pickle/)) (external site), which uses binary file to store data.

Pickling works similar as writing/reading data in a text file except for few variables. First, we must import the pickle module by declaring it.

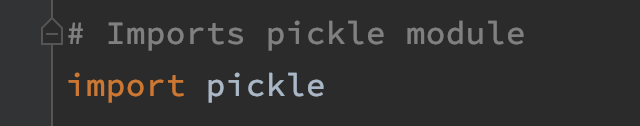


Figure 5: Importing the pickle module

In my program, I have a pre-written dictionary of my all-time favorite Seattle SuperSonics players. A .dat file named “sonics.dat” is opened for binary writing, using the “wb" method (similar to ‘w’ method for .txt files). Pickle’s dump function is then used to pickle the dictionary and write it into the .dat file. The file is then closed so it cannot be modified further. If the .dat file is not available to begin with, Python will create the file.

A screenshot of a cell phone

Description automatically generated

Figure 6: Writing a dictionary to a .dat file through pickling

Reading a binary data file is called “un-pickling”, and it works in the same fashion as writing to the file. A variable is set for the text file to be opened, and the open function is using the "rb" method. This time we use the load function of the pickle module to un-pickle the data in the file. In my program I am using a for loop to go through my list, so it prints out my data in a cleaner fashion.

A screenshot of a cell phone

Description automatically generated

Figure 7: Un-pickling a data file.

A picture containing drawing

Description automatically generated

Figure 8: Printing dictionary that was un-pickled from a data file.

# Summary

As programs get more complex, we must account for all sorts of different inputs and scenarios that the user might try with the program. Exception handling is meant just for that, as it provides us with different ways to deal with errors then simply letting a program abruptly halt. Pickling (and un-pickling) is a more efficient ways for us to manipulate writing/reading from a file as it can deal with complex objects such as dictionary or sets. It also allows transmit of data over a Transmission Control Protocol (TCP) or socket connection (Pickle in Python: Object Serialization, <https://www.datacamp.com/community/tutorials/pickle-python-tutorial>) (external site).

These new concepts will allow us to build more functional and user-friendly programs going forward.