Nick Soldano

November 27, 2021

Foundations of Programming: Python

Assignment 07

GitHub link: https://github.com/sourcecode1386/IntroToProg-Python-Mod07

Exception Handling & Pickling

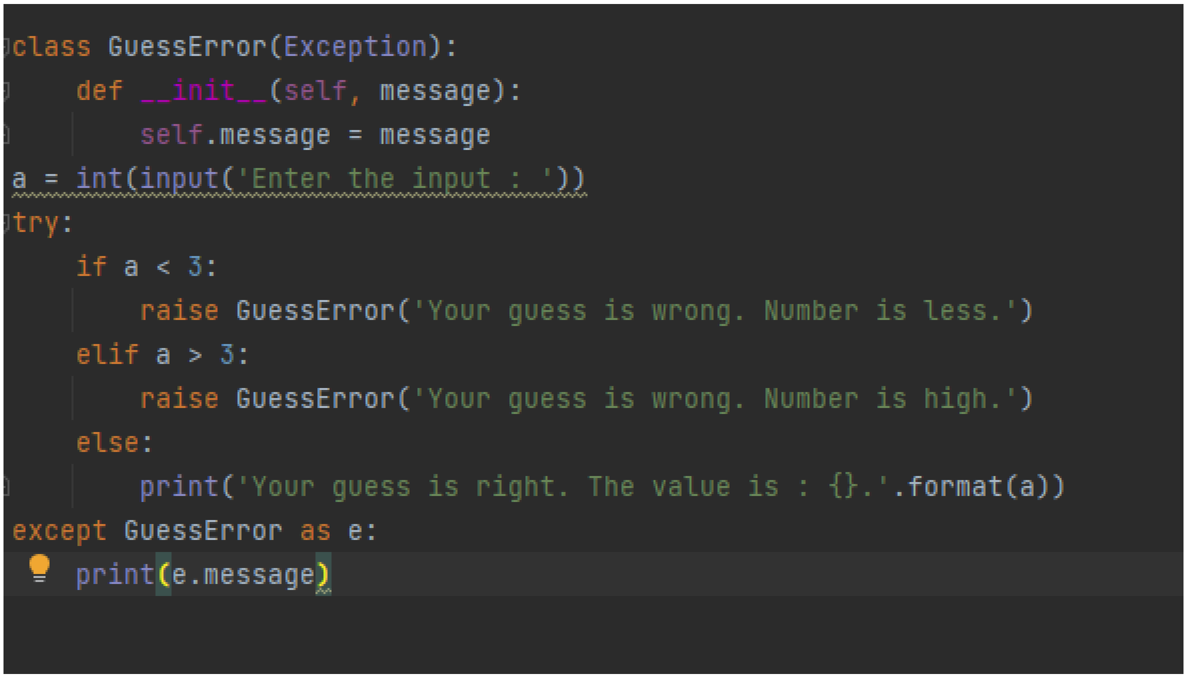
Introduction

This week's module focuses on understanding exception handling and pickling in Python. The website Medium.com has a plethora of resources related to programming in Python. Two useful articles I found on exception handling in Python are https://pub.towardsai.net/python-examples-to-make-algorithm-more-robust-with-exception-handling-6bff7a127786 and https://medium.com/technofunnel/exception-handling-in-python-fc71497e0d18. The first article has a useful code example of exception handling that I decided to use for my script assignment. As for Pickling the following article makes use of some good examples, https://python.plainenglish.io/pickling-python-objects-for-future-use-95a2ce45f443.

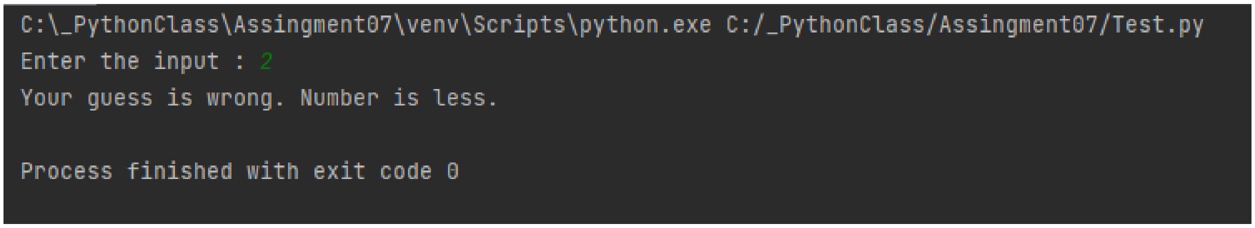
Exception Handling

Programmers require exception handling for three main reasons.

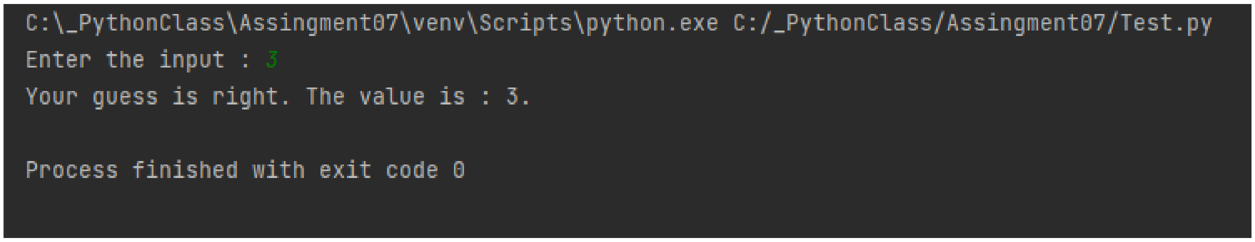
1. Error Handling: In the case of errors during runtime, exception handling can handle the scenario of Failures and avoid termination of the program.
2. Code Separation: Error Handling can help us segregate the code that is required for error handling from the main logic. The error related code can be placed inside the except block which is segregating it from the regular code that contains the application logic.
3. Grouping Error Type and Error Differentiation: It can help to segregate different kinds of errors that are encountered during the execution. It can have multiple except blocks, each handling a specific kind of error.



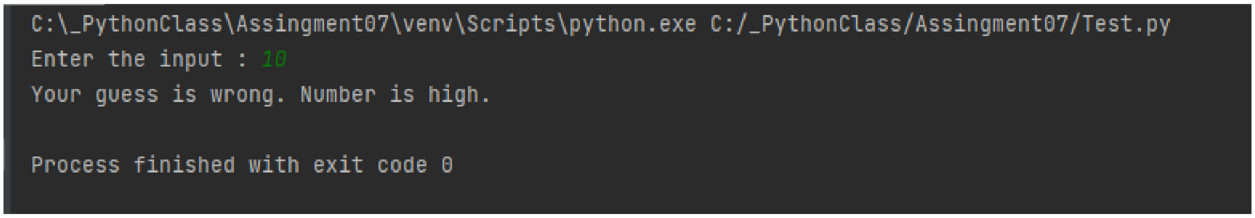
```
class GuessError(Exception):
    def __init__(self, message):
        self.message = message
a = int(input('Enter the input : '))
try:
    if a < 3:
        raise GuessError('Your guess is wrong. Number is less.')
    elif a > 3:
        raise GuessError('Your guess is wrong. Number is high.')
    else:
        print('Your guess is right. The value is : {}.'.format(a))
except GuessError as e:
    print(e.message)
```



```
C:\_PythonClass\Assingment07\venv\Scripts\python.exe C:/_PythonClass/Assingment07/Test.py
Enter the input : 2
Your guess is wrong. Number is less.

Process finished with exit code 0
```



```
C:\_PythonClass\Assingment07\venv\Scripts\python.exe C:/_PythonClass/Assingment07/Test.py
Enter the input : 3
Your guess is right. The value is : 3.

Process finished with exit code 0
```



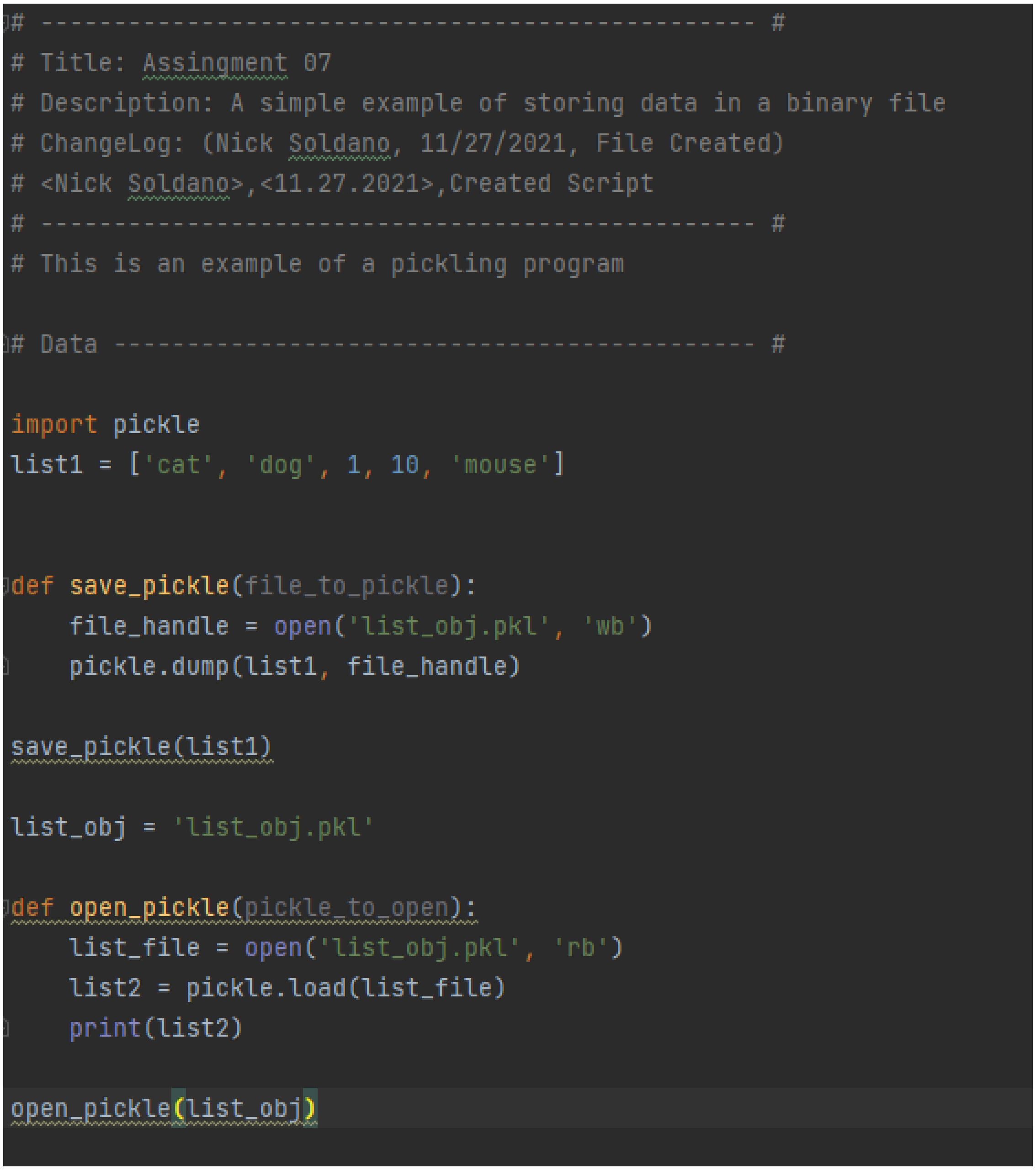
```
C:\_PythonClass\Assingment07\venv\Scripts\python.exe C:/_PythonClass/Assingment07/Test.py
Enter the input : 10
Your guess is wrong. Number is high.

Process finished with exit code 0
```

Pickling

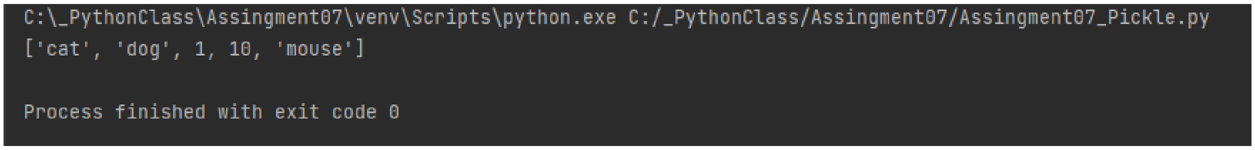
Pickling lets users save a python object in a hard disc space so that it can be used to in another program. An example of this is in Machine Learning, a predictive model can be trained on data and then saved as a pickle file. A user can then load the pickle file later and make predictions on another data set. Pickling is done for three reasons:

1. Storing data for later use

2. User's object needs to be transferred to another user /across a network
3. User needs to store some object into a database to be retrieved by some other system



```
# ------------------------------------------------ #
# Title: Assingment 07
# Description: A simple example of storing data in a binary file
# ChangeLog: (Nick Soldano, 11/27/2021, File Created)
# <Nick Soldano>,<11.27.2021>,Created Script
# ------------------------------------------------ #
# This is an example of a pickling program

# Data -------------------------------------------- #

import pickle
list1 = ['cat', 'dog', 1, 10, 'mouse']


def save_pickle(file_to_pickle):
    file_handle = open('list_obj.pkl', 'wb')
    pickle.dump(list1, file_handle)

save_pickle(list1)

list_obj = 'list_obj.pkl'

def open_pickle(pickle_to_open):
    list_file = open('list_obj.pkl', 'rb')
    list2 = pickle.load(list_file)
    print(list2)

open_pickle(list_obj)
```



```
C:\_PythonClass\Assingment07\venv\Scripts\python.exe C:/_PythonClass/Assingment07/Assingment07_Pickle.py
['cat', 'dog', 1, 10, 'mouse']

Process finished with exit code 0
```

Summary

Utilizing classes and functions for program execution in Python builds off the knowledge from previous weeks. Once the classes and functions are created calling them seems more intuitive then executing programs with just lists and dictionaries.