Dustin Strop

Mar 6, 2020

Foundations of Programming: Python

Assignment07

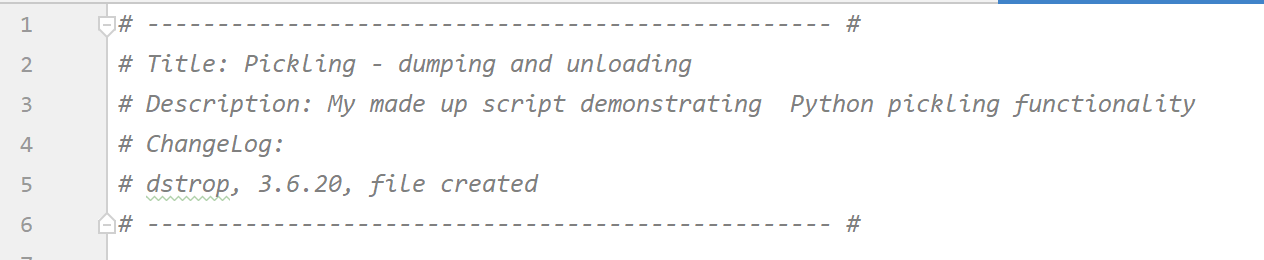
# An Explanation of Pickling and Error Handling in Python

Introduction

This document intends to cover how Python’s pickling works as well as an example of unpickling, which is just one capability of Python picking. The document will also cover Python error handling. Python language allows developers to essentially create their own errors that will raise to the end user (or other developers) and present a clear message of why something is wrong with the input value or the script. This can also help prevent the script from crashing.

Topic I: Header

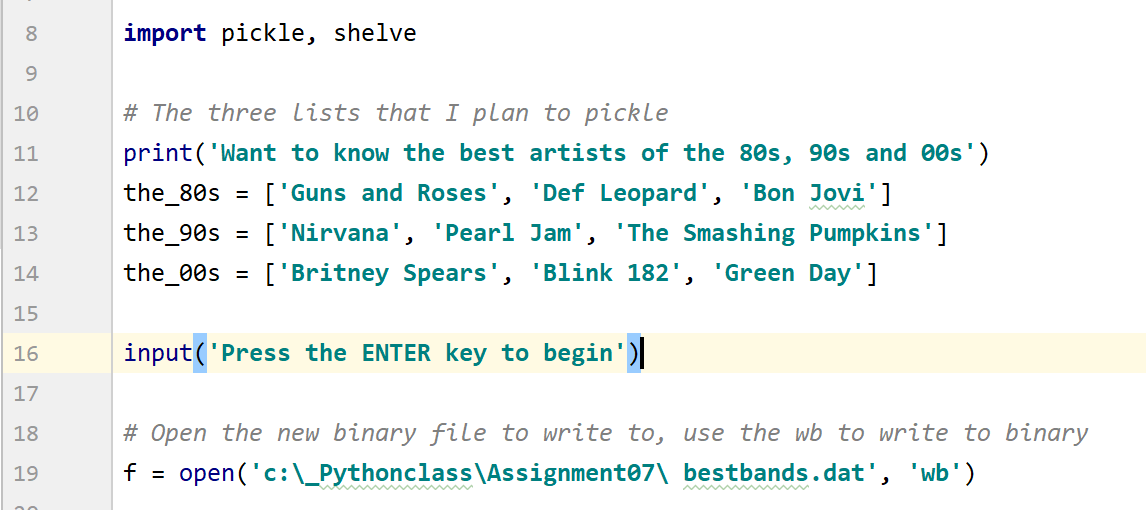
As always, the script must begin with the header. In this example, the title provide the two pickling capabilities that will be used in the script, pickle.dump() and pickle.load().



***Figure 1: Example header is short and to the point***

Topic II: Importing pickle function and pickling to a file

To use the pickle function the script must first import pickle. Line 8 on Figure 2 shows what this looks like in the script. The shelve function is also used in conjunction with the pickle function and allows the user to store and randomly access pickled objects in the file.



***Figure 2: an example of importing pickle and writing data to a file***

The program starts and prints the print statement on line 11 to the user. Lines 12, 13 and 14 don’t necessarily need to be in order in the code, but it is fine that they are. They are the declared variables and could also be placed after line 16’s input function, or before the import on line 8.

Topic III: Writing pickled data to a binary file

Pickled data can only be written to a binary file. A binary file is smaller in size so if the user is working large amounts of data, this is helpful. The binary file is also encrypted to a degree, and when opened it is not easily readable from the binary file. It reads as expected when ran using Python. Line 19 in Figure 2 is creating and writing to a binary file. Line 19 reads open the binary file bestbands.dat and write to it. ‘wb’ is the mode used when writing to a binary file. This is then assigned the value of f.

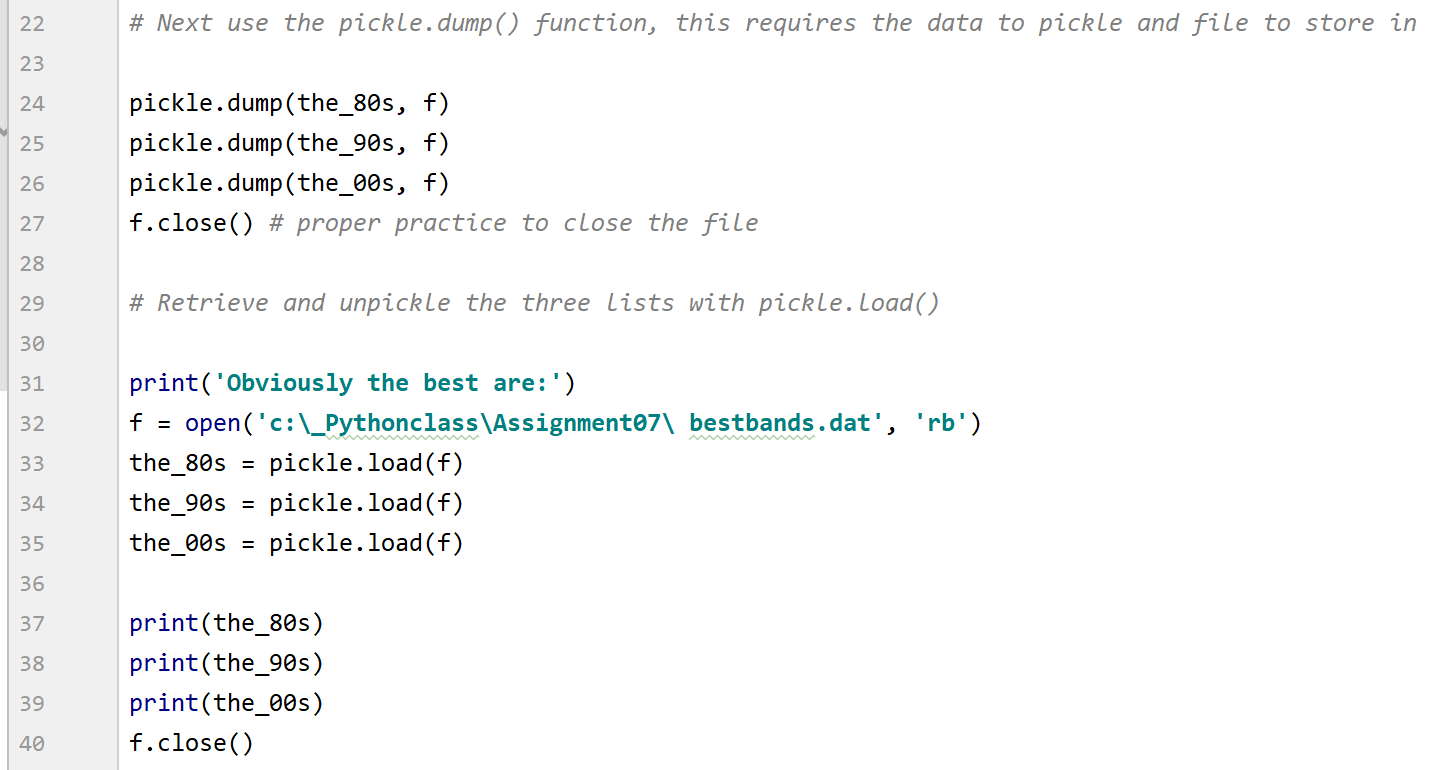
Topic III: Pickle dump and pickle unload

Subtopic: pickle dump

The script uses the pick.dump() function. What this does is essentially takes the users data and stores it in a dictionary or list, or other complex format in the file, assigned as f. Figure 3 shows the script using pickle.dump to pick up “the\_80s” “the\_90s” and “the\_00s” and storing them in the file, f. It is then proper coding to close the file. Using the close() function, as shows on line 27.

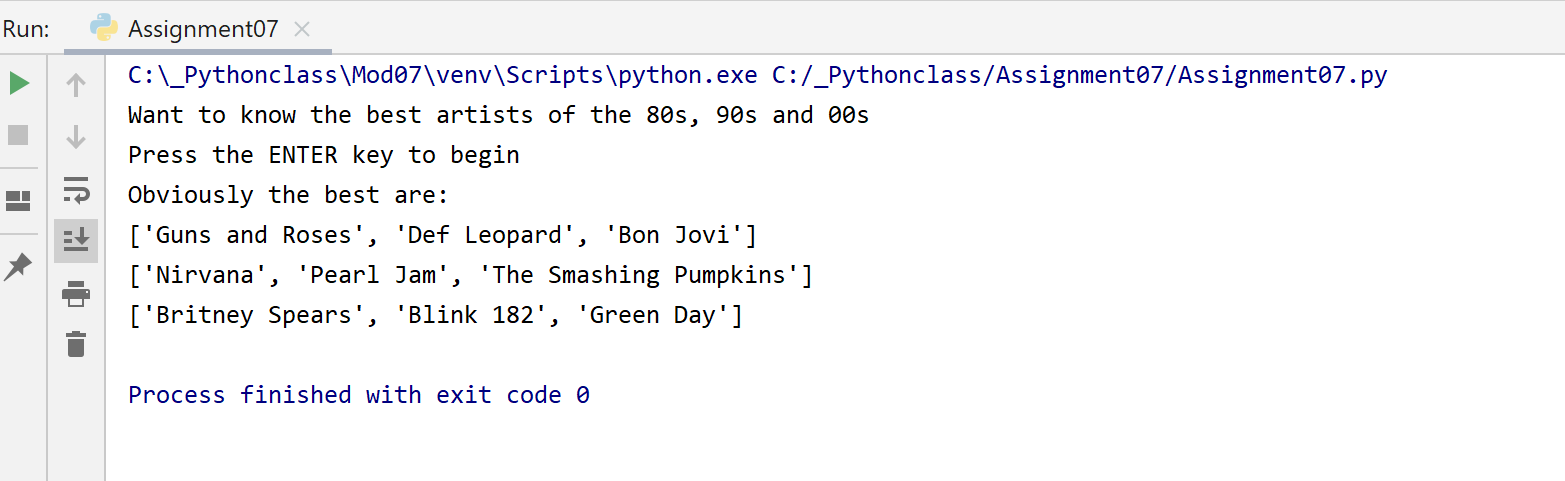
Subtopic: pickle load

After the user hits the Enter key to continue, the script will run the actions this document has just covered, plus it will then print the statement on line 31 and use the pickle load function to load the data from the save binary file. In this example the different variables are then printed.



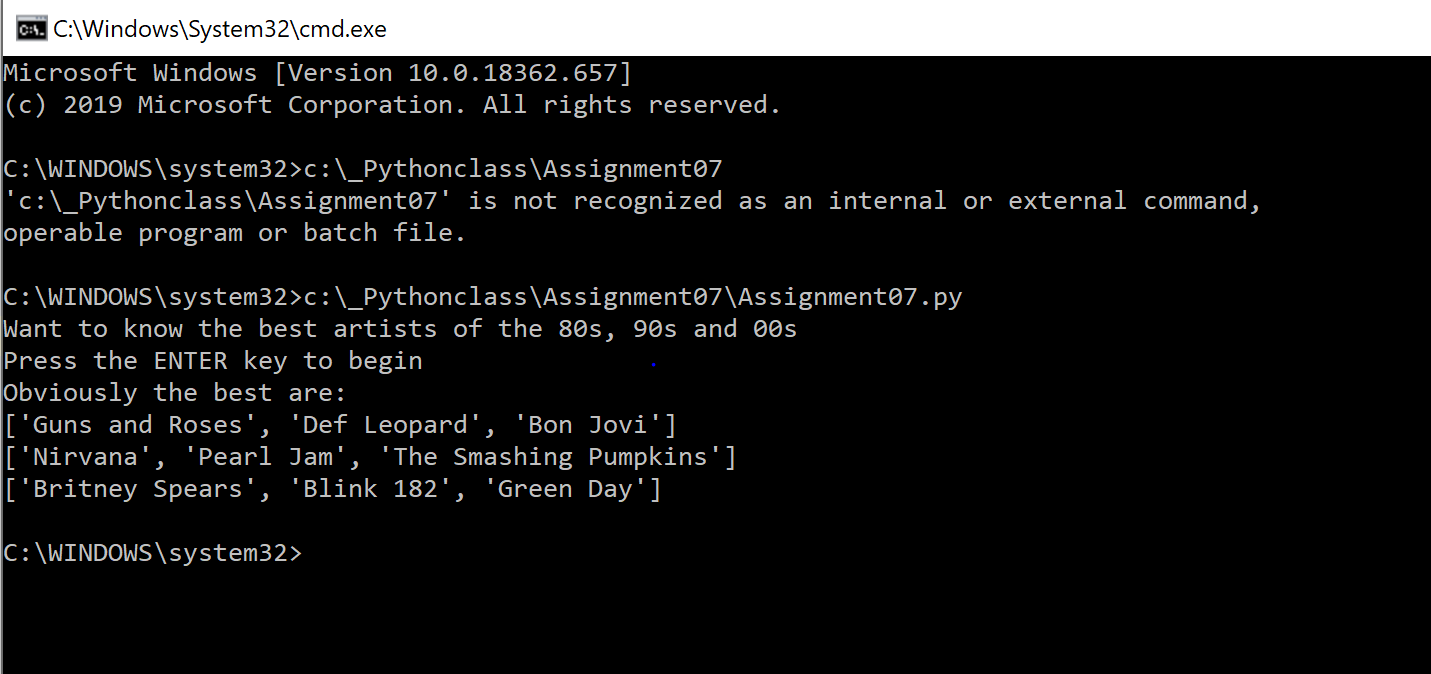
***Figure 3: an example of pickle.dump() and pickle.load()***

So far, this document has covered the scripts coding. From the user perspective the experience should look like the image in Figure 4.



***Figure 3: the above script run in Python window***

When this script is run the command window it should perform like the command window image below.



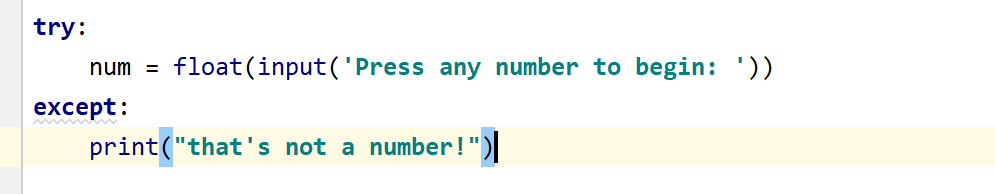
***Figure 5: An example of the command window reading a binary file of pickled data and presenting it***

Topic IV: Error Handling

There are several types of Exception Types in Python that help users and developers understand their code. This document will highlight the try Statements with except Clause.

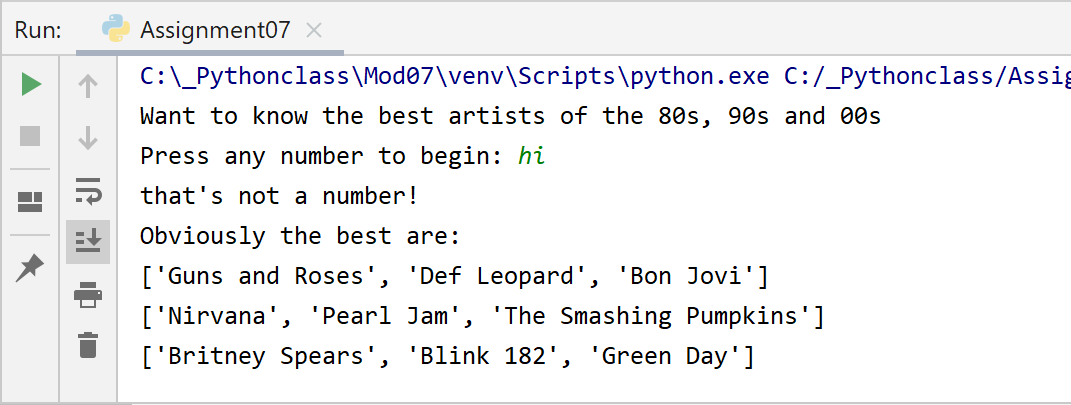
Sub-Topic III: try Statement with except Clause.

In this example a try Statement and except Clause is inserted into the code above. Line 17 has been changed to be a flout value and has been assigned the value of num, short for number.



***Figure 6: An example of the try/except error handling***

The code above essentially reads that if the user does not enter a float value, then display the print statement. This error handling helps the user know exactly what when wrong with their interaction with the code. The code will still execute but the user will receive the message displayed in Figure 7 below because they entered the string “hi” instead of a number.



***Figure 6: An example of the try/except error handling script ran in the Python window***

Topic IV: Summary

This code was generated to demonstrate how pickling data can be useful to users. As discussed, it can store data in a smaller file size and can also store complex data in dictionaries and lists. It can be used to pickup and unload data for users and has various other capabilities that can be leveraged. This document also covered basic structured error handling which is a feature that allows developers to create errors that allow their users to understand why the code is executing as expected and is delivered in a more understandable format. We also ran these scripts in the command window.