Serializing Python Objects to be Read by Other Languages

The data format used by the `pickle` module is Python-specific. It makes no attempt to be compatible with other programming languages. If cross-language compatibility is one of your requirements, you need to look at other serialization formats. One such format is JSON. "JSON" stands for "JavaScript Object Notation," but don't let the name fool you — json is explicitly designed to be usable across multiple programming languages.

Python 3 includes a `json` module in the standard library. Like the pickle module, the `json` module has functions for serializing data structures, storing the serialized data on disk, loading serialized data from disk, and unserializing the data back into a new Python object. But there are some important differences, too. First of all, the JSON data format is text-based, not binary. RFC 4627 defines the json format and how different types of data must be encoded as text. For example, a boolean value is stored as either the five-character string `'false'` or the four-character string `'true'`. All JSON values are case-sensitive.

Second, as with any text-based format, there is the issue of whitespace. JSON allows arbitrary amounts of whitespace (spaces, tabs, carriage returns, and line feeds) between values. This whitespace is "insignificant," which means that JSON encoders can add as much or as little whitespace as they like, and JSON decoders are required to ignore the whitespace between values. This allows you to "pretty-print" your JSON data, nicely nesting values within values at different indentation levels so you can read it in a standard browser or text editor. Python's `json` module has options for pretty-printing during encoding.

Third, there's the perennial problem of character encoding. JSON encodes values as plain text, but as you know, there ain't no such thing as "plain text."

JSON must be stored in a Unicode encoding (UTF-32, UTF-16, or the default,

utf-8), and section 3 of RFC 4627 defines how to tell which encoding is being used.