The Pickle Module

The `pickle` module allows you to save Python data structures (such as dictionaries) in a serialized form. By saving the serialized data structure in a file, the exact instance can be easily loaded and used later. To use the module, you will need to import `pickle` by including this line at the top of your program:

```
import pickle
```

"Pickling" is the process of using `pickle` to convert a Python object into a stream of bytes. The data must then be "unpickled" to again be used normally. There are two main operations you will need to know to use `pickle` successfully:

- `pickle.dump(object, file)` pickles an `object` and writes it to the file stream `file`.
- `pickle.load(file)` retrieves and unpickles the first object read from the file stream `file`.

Required Readings and Tutorials:

- How to Pickle: A Pickling and Unpickling Tutorial. This article will help you learn how to use the `pickle` module in Python 3, with some examples.
- Pickle in Python: Object Serialization. This is a more in-depth read that will teach you all the basics, as well as provide a comparison to JSON files.
- The Official Documentation will give you more information on exactly what you can do with the `pickle` module. Pay attention to the three types of errors you may encounter when pickling data.

Optional:

- What's so dangerous about pickles? An advanced read, this article explains how Python serialization can be dangerous and why you might want to be careful. The bottom line: do not unpickle untrustworthy data! There is no built-in security to protect you from malicious objects.

Videos:

- If you learn better by watching:
  - Pickling Data With Python! (7 min). Teaches you how to pickle and unpickle data and also gives an important message about safety when unpickling data.
  - Python Pickle Module for saving objects (serialization) (8 min). Another clear explanation of `pickle` that also motivates why you might need the module.

Practice Exercises

1. To be introduced to how the Pickle module works, it suffices to implement a very simple example. In a new directory, create a file called `dump.py`.
   (a) Open the file and create a data structure that can be pickled (see the possibilities here).
   (b) Then pickle the data structure and store it in a file called "data".
   (c) Open the `data` file to see the way the pickled object is stored.
   (d) Create a new file called `grab.py` and use it to retrieve, unpickle, and print the data structure stored in `data`.
2. As an extension, try to save and load two objects to the file rather than one. Does the second object get appended to the file, or does it overwrite the original contents? Does this change if you open the data file stream in write-append ("a") mode?