Peter McGiffin

Assignment 07

Intro to Python Programming

May 31, 2022

Introduction

The objective for assignment 07 is to research the topics of 'Pickling' and 'Exception Handling' and to demonstrate these methods in Python. Pickling is a library module in Python that allows python objects to be serialized and unserialized. Pickling allows someone to store a python object into bitstream in a way that it can be recalled back into an object at a later time. This could be used when working with a large data set that requires the programmer to read in. Pickling allows for far more efficient programming. At a fundamental level, pickling allows more efficient processing power by reducing the need for scripts to read large data sets each time they are worked with.

Exception Handling is a method in Python for planning for errors. Using Exception handling will catch errors and allow for code to be entered to handle the error in a way desired by the programmer. For example- exception handling could be used to avoid a 'file not found error' from breaking the code.

Pickling

Pickling allows any class to be serialized and deserialized. This can be incredibly useful when dealing with a large amount of data or any data that requires a large amount of computer processing power to calculate or compute. For example- a financial API that broadcasts share price and investment data could be downloaded and pickled to save time when loading this data at a future point.

Using Pickle

To use pickle in Python, you must first import the pickle module- this is done simply, by the code *import pickle.* There are four basic pickle methods that can be used- these are shown in figure one.

```
16  #Pickle methods
17  # pickle.dump() data type -- > serialized
18  # pickle.dumps() (from bit string)
19  # pickle.load() serialized -- > data type
20  # pickle.loads() (from bit string)
```

Figure One. Pickle methods in Python.

Pickle dumping is accessed by *pickle.dump()*. This serializes the data. Loading the data can be achieved with *pickle.load().* Dumping will write the object to a file and loading will load the object from the file.

Any class of data can be pickled. Figure two (below) shows a dictionary data type pickled and retrieved.

```
# example_dict = {1: "example1", 2: "example2", 3: "example3"} #Creating an example dictionary
#
# pickle_out = open("dict.pickle","wb") # creating pickle method
# pickle.dump(example_dict, pickle_out) #dumping data in pickle and parsing through dictionary
# pickle_out.close()
#
# pickle_in = open('dict.pickle', 'rb') # pulling data through pickle method.
# example_dict = pickle.load(pickle_in) # loading data from pickle
#
# print(example_dict) # printing data
# print(example_dict[2]) # printing data index 2 to demonstrate data has not changed.
```

Figure Two. A dictionary that has been pickled and loaded from the pickle.

Exception handling

Exception handling in Python code is important to write professional, standard code in Python. Anytime an error is possible in your Python code, an exception handling script should be used to build more robust and professional code. It's important to first understand the different types of errors that could be encountered in Python code. Script error, type, file not found or Syntax errors are some of the common types of errors. A full list of exceptions can be found at: https://docs.python.org/3/tutorial/errors.html

The standard way of writing Exception clauses in Python is *try, except, else* then *finally.* This can be shown in Figure Three (below).

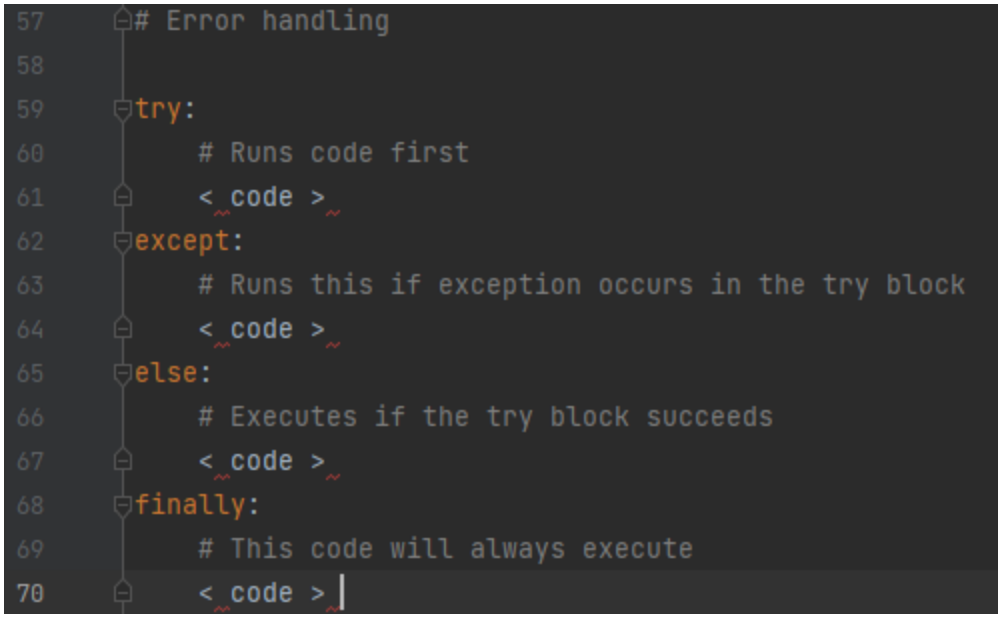

Figure three. Exception clauses in Python

Conclusion

Pickling and Exception handling are both valuable tools to write professional and efficient code. Pickling is useful when dealing with large data types that require a lot of processing power and exception handling is important when writing complex code, to ensure the code functions appropriately and to customize any error outcomes.