XML: Another TLA or the Future?

http://java.sun.com/xml/tutorial_intro.html

- XML is eXtensible Markup Language
  - It's a w3c standard (what does this mean?)
    - http://www.w3.org/XML/
  - It's simple, flexible, portable, extensible, hierarchical, complicated (see simple)
  - Basis for XMP-RPC, SOAP, UDDI, WSDL
- Why do we need to understand XML?
  - It really is simple, and it really is powerful
  - It really is in-demand and one of the FOTDay

XML in brief

- Need some method to create XML
  - Exported by applications, e.g., iTunes
  - Edited by hand, this is a bad idea
  - Done programatically, see bullet 1

- XML can be validated with a DTD
  - Document Type Definition
  - Often provided by applications
  - Really necessary for complex XML data

What does it look like?

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Major Version</key><integer>1</integer>
        <key>Minor Version</key><integer>1</integer>
        <key>Application Version</key><string>6.0.1</string>
        <key>Features</key><integer>1</integer>
        <key>MusicFolder</key>
        <string>file://localhost/Users/ola/Music/iTunes/iTunes%20Music/</string>
        <key>Library Persistent ID</key><string>2D2047AF6D8A9673</string>
        <key>Tracks</key>
        <dict>
                <key>40</key>
                <dict>
                        <key>Track ID</key><integer>40</integer>
                        <key>Name</key><string>Track 01</string>
                        <key>Kind</key><string>AAC audio file</string>
```

Elements of XML

- There's an XML header, specifies global "stuff"
- There's the content
  - Root element
  - Sub elements
- What is an element?
  - Tagged, case-sensitive, semantic
  - Can contain text, data, attributes

```
<project name="foo" type="java">
    <file>Glob.java</file>
    <file>Blib.java</file>
</project>
```

Handling XML

- There are two ways to parse/read XML
- SAX: Simple API for XML
  - Stream based, register callbacks that are called when certain tags/data parsed
  - Can't move around, but can read and discard
- DOM: Document Object Model
  - Entire tree-like structure read into memory
  - Easy to move around

Saving and restoring objects

- Classes should implement Serializable, this is a tag interface, not necessary to implement a function (see Cloneable)
  - mark non-serializable fields as *transient*
    - platform specific objects like font sizes, these need to be reconstructed rather than re-read
    - fields that aren't needed when an object is deserialized
  - use ObjectOutputStream and ObjectInputStream, can customize behavior using private?! functions below

```
private void writeObject(java.io.ObjectOutputStream out) throws
   IOException

private void readObject(java.io.ObjectInputStream in) throws
   IOException, ClassNotFoundException;
```

  - also `defaultReadObject()` and `defaultWriteObject()`