Steven Huang

May 24, 2020

IT FDN 100 A Sp 20: Foundations Of Programming: Python

Assignment06

<https://github.com/shpy086/IntroToProg-Python-Mod06>

Assignment Recap

Introduction

In this assignment, we are asked to create a script that demonstrates how pickling and structured error handling are implemented in Python. In the first part of the script, I will explain how pickling can be used to store data on a .dat file in binary format and then loaded from a .dat file and converted into its original readable format. In the second part of the script, I will explain how structured error handling can be used to trap errors in the program and relay them back to the user.

Pickling

Pickling is a technique that allows us to preserve data in binary format as opposed to storing it as “plain text”. [Python.org](https://docs.python.org/3/library/pickle.html) defines pickling as “implementing binary protocols for serializing and de-serializing a Python object structure.” The main benefits of pickling are the ability to store more complex data (relative to text files) with a single line of code and reducing its storage size. Figure 1 below shows the code in the data and processing layers of the script:

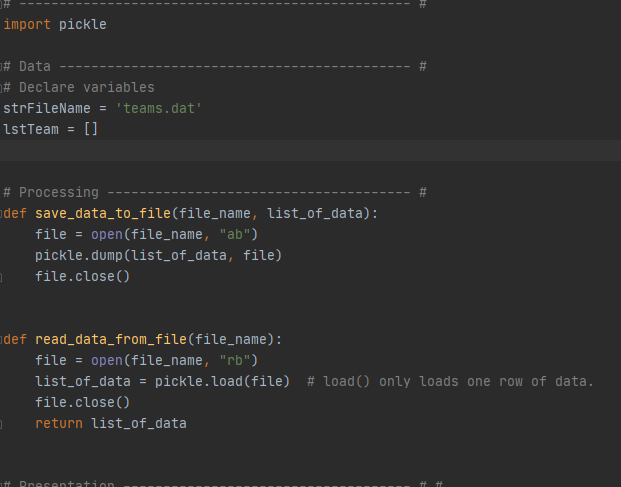


Figure 1

In the data layer, I declare the two variables that we will be using: the first is the .dat file and the second is the list object that will be stored in the file. In the processing layer of the code, we have created two functions that that will be used to manipulate the user’s data. The first function will write the data to a .dat file in binary code. The second function will read binary code from the .dat file and convert it back to readable format.

In the third and final layer of the script, the presentation layer, the user is prompted to enter data which establishes two more variables: “sport” and “team”. Once the user inputs both pieces of data, the program calls the “save data to file” function store the data as a list object in the .dat file. Then, the program calls the “read data from file” function to read data from the .dat file and print its result. Figure 2 below shows the presentation code:

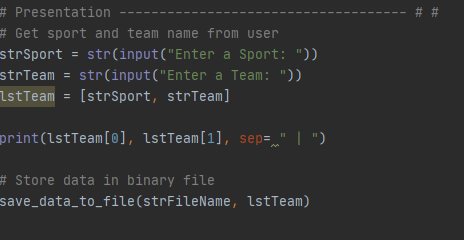


Figure 2

When running the script in PyCharm, it executes the script successfully. In this case, when the user is prompted to enter a sport, they input “football” and when prompted to enter a team, they input “Giants”. When the two are combined, they form a list object. The sport is the category of the list object and the team is one of the teams within that category. Figure 3 below shows the script running in PyCharm:

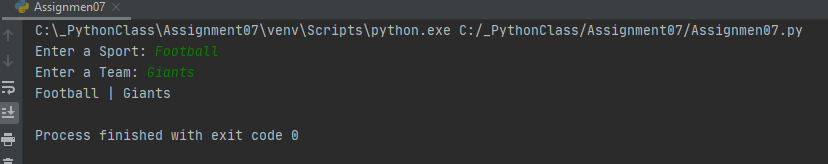


Figure 3

After testing the script in PyCharm, we then tested the script in the command terminal. In this case, the user inputted the sport as “basketball” and the team as “Knicks”. Figure 4 shows the script running in the command terminal:

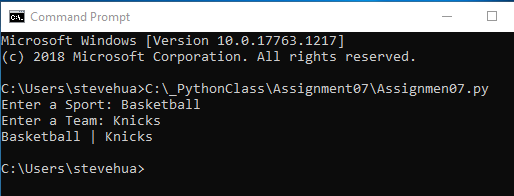


Figure 4

Lastly, we opened the team.dat file in notepad to confirm the list object has been stored. As you can see, the data is stored in semi-readable format:

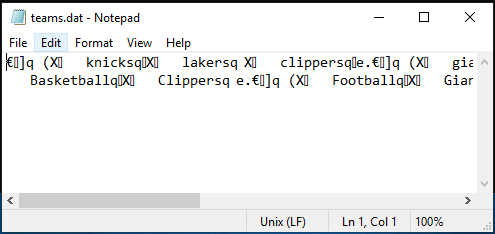


Figure 5

Structured Error Handling

Per [datacamp.com](https://www.datacamp.com/community/tutorials/exception-handling-python), “Python exceptions are errors that are detected during execution that are not unconditionally fatal.” An exception object is created when a script raises an exception and if the script doesn’t handle the exception, the program will be forced to terminate. Handling or trapping these exceptions can also be helpful in relaying detailed information to the user. Exceptions are captured in try-except blocks, ‘try’ will run a block of code in which an error is expected and ‘except’ will define this type of exception.

Using exception handling, I added a ‘try-except’ block to this script after implementing the pickling code. In the event that another user is running this program, one mistake I anticipate the user making is entering data as a number rather than a string. To prevent this from happening, I added a ‘try-except’ block in the presentation layer of the script. See Figure 6 below:



Figure 6

Here, when the user is prompted to enter a sport (the first part of the list object to be stored), the user’s data then run’s through a ‘try’ statement. If the user’s data is numeric, an exception is raised and the program prints a custom error message. Figure 7 below shows the outcome of this script in PyCharm when this exception is raised:

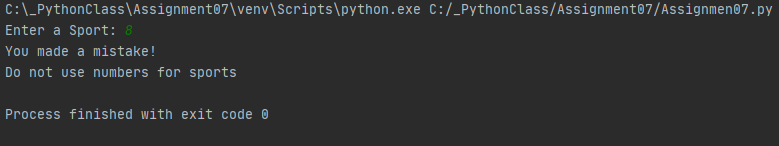


Figure 7

After testing the script in PyCharm, I then ran the script in the command terminal to confirm it worked as a console application. Figure 8 shows the program running successfully in the command terminal:

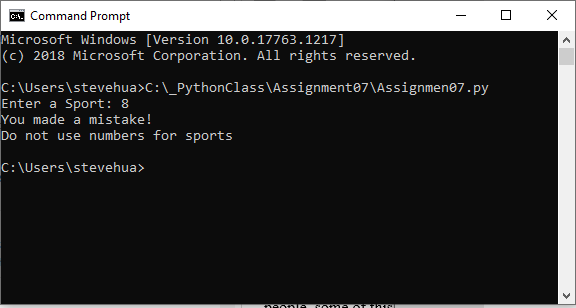


Figure 8

Conclusion

In summation, I was able to demonstrate examples of pickling and error handling and combine them in this script. First, I used pickling to store data on a .dat file in binary format and then load data from a .dat file and convert into its original readable format. Then, I used exception handling can be used to trap potential errors and ensure users enter the correct data.