Session 15

XML

XML Reading and Reference

- Reading
  - XML in a Nutshell (Ch. 1-3), available in Safari On-line
  - JavaWorld XML tutorial:
    www.javaworld.com/javaworld/jw-04-1999/jw-04-xml_p.html
  - Book chapter, describes how to design an xml document that will later be used to generate html
    www.phptr.com/articles/article.asp?p=170571&seqNum=1
- Reference:
  - W3C General spec of Document Object Model-
    www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/
  - XML Glossary -
    www.javaworld.com/javaworld/jw-09-2002/jw-0927-xmlglossary.html

Lecture Objectives

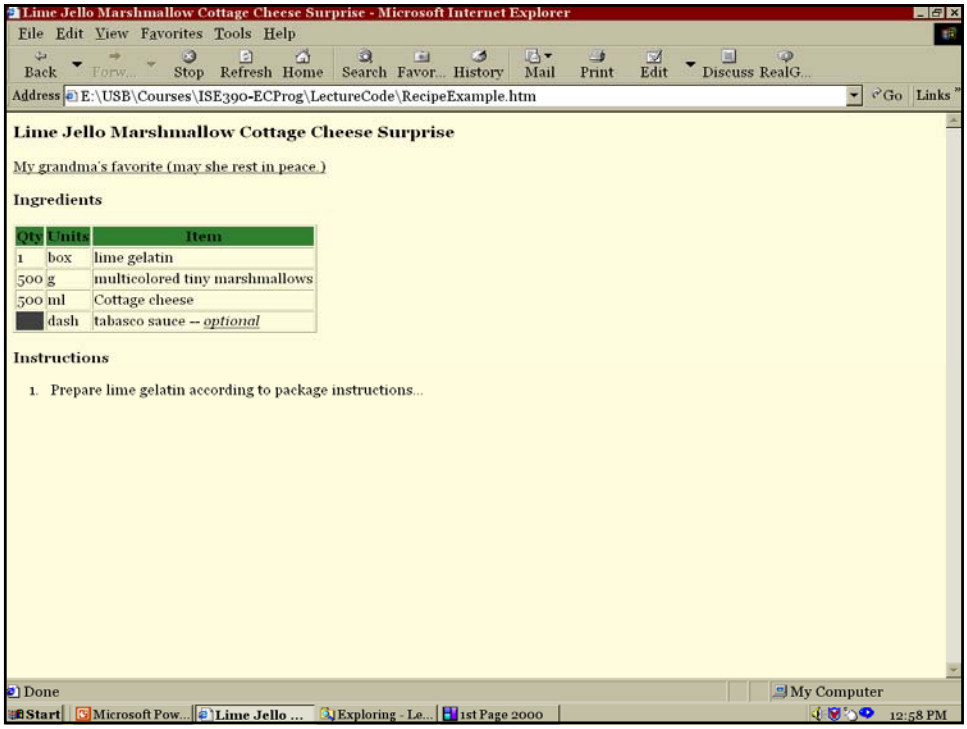
- Understand the goal of application specific markup languages
- Understand XML as a meta language that defines application specific languages
- Understand general concept of tree-structured access to an XML document
- Understand need to dynamically modify XML (e.g., xhtml) during browser rendering
- Be familiar with DTDs as a way of defining the rules of an XML document

 3

HTML Limitations

- Not extensible (unless you are Microsoft)
- Intended for display (not processing)
- Appearance information is intertwined with content information
- Working HTML is rarely syntactically correct
- No way to specify the meaning of data

 4



Example - HTML Code

```
<HTML> <HEAD>
<TITLE>Lime Jello Marshmallow Cottage Cheese Surprise</TITLE>
</HEAD> <BODY>
 <H3>Lime Jello Marshmallow Cottage Cheese Surprise</H3>
 <U>My grandma's favorite (may she rest in peace.)</U>
 <H4>Ingredients</H4>
 <TABLE BORDER="1">
<TR BGCOLOR="#308030"> <TH>Qty</TH> <TH>Units</TH>
<TH>Item</TH></TR>
 <TR> <TD>1</TD> <TD>box</TD> <TD>lime gelatin</TD></TR>
 <TR> <TD>500</TD> <TD>g</TD> <TD>multicolored tiny
marshmallows</TD></TR>
<TR> <TD>500</TD> <TD>ml</TD> <TD>Cottage cheese</TD></TR>
<TR> <TD BGCOLOR="#404040"/> <TD>dash</TD> <TD>tabasco sauce
<SPAN> -- <i><u>optional</u></i></SPAN></TD></TR>
</TABLE>
<H4>Instructions</H4>
<OL> <LI>Prepare lime gelatin according to package
instructions...</LI> </OL> </BODY> </HTML>
```

Recipe - Tasks That Would be Difficult

- Directly manipulate the recipe object in your server program to:
  - Aggregate information about multiple recipes
  - Build a shopping list
  - Convert from grams/liters to ounces/quarts

 7

Electronic Commerce Vision

- New EC applications are possible when all applications on all platforms interact
  - Product finders
  - Billing data (invoices, purchase orders)
  - Medical records
- Possible when applications are interacting through standard data exchange rather than DB / application method access

 8

Simple Recipe as an XML Document

```
<?xml version="1.0"?>
<!DOCTYPE Recipe SYSTEM "recipe.dtd">
<Recipe>
  <Name>Lime Jello Marshmallow Cottage Cheese Surprise</Name>
  <Description> My grandma's favorite (may she rest in peace).
   </Description>
  <Ingredients>
    <Ingredient>
      <Qty unit="box">1</Qty>
      <Item>lime gelatin</Item>
    </Ingredient>
    <Ingredient>
      <Qty unit="g">500</Qty>
      <Item>multicolored tiny marshmallows</Item>
    </Ingredient>
</Ingredients>
  <Instructions>
    <Step>Prepare lime gelatin according to package instructions
   </Step>
    <!-- And so on... -->
  </Instructions>
</Recipe>
```

Notice that the element names and attribute names refer to recipes

9

Well-Formed (Parsable) XML

- How does parsable XML help you write Electronic Commerce applications?
- Basic Rules (common to all XML documents)
  - No unclosed tags
  - No overlapping tags
  - Attribute values must be enclosed in quotes
  - The text characters >, <, and " must always be represented by character entities
- Extended rules (Specific to each XML application)

10

XML Document

- Structures textual information
- Does not contain styling information
- Defines a hierarchical structure
- Contains elements and attributes
- Follows basic XML syntax rules
- Usually adheres to a set of domain rules
  - Element names
  - Attribute names
  - Containment rules

11

XHTML

- Extensible Hypertext Markup Language
- An official W3C recommendation
- Designed to bring the structure and accuracy of XML to HTML
- If an HTML page conforms to an XML DTD you can:
  - Easily extract information
  - Ensure consistent display
  - Convert to other markup languages (i.e., device specific languages)

12

XHTML Syntax ...

- Conforms to XML syntax rules (embedding, null tags, etc.)
- Very similar to HTML 4.01 (Strict, Transitional ...)
- Major differences:
  - Elements must be properly nested
  - Documents must be well-formed
  - Tag names and attribute names must be in lower case
  - All elements must be closed
  - Attribute values must be quoted

 13

... XHTML Syntax ...

- Attribute minimization is forbidden

```
<dl compact>          <dl compact="compact">
<input checked>       <input checked="checked">
<input readonly>  →   <input readonly="readonly">
<input disabled>      <input disabled="disabled">
<option selected>     <option selected="selected">
<frame noresize>      <frame noresize="noresize">
```

 14

Application-Specific XML Rules

- Rules define each unique XML language (e.g. the simple recipe language)
- Examples of document rules:
  - Names of the elements and attributes
  - Rules for the maximum and minimum number of ingredients in a recipe
  - Rules for the maximum and minimum number of quantities in an ingredient
- Defined in a schema
  - DTD (Document Type Definition)
  - XML Schema

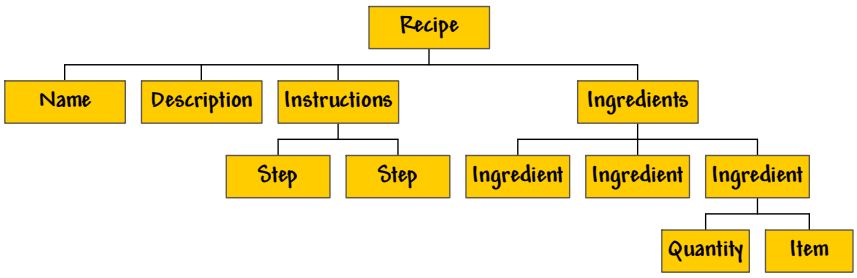
 15

Simple Recipe DTD

```
<!ELEMENT Recipe (Name, Description?, Ingredients?,
   Instructions?)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Ingredients (Ingredient)*>
<!ELEMENT Ingredient (Qty, Item)>
<!ELEMENT Qty (#PCDATA)>
<!ATTLIST Qty
   unit CDATA #REQUIRED
>
<!ELEMENT Item (#PCDATA)>
<!ATTLIST Item
   optional CDATA "0"
   isVegetarian CDATA "true"
>
<!ELEMENT Instructions (Step)+>
<!ELEMENT Step (#PCDATA)>
```

 16

The Simple Recipe as a Tree



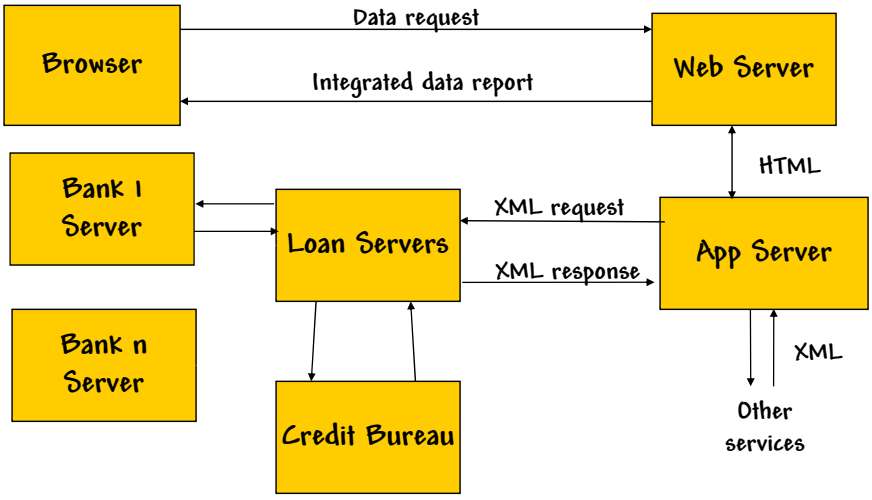
 17

XML as a Data Exchange Standard

- XML documents are often transformed into
  - HTML (XHTML)
  - XML
  - Java (and other languages) object

 18

Case - Banking Use of XML



 19

Document Object Model (DOM)

- Hierarchical object representation of an XML or HTML document
  - Produced by XML parsers
- Your Java/JavaScript program can
  - Extract a given node (element)
  - Walk the tree
  - Search for particular nodes or data (e.g., img tags)
  - Modify the nodes
  - Generate a new document as
    - A DOM object
    - An XML text file

You will use DOM for the Ajax part of the course

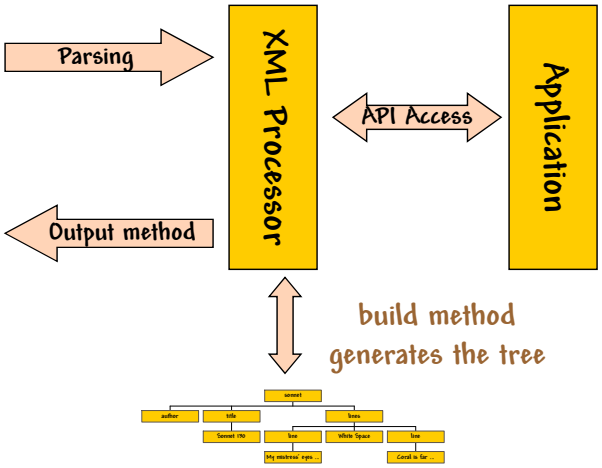
You will also use JQuery to access DOM elements

You will use a JSP tag library to access an XML document, but you should understand the concepts of DOM for browser HTML changes

 20

XML DOM

```
<?xml version="1
<!DOCTYPE sonne
<sonnet type="S
  <author>
    <last-name>S
    <first-name>
    <nationality
    <year-of-bir
    <year-of-dea
  </author>
  <title>Sonnet
  <lines>
    <line>My mist
    <line>Coral i
    <line>If snow
    ...
```



21

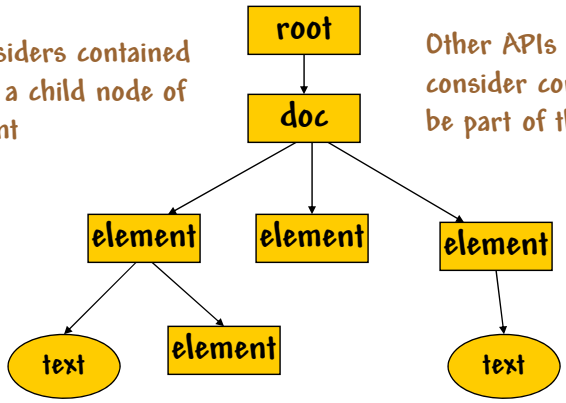
Document Validity

- Well-formed - follows the rules of XML
- Valid - Corresponds to the specified schema

22

Access a DOM Tree

DOM considers contained text to be a child node of the element



Other APIs (e.g., JDom) consider contained text to be part of the element

 23

XML Applications

- Extending a Web application requires agreement among the trading partners
  - Structure of XML document
  - Names of elements and attributes
- Usually defined as a
  - Company standard - e.g., credit reports
  - Industry standard - e.g., HL7
- The standard is defined as a schema, using one of may languages (e.g., DTD)

 24

XML Schema (XSchema)

- W3C standard
- Individual schemas define a class of XML documents (a schema file usually has an .xsd extension)
- An individual document that conforms to a particular schema is called an instance document

Example - DTD/Schema

```
<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

DTD

Corresponding schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://www.w3schools.com"
   xmlns="http://www.w3schools.com"
   elementFormDefault="qualified">
<xs:element name="note">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="to" type="xs:string"/>
         <xs:element name="from" type="xs:string"/>
         <xs:element name="heading" type="xs:string"/>
         <xs:element name="body" type="xs:string"/>

</xs:sequence> </xs:complexType> </xs:element></xs:schema>
```

root

Namespace declaration

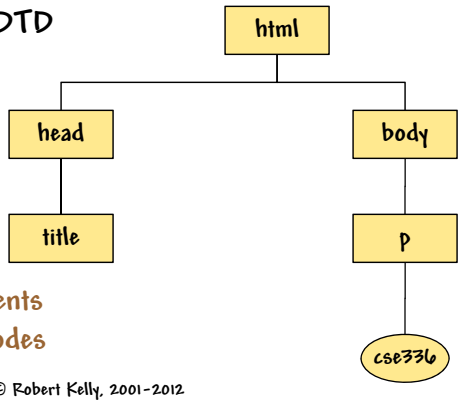
Corresponds to namespace declaration in XML document

Are We on Track?

- For the document tree below
  - Write a DTD
  - Write a sample xml document that will be valid according to the DTD

Try to use XML Spy or EditiX if you can download and install it quickly

head and body elements are required



Rectangles represent elements and ovals represent text nodes

Were We on Track?

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT html (head, body)>
<!ELEMENT head (title?)>
<!ELEMENT body (p?)>
<!ELEMENT p (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html SYSTEM "Track-XML.dtd">
<html>
   <head/>
   <body>
   <p>cse336</p>
   </body>
</html>
```

Have You Satisfied the Lecture Objectives?

- Understand the goal of application specific markup languages
- Understand XML as a meta language that defines application specific languages
- Understand general concept of tree-structured access to an XML document
- Be familiar with DTDs as a way of defining the rules of an XML document

Assignment 6

- Prepare an XML document containing a "typical" set of responses to the questions in the project form
  - Prepare a schema (using either DTD or XML Schema) using an XML tool (XML Spy or EditiX)
  - The inclusion rules in your schema should be consistent with the meaning of the form elements (e.g., first name and last name are children of a name element)
  - Prepare the XML document consistent with the schema