Saving Data to a Pickle File

The `pickle` module works with data structures. Let's build one.

```
shell = 1
print (shell)
#1
entry = {}
entry['title'] = 'Dive into history, 2009 edition'
entry['article_link'] = 'http://diveintomark.org/archives/2009/03/27/dive-into-history-2009-e
entry['comments_link'] = None
entry['internal_id'] = b'\xDE\xD5\xB4\xF8'
entry['tags'] = ('diveintopython', 'docbook', 'html')
entry['published'] = True
import time
entry['published_date'] = time.strptime('Fri Mar 27 22:20:42 2009')
print (entry['published_date'])
#time.struct_time(tm_year=2009, tm_mon=3, tm_mday=27, tm_hour=22, tm_min=20, tm_sec=42, tm_wc
```

① Follow along in Python Shell #1.

② The idea here is to build a Python dictionary that could represent something useful, like an entry in an Atom feed. But I also want to ensure that it contains several different types of data, to show off the `pickle` module. Don't read too much into these values.

③ The `time` module contains a data structure (`struct_time`) to represent a point in time (accurate to one millisecond) and functions to manipulate time structs. The `strptime()` function takes a formatted string an converts it to a `struct_time`. This string is in the default format, but you can control that with format codes. See the time module for more details.

That's a handsome-looking Python dictionary. Let's save it to a file.

```
shell                              #①
```

```
import pickle
with open('entry.pickle', 'wb') as f:    #②


    pickle.dump(entry, f)                #③
```

① This is still in Python Shell #1.

② Use the `open()` function to open a file. Set the file mode to `'wb'` to open the file for writing in binary mode. Wrap it in a with statement to ensure the file is closed automatically when you're done with it.

③ The `dump()` function in the `pickle` module takes a serializable Python data structure, serializes it into a binary, Python-specific format using the latest version of the pickle protocol, and saves it to an open file.

That last sentence was pretty important.

- The `pickle` module takes a Python data structure and saves it to a file.
- To do this, it *serializes* the data structure using a data format called "the pickle protocol."
- The pickle protocol is Python-specific; there is no guarantee of cross-language compatibility. You probably couldn't take the `entry.pickle` file you just created and do anything useful with it in Perl, PHP, Java, or any other language.
- Not every Python data structure can be serialized by the `pickle` module. The pickle protocol has changed several times as new data types have been added to the Python language, but there are still limitations.
- As a result of these changes, there is no guarantee of compatibility between different versions of Python itself. Newer versions of Python support the older serialization formats, but older versions of Python do not support newer formats (since they don't support the newer data types).
- Unless you specify otherwise, the functions in the `pickle` module will use the latest version of the pickle protocol. This ensures that you have maximum flexibility in the types of data you can serialize, but it also means that the resulting file will not be readable by older versions of Python that do not support the latest version of the pickle protocol.
- The latest version of the pickle protocol is a binary format. Be sure to

open your pickle files in binary mode, or the data will get corrupted during writing.