CS 312
Internet Concepts

Web Programming/Scripting:
XML

Dr. Michele Weigle
Department of Computer Science
Old Dominion University
mweigle@cs.odu.edu

http://www.cs.odu.edu/~mweigle/CS312-F08/

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

What is XML?

- eXtensible Markup Language
  - » markup language, like HTML
- Designed to transport and store data
  - » unlike HTML (designed to display data)
- XML tags are user-defined
- XML doesn't *do* anything
  - » provides a way to structure and store information

http://www.w3schools.com/xml/default.asp

XML Example

```
<person>
        <name> Richard Smith </name>
        <address> 2004 Lakewood Drive
        Norfolk, VA 23556</address>
</person>
```

- XML is self-descriptive
  - » example describes a person, their name and address
  - » just identifies the structure of the *data*, not how it might be displayed
- XML is plain-text
- XML has no pre-defined tags

http://www.w3schools.com/xml/default.asp

XML and HTML

- XML separates data from HTML
  - » separates data from how it is displayed
- Example: Create a webpage to display data that's updated frequently
  - » with HTML only
    - ❖ edit HTML each time data changes
  - » with HTML and XML
    - ❖ create HTML layout once
    - ❖ only need to update XML with new data
  - » We'll use Javascript later to demonstrate this

http://www.w3schools.com/xml/default.asp

More Benefits of XML

- Simplifies data sharing
  - » XML is plain-text, so no complicated, proprietary format to parse (software and hardware independent)
  - » Allows for the creation of data that different applications can share
- Simplifies data transport
  - » XML can be used to exchange information between incompatible systems
- Simplifies platform changes
  - » XML is plain-text, so software or hardware changes won't affect the data

http://www.w3schools.com/xml/default.asp

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

<?xml?>

<xml/>

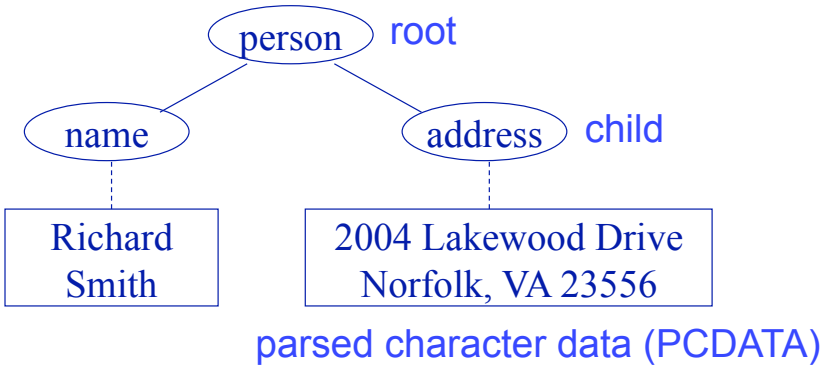
XML Document Structure

```
<?xml version="1.0" encoding="ISO-8859-1"?>   XML declaration
<person>      root element
        <name>Richard Smith </name>
        <address>2004 Lakewood Drive      child elements
        Norfolk, VA 23556</address>
</person>     end of root element
```

- XML declaration
  - » version and encoding
- Root element
- Child elements
  - » child elements can have their own child elements
- End of root element

XML Tree

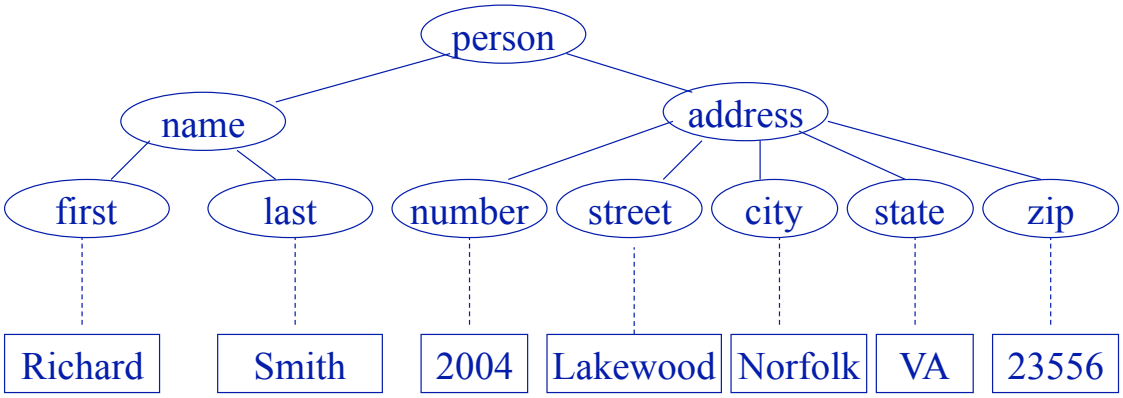
```
<person>
  <name>Richard Smith </name>
  <address>2004 Lakewood Drive
    Norfolk, VA 23556</address>
</person>
```



Another Representation

```
<person>
  <name> <first>Richard</first>
         <last>Smith</last>
  </name>
  <address> <number>2004</number>
         <street>Lakewood Drive</street>
         <city>Norfolk</city>
          <state>VA</state>
         <zip>23556</zip>
   </address>
</person>
```

Corresponding Tree Structure



XML Syntax Rules

- Similar to XHTML syntax
  - » XHTML is a combination of HTML and XML
- All elements must have a closing tag
- Tags are case-sensitive
  - » <name> is different than <Name>
- Tags must be properly nested
  - » <address><street>Lakewood</street></address>
- All XML documents must have a root element
- Attribute values must be quoted
  - » <address type="home"> or <address type='home'>
- Comments are the same as in HTML
- Whitespace is preserved (unlike HTML)

http://www.w3schools.com/xml/default.asp

XML Elements

- Everything from the element's start tag to its end tag, inclusive
- Can contain other elements, simple text, or a mixture
- Can have attributes

<address> has element contents

<state> has text content

```
<person>
  <name> <first>Richard</first>
         <last>Smith</last>
  </name>
  <address> <number>2004</number>
            <street>Lakewood Drive</street>
            <city>Norfolk</city>
            <state>VA</state>
            <zip>23556</zip>
   </address>
</person>
```

http://www.w3schools.com/xml/default.asp

XML Element Naming Rules

- Names can contain letters, numbers, or other characters
- Names must not start with a number or punctuation
- Names must not start with the letters 'xml'
- Names cannot contain spaces
- There are no reserved words
  - » any name can be used

http://www.w3schools.com/xml/default.asp

Good Naming Practices

- Make names descriptive
- Names should be short and simple
- Avoid '-' characters
  - » some software may interpret as minus
- Avoid "." characters
  - » some software may interpret as properties
- Avoid ":" characters
  - » colons are reserved in XML

http://www.w3schools.com/xml/default.asp

Attributes vs. Elements

- Attributes
  - » best used for *meta-data* (data about data)
  - » ex: ID numbers
- Elements
  - » best used for the any part of the data itself

```
<phone type="mobile">
  757-123-4567
</phone>
```

✗

```
<phone>
  <mobile>757-123-4567</mobile>
</phone>
```

✔

http://www.w3schools.com/xml/default.asp

Checking XML Syntax

- http://www.w3schools.com/xml/xml_validator.asp
  - » has a tool where you can provide the URL of an XML file and it will check that the XML is "well-formed"

Specifying XML Structure

DTD
(document type definitions)

```
<!DOCTYPE person [
  <!ELEMENT person (name, address)>
  <!ELEMENT name  (#PCDATA)>
  <!ELEMENT address  (#PCDATA)>
]>
```

```
<xs:element name="person">

<xs:complexType>
  <xs:sequence>
     <xs:element name="name"  type="xs:string"/>
     <xs:element name="address"  type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

XML Schema

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

<?xml?>

<xml/>

Viewing XML Files

- Because XML does not describe how to format the data, browsers will display XML files without much formatting
- root and child elements will be color-coded
- '+' or '-' will be displayed to the left of each element
  - » click on the '+' to expand it
  - » click on the '-' to collapse it
- To see the raw XML source, use the browser's 'View Source' option.

http://www.w3schools.com/xml/default.asp

Example XML Files

- http://www.w3schools.com/xml/note.xml
  - » sample XML file
- http://www.w3schools.com/xml/note_error.xml
  - » XML file with an error
- http://www.w3schools.com/xml/cd_catalog.xml
  - » CD catalog
- http://www.w3schools.com/xml/simple.xml
  - » restaurant menu

Styling XML with CSS

- Include a line in the XML file to load the CSS
  <?xml-stylesheet type="text/css" href="*filename*.css"?>
- The CSS tags should be the same as the XML element names
  - » http://www.w3schools.com/xml/cd_catalog.txt
- Example XML formatted with the CSS
  - » http://www.w3schools.com/xml/cd_catalog_with_css.xml
- The standards folks (W3C) recommend using XSLT instead of CSS for XML styling

http://www.w3schools.com/xml/default.asp

Styling XML with XSLT

- eXtensible Stylesheet Language Transformations
  - » more sophisticated than CSS
  - » transforms XML into HTML

- Include line in the XML to load the XSLT

  <?xml-stylesheet type="text/xsl" href="*filename*.xsl"?>

http://www.w3schools.com/xml/default.asp

XSLT

- Special tags in HTML that reference the XML elements
- Begins with version information

  <html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
- Select each element with
  - » <xsl:for-each select="*root/element*"> … </xsl:for-each>
- Inside the for-each loop, display sub-elements with
  - » <xsl:value-of select="*element*"/>

http://www.w3schools.com/xml/default.asp

Example: Plant Catalog

- http://www.cs.odu.edu/~mweigle/cs312/xml/plant_catalog.xml
- http://www.cs.odu.edu/~mweigle/cs312/xml/plants.xsl

http://www.w3schools.com/xml/default.asp

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

Parsing XML

- Most modern browsers have a built-in XML parser
  - » reads XML into memory
  - » converts it into an XML DOM (document object model) object that can be accessed with Javascript
  - » some differences between the parsers in IE and other browsers

http://www.w3schools.com/xml/default.asp

Parsing XML in IE

```
var xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
xmlDoc.async="false";
xmlDoc.load("note.xml");
```

- Create an empty Microsoft XML document object
- Turn off asynchronous loading
  - » make sure parser doesn't start until entire document loaded
- Load an XML document called "note.xml"

http://www.w3schools.com/xml/default.asp

Parsing XML in Other Browsers

```
var xmlDoc=new document.implementation.createDocument("", "", null);
xmlDoc.async="false";
xmlDoc.load("note.xml");
```

- Create an empty XML document object
- Turn off asynchronous loading
  - » make sure parser doesn't start until entire document loaded
- Load an XML document called "note.xml"

http://www.w3schools.com/xml/default.asp

XML DOM

- XML Document Object Model
  - » defines a standard way for accessing and manipulating XML documents
- Elements are accessed through a tree-structure

```
xmlDoc.getElementsByTagName("COMMON")[0].childNodes[0].nodeValue
```

xmlDoc - XML document created by the parser
getElementsByTagName("COMMON")[0] – the first <COMMON> element
childNodes[0] – the first child of the <COMMON> element (the text node)
nodeValue – the value of the node (the text itself)

http://www.w3schools.com/xml/default.asp

Cross-Browser Code

```
try //Internet Explorer
  {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  }
catch(e)
  {
  try //Firefox, Mozilla, Opera, etc.
    {
    xmlDoc=document.implementation.createDocument("","",null);
    }
  catch(e)
    {
    alert(e.message);
    return;
    }
  }
```

http://www.w3schools.com/xml/default.asp

Parsing Example

- Note that the XML and the HTML/Javascript must be on the same web server.

- Javascript function parseXML()
  - » loads plant_catalog.xml
  - » extracts first element
  - » inserts element information into the HTML
- HTML code to format how the element will look

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-parser.html

http://www.w3schools.com/xml/default.asp

Displaying All of the Data

- Use Javascript to loop through the elements

- Steps for another example:
  - » check browser and load XML using correct parser
  - » create an HTML table
  - » use getElementsByTagName() to get all XML PLANT nodes
  - » for each PLANT node, display data from COMMON and PRICE as table data

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-table.html

http://www.w3schools.com/xml/default.asp

XMLHttpRequest Object

- Provides a way to communicate with a server after a web page has loaded

- Allows
  - » update a web page with new data without reloading the page
  - » request data from a server after the page has loaded
  - » receive data from a server after the page has loaded
  - » send data to a server in the background

http://www.w3schools.com/xml/default.asp

Setting Up XMLHttpRequest

```
var xmlhttp;
function loadXMLDoc(url)
{
 xmlhttp=null;
 if (window.XMLHttpRequest)
 {// code for all new browsers
   xmlhttp=new XMLHttpRequest();
 }
 if (xmlhttp!=null)
 {
   xmlhttp.onreadystatechange=state_Change;
   xmlhttp.open("GET",url,true);
   xmlhttp.send(null);
 }
 else
 {
    alert("Your browser does not support XMLHTTP.");
 }
}
```

state changes from 0 (uninitialized) to 4 (loaded)

name of the function called when the state of the XMLHttpRequest object changes

http://www.w3schools.com/xml/default.asp

Setting Up XMLHttpRequest

```
function state_Change()
{
 if (xmlhttp.readyState==4)
 {// 4 = "loaded"
  if (xmlhttp.status==200)
  {// 200 = OK
    // ...our code here...
  }
  else
  {
    alert("Problem retrieving XML data");
  }
 }
}
```

HTTP response code

http://www.w3schools.com/xml/default.asp

Example

- Update data on page without reload
  - » use loadXMLDoc() function
  - » use stateChange() function with one line added

    document.getElementById('T1').innerHTML =
    xmlhttp.responseText;

```
<body onload="loadXMLDoc('test_xmlhttp.txt')">

<div id="T1" style="border:1px solid black; height:50; width:300; padding:10"></
div>

<p><button onclick="loadXMLDoc ('test_xmlhttp2.txt')">Click</button> </p>
</body>
```

http://www.cs.odu.edu/~mweigle/cs312/xml/text-noreload.html

http://www.w3schools.com/xml/default.asp

Example

- Display HTTP response headers
  - » change one line in state_Change() function

    document.getElementById('p1').innerHTML =
    xmlhttp.getAllResponseHeaders();

```
<p>getAllResponseHeaders() returns the HTTP response header upon loading a
URL.</p>
<p id="p1" style="white-space: pre"></p>

<button onclick="loadXMLDoc ('test_xmlhttp.txt')">Get Headers</button>
</body>
```

http://www.cs.odu.edu/~mweigle/cs312/xml/get-header.html

http://www.w3schools.com/xml/default.asp

Response Headers

- To get a particular response header, use getResponseHeader('*OptionName'*)
- Examples:
  - » xmlhttp.getResponseHeader('Last-Modified');
    - ❖ Wed, 05 Nov 2008 18:49:18 GMT
  - » xmlhttp.getResponseHeader('Content-Length');
    - ❖ 93

XML Application Example

- Display table of plants
- When click on table entry, display full information about that plant
  - » write show() Javascript function that displays the full information
  - » call show() when table clicked

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-app.html

http://www.w3schools.com/xml/default.asp

XML Application Example

- Same code as before to load the XML
- Add show() function

```
x = xmlDoc.getElementsByTagName (“PLANT”);

function show (i)
{
  common=(x[i].getElementsByTagName(”COMMON")[0].childNodes[0].nodeValue);
  botanical=(x[i].getElementsByTagName(”BOTANICAL")[0].childNodes[0].nodeValue);
  zone=(x[i].getElementsByTagName(”ZONE")[0].childNodes[0].nodeValue);
  light=(x[i].getElementsByTagName(”LIGHT")[0].childNodes[0].nodeValue);
  price=(x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue);

  txt=”Name: "+name+” (<i>”+botanical+”</i>)<br />Zone: "+zone+
      "<br />Light: "+light+"<br />Price: "+price;
  document.getElementById("show").innerHTML = txt;
}
```

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-app.html

http://www.w3schools.com/xml/default.asp

XML Application Example

- Create ‘show’ div section for the information to be placed in

```
<body>
<div id=“show”>
Click on one of the table rows to display the full plant information.
</div>
```

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-app.html

http://www.w3schools.com/xml/default.asp

XML Application Example

- Javascript to create the table

```
document.write("<table border='1'>");
for (var i=0;i<x.length;i++)
{
  document.write("<tr onclick='show(" + i + ")'>");
  document.write("<td>");
  document.write(x[i].getElementsByTagName("COMMON")[0].childNodes[0].nodeValue);
  document.write("</td>");

  document.write("<td>");
  document.write(x[i].getElementsByTagName("PRICE")[0].childNodes[0].nodeValue);
  document.write("</td>");
  document.write("</tr>");
}
document.write("</table>");
```

http://www.cs.odu.edu/~mweigle/cs312/xml/plant-app.html

http://www.w3schools.com/xml/default.asp

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

<?xml?>

<xml/>

XML Encodings

- Remember this?

  <?xml version="1.0" encoding="ISO-8859-1"?>
- ISO-8859-1 specifies a particular type of character encoding
  - » Latin alphabet (covers most Western languages)
  - » Each character is encoded with a single byte
- Other single-byte encodings
  - » Windows-1252, UTF-8
- Double-byte encodings
  - » UTF-16

http://www.w3schools.com/xml/default.asp

XML Encodings

- Windows Notepad saves files as single-byte ANSI (ASCII) by default
  - » You can use 'Save as…" to save as double-byte Unicode (UTF-16)
- Make sure that you
  - » always use the encoding attribute when you create XML files
  - » use an editor that supports encoding
  - » know what encoding your editor supports

http://www.w3schools.com/xml/default.asp

Real-Life XML Examples

- XMLNews
  - » specification for exchanging news and other information
  - » News Industry Text Format (NITF) – XML document type definition (DTD)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nitf>
 <head>
  <title>Colombia Earthquake</title>
 </head>
 <body>
  <headline>
   <hl1>143 Dead in Colombia Earthquake</hl1>
  </headline>
  <byline>
   <bytag>By Jared Kotler, Associated Press Writer</bytag>
  </byline>
  <dateline>
   <location>Bogota, Colombia</location>
   <date>Monday January 25 1999 7:28 ET</date>
  </dateline>
 </body>
</nitf>
```

http://www.w3schools.com/xml/default.asp

Real-Life XML Examples

- XML Weather Service
  - » Used by NOAA

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<current_observation>
 <credit>NOAA's National Weather Service</credit>
 <credit_URL>http://weather.gov/</credit_URL>
 <location>New York/John F. Kennedy Intl Airport, NY</location>
 <station_id>KJFK</station_id>
 <latitude>40.66</latitude>
 <longitude>-73.78</longitude>
 <observation_time_rfc822>Mon, 11 Feb 2008 06:51:00 -0500 EST
   </observation_time_rfc822>
 <weather>A Few Clouds</weather>
 <temp_f>11</temp_f>
 <temp_c>-12</temp_c>
 <relative_humidity>36</relative_humidity>
 <wind_dir>West</wind_dir>
 <wind_degrees>280</wind_degrees>
 <wind_mph>18.4</wind_mph>
 <wind_gust_mph>29</wind_gust_mph>
 <pressure_mb>1023.6</pressure_mb>
 <pressure_in>30.23</pressure_in>
 <dewpoint_f>-11</dewpoint_f>
 <dewpoint_c>-24</dewpoint_c>
 <windchill_f>-7</windchill_f>
 <windchill_c>-22</windchill_c>
 <visibility_mi>10.00</visibility_mi>
</current_observation>
```

http://www.w3schools.com/xml/default.asp

XML

- What is XML?
- XML Syntax
- Viewing XML
- XML and Javascript
- XML in Real Life

<?xml?>

<xml/>