May 20, 2021

**Tao Peng**

**Writing python scripts to demonstrate pickling and Exception handling in Assignment 07**

**Introduction**

During the past week python class, we learned the concept of pickling/unpickling and python exceptions. In this assignment (Assignment 07), I try to learn more about these concepts by searching and exploring different websites and write python scripts to demonstrate how python pickling/unpickling, and python exception handling work as part of assignment 07.

**Results**

The assignment07 is as following:

**Doing research on python pickling and exception handling on webs and write scripts to demonstrate how python pickling and exception handling works.**

After searching and exploring several websites on python pickling, I recommended this site:

<https://www.tutorialspoint.com/python-pickling>

**Definition of python pickling and unpickling:**

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can convert the byte stream (generated through pickling) back into python objects by a process called as unpickling.

**Why do we need python pickling and unpickling?**

In the real world, the use of pickling and unpickling are widespread as they allow us to easily transfer data from one server/system to another and then store it in a file or database.

**My python script to demonstrate how pickling and unpickling work:**

Please see the python script (Pickling.py) and the PDF file showing screen shots to illustrate how running of the script works.

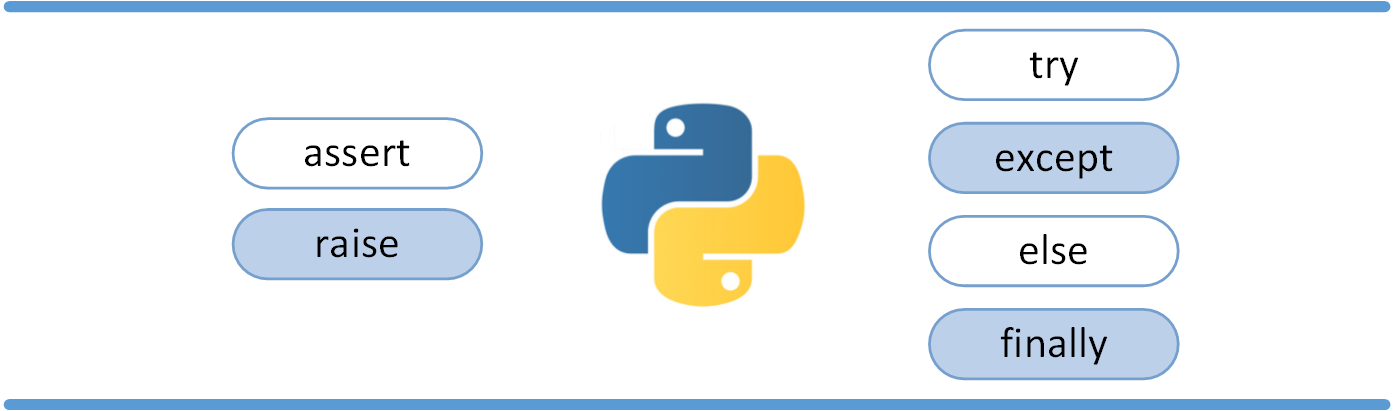
**Python exceptions**

I found this site (<https://realpython.com/python-exceptions/>) does a great job to explain python exceptions.

**Understanding the concept of python exceptions:**

A Python program terminates as soon as it encounters an error. In Python, an error can be a syntax error or an exception. Syntax errors occur when the parser detects an incorrect statement while **exception error** occurs whenever syntactically correct python code results in an error (See the PDF file to show screen captures of running python scripts to illustrate python exception handling). Python comes with various built-in exceptions as well as the possibility to create self-defined exceptions. For example, if you run the following python code: print (0/0) will result in python built-in error: ZeroDivisionError: division by zero. In **Figure 1**, I will explain the concepts of raising exceptions, making assertions, and try and except block.

**Figure 1**:



**Raising an Exception**: we can use raise to throw an exception if a condition occurs. The statement can be complemented with a custom exception. An example is shown in Assignment07\_screen\_capture.

**The AssertionError Exception**. Instead of waiting for a program to crash midway, you can start by making an assertion in Python. We assert that a certain condition is met. If this condition turns out to be true, the program can continue. If the condition turns out to be false, you can have the program throw an AssertionError exception. An example is shown in Assignment07\_screen\_capture.

**The try and except block: Handling Exceptions.** The try and except block in python is used to catch and handle exceptions. Python executes code following the try statement as a “normal” part of the program. The code that follows the except statement is the program’s response to any exceptions in the preceding try clause. An example is shown in Assignment07\_screen\_capture.

**The else and finally clause**. In python, using the else statement, you can instruct a program to execute a certain block of code only in the absence of exceptions. Imagine that you always had to implement some sort of action to clean up after executing your code. Python enables you to do so using the finally clause. An example is shown in Assignment07\_screen\_capture.

**Summary**.

Through Assignment07, I have had some unique learning experience:

1. I learned to use the rich resources in the public domains (websites as many as the stars in the sky!) to gain more knowledge on python picking and exception handling.
2. I learned to write a small script to illustrate the concepts of pickling/unpickling, and exception handling.
3. I learned to write a document to explain python pickling and exception handling.

**GitHub repository**:

The link:

<https://github.com/taopeng1100/IntroToProg-Python-Mod07>

The GiHub webpage for this repository: