Kevin Wong

November 20, 2019

Foundations of Programming: Python

University of Washington

Professor: Randal Root

Assignment 07

**Introduction**

For this assignment 7, we are to use the web to research pickling and error handling in python. And to find good examples of websites that help us understand these topics. We are then to explain the subject and why we feel these websites are good. We are then to post our assignment on GitHub, create a GitHub webpage, and then take the text, images, and links from our Word assignment document and put them into the GitHub webpage.

**Assignment**

For this assignment, I use the web to research Exception Handling and Error Handling in Python. I have organized this paper to showcase the few webpages that provide good example to learn these topics. And to explain why I chose them. I also added examples that I have learn from these websites to explain the topic.

**Research Exception Handling in Python**

I found these four webpages below on except handling exceptionally good. To me, what is a good website for teaching anything, let alone programming, is to start with the very extreme basic. A good website will not confuse the student with additional information that is unnecessary at the beginning. They will use as minimal of information as possible to get their point across. A good website will also use multiple simple examples to augment their point. A good website must also go through each step carefully because skipping a step may confuse the student. Therefore, I found these websites to be good, especially the first one on the list from w3wschools.com. What is exceptionally good about w3wschool.com is that they also format their webpages so that they are very easy to read, with lots of white spaces and large fonts and good use of background color to separate the different examples:

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

<https://www.pythonforbeginners.com/error-handling/exception-handling-in-python>

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

**Exception Handling Example**

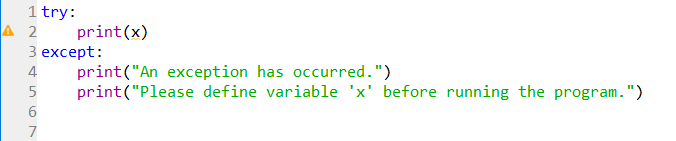
According to the website pythonforbeginners.com in the 2nd link above, if describes an exception, “as an error that happens during execution of a program. When that error occurs, Python generate an exception that can be handled, which avoids your program to crash.” The website goes to say that, “exceptions are convenient in many ways for handling errors and special conditions in a program. When you think that you have a code which can produce an error then you can use exception handling.”

Let’s look at an extremely simple example the benefit of structured error handling. Figure 1 is code that will result in an error when ran. This error is shown in Figure 2.

**Figure 2. Screenshot of result from running the code**

**Figure 1. Screenshot of a simple code to demonstrate an error.**



The error shown in Figure 2 is useful to the software developer but not too useful to the user of the program. If the error is expected, the developer can add some code to handle the error in a much more elegant way. Here, in Figure 3, is the modification to the code to solve this problem.

**Figure 3. Screenshot of modified code to handle errors.**

The resulting output from running the program, as shown in Figure 4, is much more palatable to the user. The output can be written in plain simple English as oppose to the gibberish that the user sees from the earlier output in Figure 2.



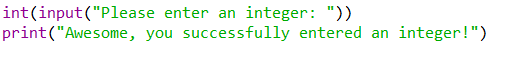
**Figure 4. Screenshot of resulting output from the modified code to handle errors.**

**Built-in Exceptions**

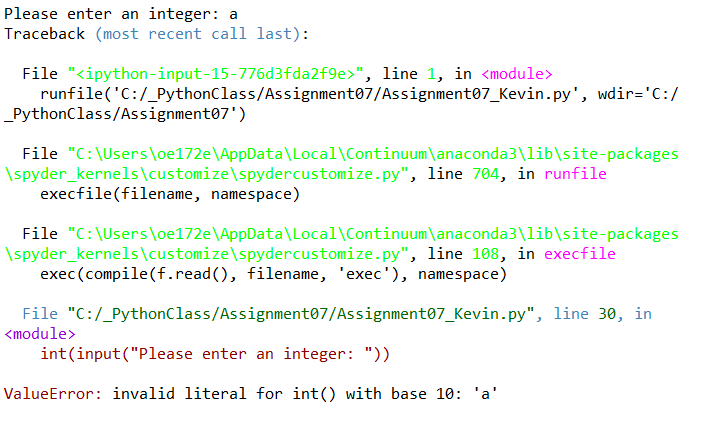
Python provides some basic exception classes which are already defined and can be used in generic cases. A list of Built-in exceptions is shown Figure 5 from the datacamp.com website.

**Figure 5. Screenshot of a list of Built-in exceptions from https://www.datacamp.com/community/tutorials/exception-handling-python**

**Exception Handling Example: ValueError**

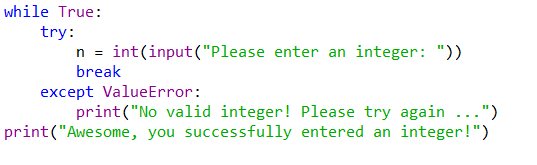
Now look at the code as shown in Figure 6. It’s asking the user to enter an integer. When the user does that, the program should display a sentence that the user has successful entered an integer.

**Figure 6. Screenshot of code that ask user to input an integer.**

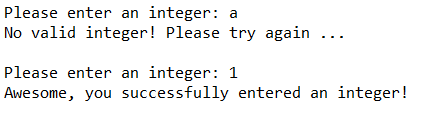
However, as you shall see in Figure 7, if the user does not enter an integer, but instead enter a letter “a”, a long and confusing error message appears.

**Figure 7. Screenshot of the error message output when user enters letter “a”.**

In order to handle this error more appropriately, we can use one of the python’s Built-in exception called “ValueError”. Look at Figure 8 for the modified code.

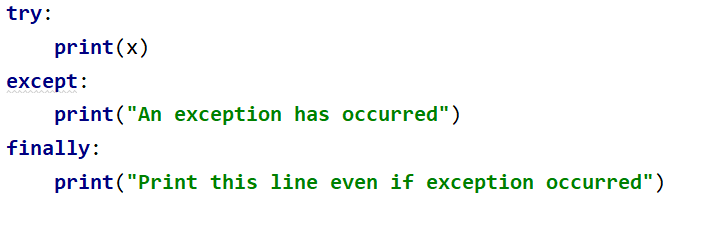


**Figure 8. Screenshot of the modified code from Figure 6 to handle the exception error.**

Now, when the user runs the program and enters the letter “a”, it will display, “No valid integer!” and to, “Please try again…” See Figure 9. The program will continuously display the same error message until the user enters an integer, then it will display, “Awesome, you successfully entered an integer!” This way of handling the error is much more user friendly.

**Figure 9. Screenshot of the output from running the code in Figure 8.**

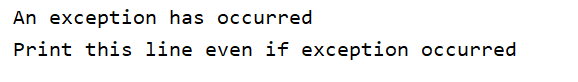
**Exception Handling Example: finally**

Now, we look at the **finally** block. Once specified, it will execute whether the **try** block raises and error or not. Look at the code example in Figure 10.

**Figure 10. Screenshot of the code using the finally block with try.**

You can see that with the **finally** block, as shown in the output in Figure 11, it will execute the code under it even if the error occurs.

**Figure 11. Screenshot of the output using the finally block with try.**



**Research Pickling in Python**

As for websites research on pickling in Python, I found these websites below to be good. The reasons why I chose them are the same reasons as mentioned above for exception handling.

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

<https://www.pythoncentral.io/how-to-pickle-unpickle-tutorial/>

<https://www.geeksforgeeks.org/understanding-python-pickling-example/>

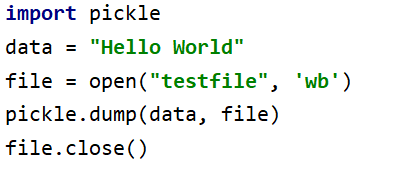
As described in the website [pythoncentral.io](https://www.pythoncentral.io/) above, the term, “pickle means storing something in a saline solution. Only here, instead of vegetables its objects.” As the webpage continues, “pickling converts any kind of complex data to 0s and 1s (byte streams). The resulting byte stream can also be converted back into Python objects by a process known as Unpickling.” In a nutshell, pickling is saving data in a .DAT file.

## **Advantages and disadvantages of Pickling**

According to the datacamp.com website above, pickling is useful for, “applications where you need some degree of persistency in your data.” The pythoncentral.io website above notes that pickling helps in, “saving complicated data.” And that it is, “quite easy to use, [as it] doesn’t require several lines of code and hence not bulky.” Additionally, the saved data is, “not so readable hence provides some data security.” However, datacamp.com warns that we should try not to, “unpickle data from an untrusted source as malicious code inside the file might be executed upon unpickling.”

**Pickling Example**

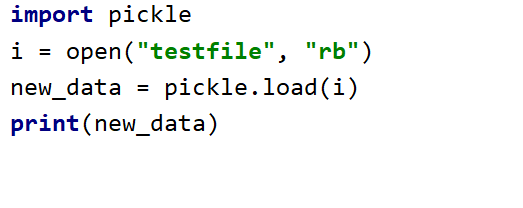
See Figure 12 for a very simple example of how pickling in done in python. We have to import the pickle function and then use the code pickle.dump() to save the data.



**Figure 12. Screenshot of a simple**

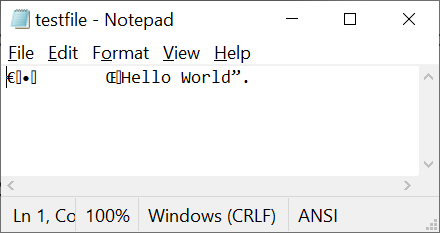
**code as an example of using pickling.**

The data is saves as a binary file. Opening the file shows a bunch of gibberish information as shown in Figure 13.



**Figure 14. Screenshot of a code file to unpickle.**

In order to open the save data or unpickle it, we use the following code as shown in Figure 14.



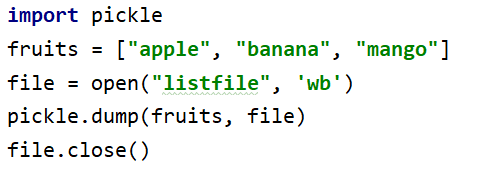
**Figure 13. Screenshot of a binary file created from pickling.**

The output from running the code above in Figure 14 is shown in Figure 15.

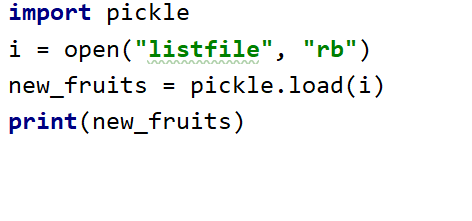


**Figure 15. Screenshot of a code file to unpickle.**

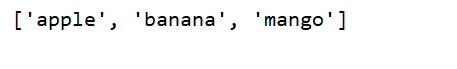
**Pickling Example: List**

As another example, we are going to use the same pickling code to save a list instead. This code is shown in Figure 16.

**Figure 16. Screenshot of a code file to pickle a list.**

In order to unpickle the list, we use the code shown in Figure 17.

**Figure 17. Screenshot of a code file to unpickle the list.**

The resulting output is shown in Figure 18.

**Figure 18. Screenshot of the output from unpickling the list.**

**Post Files to GitHub**

In this module, we need to postour Word document and Python file on a public GitHub repositoryso that others may review it.

My GitHub repository:

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

**Add a GitHub Webpage**

We will also need to add a GitHub webpage to your repository. In this assignment, we must take the text, images, and links from your Word document and put them into the GitHub webpage.

My GitHub Webpage:

<https://mynameiswong.github.io/IntroToProg-Python-Mod07/>

**Summary**

I felt that in assignment 7, half my time was spent learning about two new items in python: error handling and pickling, and the other half was learning how to transfer my Word document and on to a GitHub webpage, and to make it look presentable. Pickling is interesting as it allow the user to save more types of data than what you can just into a text file. And structured error handling is just a fancier way to present an error message, one that a layman can understand rather than a software developer.

**Readings**

1. **Chapter 7 ofPython Programming for the Absolute Beginner(Third Edition) by Michael Dawson.**

## **Module 7 PDF - \_Mod7PythonProgrammingNotes.pdf**

# List of Built-in Exceptions

<https://docs.python.org/3/library/exceptions.html#bltin-exceptions>

# GitHub Basic writing and formatting syntax

<https://help.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax>

1. **Mastering Markdown**

<https://guides.github.com/features/mastering-markdown>

1. **Markdown by John Gruber**

<https://daringfireball.net/projects/markdown/>

**Videos**

# 1. PythonMod7Project by Randal Root

# <https://youtu.be/4IkIdXJBC6o>