XML and HTML

Michael Li
Email: jwl@cs.nott.ac.uk

What is XML?

• XML stands for EXtensible Markup Language
• XML is a markup language much like HTML
• XML was designed to store and transfer data
• XML tags are not predefined. You must define your own tags
• XML is designed to be self-descriptive
• XML is a W3C Recommendation

What is HTML?

• HTML stands for Hyper Text Markup Language
• HTML is not a programming language, it is a markup language
• HTML was designed to display data
• A markup language is a set of markup tags
• HTML uses markup tags to describe web pages

XML documents are Trees

• XML has nothing to do with pointy brackets...
• At their heart, XML documents describe a tree of nodes
• Root is the Document node
• Under this is an Element node
• Elements can have Elements and Text nodes as children
• Composite Pattern — everything is a Node
The Difference Between XML and HTML

- XML is not a replacement for HTML.
- XML and HTML were designed with different goals:
  - XML was designed to transport and store data, with focus on what data is
  - HTML was designed to display data, with focus on how data looks
- HTML is about displaying information, while XML is about carrying information.

XML history

- Meta language: a language for creating other languages.
- SGML:
  - --> HTML (HyperText Markup Language)
  - --> XML (eXtensible Markup Language)

HTML Tags

- HTML markup tags are usually called HTML tags
- HTML tags are keywords surrounded by angle brackets like <html>
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- Start and end tags are also called opening tags and closing tags

HTML Documents

- HTML Documents = Web Pages
- HTML documents describe web pages
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages
- The purpose of a web browser (like Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML example

```html
<html>
<head>
<title>xml how to program</title>
</head>
<body>
<p>welcome to new web site!</p>
</body>
</html>
```
HTML Basic Document

```html
<html>
<head>
<title>Title of document goes here</title>
</head>
<body>
<h1>My First Heading</h1>
<p>Visible text goes here...</p>
</body>
</html>
```

- The text between `<html>` and `</html>` describes the web page
- The text between `<body>` and `</body>` is the visible page content
- The text between `<h1>` and `</h1>` is displayed as a heading
- The text between `<p>` and `</p>` is displayed as a paragraph

Heading Elements

- `<h1>` Largest Heading</h1>
- `<h2>` . . . </h2>
- `<h3>` . . . </h3>
- `<h4>` . . . </h4>
- `<h5>` . . . </h5>
- `<h6>` Smallest Heading</h6>

Text Elements

- `<p>` This is a paragraph</p>
- `<br>` (line break)
- `<hr>` (horizontal rule)
- `<pre>` This text is preformatted</pre>
- `<em>` This text is emphasized</em>
- `<strong>` This text is strong</strong>
- `<code>` This is some computer code</code>
- `<b>` This text is bold</b>
- `<i>` This text is italic</i>

Links

- Ordinary link:
  > `<a href="http://www.example.com/">Link text goes here</a>`
- Image-link:
  > `<img src="URL" alt="Alternate Text" /></a>`
- Mailto link:
  > `<a href="mailto:webmaster@example.com">Send e-mail</a>`

Other Elements

- Tables: `<table> </table> ...`
- List: `<ul> </ul>`, `<ol> </ol> ...`
- Forms: `<form> </form> ...`
- Frames: `<frame> </frame> ...`
- Entities: `&entity_name;`, `&#entity_number;`
- Advanced: Script, Doctypes, Media, ...

HTML Tutorial: http://www.htmlcodetutorial.com/
XML

- With XML You Invent Your Own Tags
  - The tags in a XML document are not defined in any XML standard. They are “invented” by the author of the XML document.
  - That is because the XML language has no predefined tags.
  - The tags used in HTML are predefined. HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.).
  - XML allows the author to define his/her own tags and his/her own document structure.

XML does not do anything

- XML was created to structure, store, and transport information.
  - <note>
    <to>Tom</to>
    <from>Jerry</from>
    <heading>Reminder</heading>
    <body>Don’t forget me this weekend!</body>
  </note>

  - This XML document does not DO anything. It is just information wrapped in tags. Someone must write a piece of software to send, receive or display it.

XML syntax

- All XML Elements Must Have a Closing Tag
- XML Tags are Case Sensitive
- XML Elements Must be Properly Nested
- XML Documents Must Have a Root Element
- XML Attribute Values Must be Quoted

XML Elements Must be Properly Nested

- In HTML, you might see improperly nested elements:
  - <b><i>This text is bold and italic</i></b>

- In XML, all elements must be properly nested within each other:
  - <b><i>This text is bold and italic</i></b>

XML attributes

- XML elements can have attributes, just like HTML. Attributes provide additional information about an element.
  - Attribute values must always be quoted. Either single or double quotes can be used.
  - Attributes are difficult to read and maintain.

XML Attribute Values Must be Quoted

HTML code:
<note date="12/11/2007">
  <to>Tom</to>
  <from>Jerry</from>
  <heading>Reminder</heading>
  <body>Don’t forget me this weekend!</body>
</note>

<note date="12/11/2007">
  <to>Tom</to>
  <from>Jerry</from>
</note>
XML Validation

- Well Formed XML Documents:
  - A "Well Formed" XML document has correct XML syntax.
- Valid XML Documents:
  - A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD):
Summary

- Markup language
  - markup tags
  - Syntax
- HTML
- XML

Readings

- "SAX, the power API" (Benoît Marchal, developerWorks, August 2001): Learn about when to use the SAX API instead of DOM, plus get an overview of commonly used SAX interfaces and detailed examples in a Java-based application with many code samples.

- "Simplify XML programming with JDOM" (Wes Biggs and Harry Evans, developerWorks, May 2001): Explore an alternate object model API that is optimized for the Java language.