HTML:
A Crash Course

Originals of Slides and Source Code for Examples:
http://courses.coreservlets.com/Course-Materials/ajax.html

For live Ajax & GWT training, see training courses at http://courses.coreservlets.com/.

Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

Topics in This Section

- **Differences between xhtml and HTML 4**
- **Differences between HTML 5 and HTML 4**
- **Basic structure of an xhtml document**
- **Hypertext links and URLs**
- **Block-level elements**
- **Inline elements**
- **Tables**
- **Forms**
- **CSS: thumbnail overview**
- **References**

4

Differences between xhtml and HTML 4

xhtml: Case

- **In HTML 4, case does not matter for tag names, attribute names, and predefined attribute values**
  - <BODY>, <Body>, and <body> are equivalent
  - <H1 ALIGN="..."> is equivalent to <H1 aLiGn="...">
  - <INPUT TYPE="TEXT"> is equivalent to <INPUT TYPE="text">
- **In xhtml, use lower case for tag names, attribute names, and predefined values**
  - <body>
  - <h1 align="...">
  - <input type="text" />

6

xhtml: Quotes

- **In HTML 4, quotes are optional if attribute value contains only alphanumeric values**
  - <H1 ALIGN="LEFT"> or
  - <H1 ALIGN=LEFT>

- **In xhtml, you must always use single or double quotes**
  - <h1 align="left"> or
  - <h1 align='left'>

7

xhtml: End Tags

- **HTML 4**
  - Some tags are containers
    - <H1>...</H1>, <A HREF...>...</A>
  - Some tags are standalone
    -
, <HR>
  - Some tags have optional end tags
    - <P>, <LI>, <TR>, <TD>, <TH>
- **XHTML**
  - All tags are containers. End tags always required.
    - <p>...</p>, <li>...
  - If there is no body content, start/end tags can be merged
    -
</br> →
      - An IE bug prevents this for script tags. Use <script ...></script> always.
      - Some people use
 (space before slash) for compatibility with very old browsers. But these old browsers won't support XMLHttpRequest anyhow, so this is less important in Ajax apps.

8

xhtml: Boolean Attributes

- **In HTML 4, you can use minimized notation for boolean attributes**
  - <option value="1" selected>...</option>
  - <dl compact>
- **In xhtml, you must write out the attribute values**
  - <option value="1" selected="selected">...</option>
  - <dl compact="compact">

9

Ajax and Dynamic xhtml

- **Dynamically inserted HTML should follow same rules as regular HTML**
  – Case, quotes, end tags, and boolean attributes
- **Wrong**
  – document.getElementById(resultRegion).innerHTML = '<H1 ALIGN="CENTER">Some
Text</H1>';
- **Right**
  – document.getElementById(resultRegion).innerHTML = '<h1align="center">Some
Text</h1>';

General Format of xhtml Documents

xhtml Template

- **Minimal format**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Some Title</title></head>
<body>
...
</body></html>
```

- **Common format**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title>Some Title</title></head>
<body>
...
</body></html>
```

12

Internet Explorer Bug

- **The xhtml specification recommends an XML declaration at the top**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```

- **IE runs in "quirks" mode if DOCTYPE is not first line of document**
  - So always omit XML declaration

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```

13

Legal DTDs

- **Transitional (non-stylesheet formatting like <font> and <i> allowed)**
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- **Strict (no non-stylesheet formatting)**
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- **Frame (for top-level page that uses frames)**
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
  - Note that the pages that are the frame cells do not use this DTD.
    Only the top-level page that sets up the overall layout.

14

The head and body elements

- **head**
  - Must contain title
  - May contain the following
    - meta, script, style, base
- **body**
  - Contains main part of the page
    - The part that you directly see in browser window
  - Attributes
    - bgcolor, background, text, link, vlink, alink
      - E.g.: <body bgcolor="blue">
      - Style sheets often used instead
    - onload, onunload, onfocus, onblur
      - For JavaScript event handling

15

HTML 5

Overview

- **DOCTYPE and minimal document format**
  - Emphasis on simplicity
  - Works in old browsers
    - Uses conventions that browsers have *already* been using
  - Does not strictly require the rules of XML
    - But XML rules (close tags, lowercase, quotes) normally followed
- **New input elements**
  - `<input type="number"/>`, `<input type="range"/>`, `<input type="date"/>` etc.
    - Reverts to normal textfield in old browsers
- **Canvas and new JavaScript APIs**
  - Gradually being introduced to new browsers

HTML 5 Document Format

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8"/>
<link href="css/some-stylesheet.css"
      rel="stylesheet"/>
<script src="scripts/some-script.js"></script>
</head>
<body>
...
</body>
</html>
```

Note the simple DOCTYPE, simplified meta tag, and omission of "type" in both the style sheet and script references. All of those work in old, pre-HTML5 browsers.

18

Hypertext Links and URLs

Hypertext Links

- **Usage**
  - Links can contain images and other inline elements (i.e., <a href…> ...)
- **Link to Absolute URL**
  - Use a complete URL beginning with http://
    ```
    Java is discussed in
    <a href="http://host/path/chapter2.html">
    Chapter 2</a>.
    ```
- **Link to Relative URL**
  - Use a filename or relative path to filename
    - If no slash, interpreted wrt location of current file
    ```
    Java is discussed in
    <a href="chapter2.html">Chapter 2</a>.
    ```

Hypertext Links (Continued)

- **Link to Section**
  - Use a section name (see below) preceded by #
    ```
    Images are discussed in
    <a href="#section2">Section 2</a>.
    ```
- **Link to Section in URL**
  - Use absolute or relative URL, then #, then section name
    ```
    Images are discussed in
    <a href="chapter1.html#section2">
    Section 2 of Chapter 1</a>.
    ```
- **Naming a Section**
  - Use <a name="..."> and do not include the pound sign
    ```
    <h2><a name="section2">Images</a></h2>
    ```

Interpreting Relative URLs

- **URLs starting with http:// (absolute URLs)**
  - Interpreted independently of the URL of current page
- **URLs starting with .**
  - Interpreted with respect to the directory of current file
    - Really with respect to the rightmost / in the URL, since URLs don't necessarily refer to real folders or files
- **URLs starting with ..**
  - Interpreted with respect to the parent directory of current file
    - Again, based on the URL the browser sees, not necessarily the real location of the resource
- **URLs starting with /**
  - Interpreted with respect to the hostname of the URL
- **Notes**
  - In addition to <a href...>, these URLs can apply to img, form, script, style, and many other elements

22

Interpreting Relative URLs (Examples)

- **Assume URL of current page is http://somehost.com/a/b/c/d/e/f.html**

Link	Result
...	http://otherhost.com/a/b.html
...	http://somehost.com/a/b/c/d/e/g.html
...	http://somehost.com/a/b/c/d/h.html
...	http://somehost.com/i.html

23

Block-Level Elements (Paragraph-Like Elements)

Block-Level Elements

- **Headings**
  - h1 ... h6
    - align attribute
- **Basic Text Sections**
  - p
  - pre
- **Tables**
  - Covered later in this lecture
- **Forms**
  - Covered in later lecture

- **Lists**
  - ol
    - li
  - ul
    - li
  - dl
    - dt
    - dd
- **Misc.**
  - hr
  - div
  - center

Shared Attributes

- **id**
  - A *unique* identifier, usually used so that JavaScript can reference the tag later.
    - <div id="placeholder-for-ajax"></div>
    - <input type="text" id="firstNameField" name="..."/>
- **class**
  - A stylesheet class name
  - <p class="warning">...</p>
- **style**
  - An inline style
  - <p style="color: red; font-size: 18px;">Warning!</p>
- **JavaScript event handlers**
  - onclick, onchange, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onkeypress, onkeydown, onkeyup
    - Discussed in later section on JavaScript

Headings

- **Heading Types**
  - <h1 ...> ... </h1>
  - <h2 ...> ... </h2>
  - <h3 ...> ... </h3>
  - <h4 ...> ... </h4>
  - <h5 ...> ... </h5>
  - <h6 ...> ... </h6>
- **Attributes: align**
  - Values: left (default), right, center
- **Nesting tags**
  - Headings and other block-level elements can contain inline (text-level) elements, but *not* vice versa

Headings: Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Headings</title>
</head>
<body bgcolor="#fdf5e6">
<p>
Samples of the six heading types:
</p>
<h1>Level-1 (h1)</h1>
<h2 align="center">Level-2 (h2)</h2>
<h3><i>Level-3 (h3) in italics</i></h3>
<h4 align="right">Level-4 (h4)</h4>
<h5>Level-5 (h5)</h5>
<h6>Level-6 (h6)</h6>
</body></html>
```

28

Headings: Result
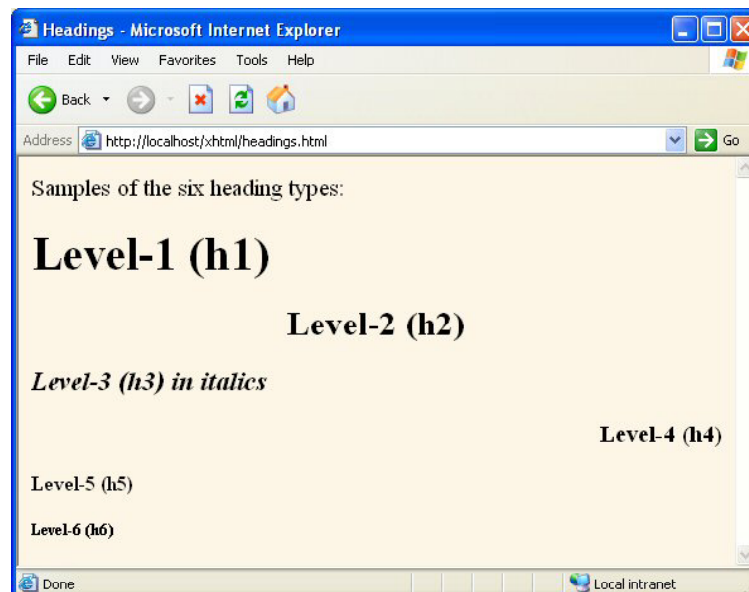
29

p – The Basic Paragraph

- **Attributes: align**
  - left (default), right, center. Same as headings.
  - Whitespace ignored (use <br/> for line break)
    - Consecutive <p/>'s do not yield multiple blank lines
      - But multiple
's do
  - xhtml: End tags required

xhtml:
```
<body>
<p>
Paragraph 1
</p>
<p>
Paragraph 2
</p>
<p>
Paragraph 3
</p>
</body>
```

HTML 4:
```
<BODY>
Paragraph 1
<P>
Paragraph 2
<P>
Paragraph 3
</BODY>
```

---

pre: Preformatted Paragraphs

- **Characteristics**
  - Line wrapping disabled
  - Whitespace no longer ignored
  - Fixed-width font used
- **Problem: Special Characters**

```
<pre>
if (a<b) {
   doThis();
} else {
   doThat();
}
</pre>
```

Desired Character	Text Required
<	<
>	>
&	&
"	"
Non-breaking space	

div

- **Purpose**
  - A container for enclosing other block-level elements
- **Attributes**
  - align, style, class, id
- **Usage**
  - Apply alignment or styles to multiple elements
  - Make a placeholder where Ajax results will be placed
    - Differs from span in that span can only enclose inline elements, whereas div can enclose both block and inline
- **The center tag**
  - <div align="center">...</div> is equivalent to <center>...</center>
    - div is allowed in both xhtml transitional and strict
    - center is allowed only in xhtml transitional

32

div and span: Example (Code)

```
<body bgcolor="#fdf5e6">
<p>Some text before the div</p>
<div style="border: inset; background-color: #cccccc;"
     align="center">
<h2>Using div and span</h2>
<p>A list inside the div:</p>
<ol>
  <li>List Item One</li>
  <li><span style="background-color: yellow;">
      List Item Two (inside span)</span></li>
  <li>List Item Three</li>
</ol>
</div>
<p>Some text after the div</p>
</body>
```
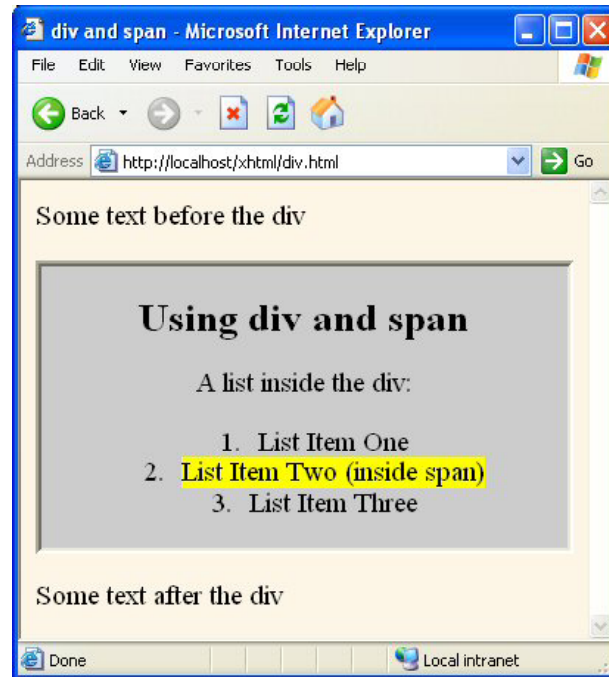
33

div and span: Example (Result)

ol: Ordered (Numbered) Lists

- **OL Element**
  - \
    \
…\
    \
…\
    ...
    \
  - Attributes: type, start, compact
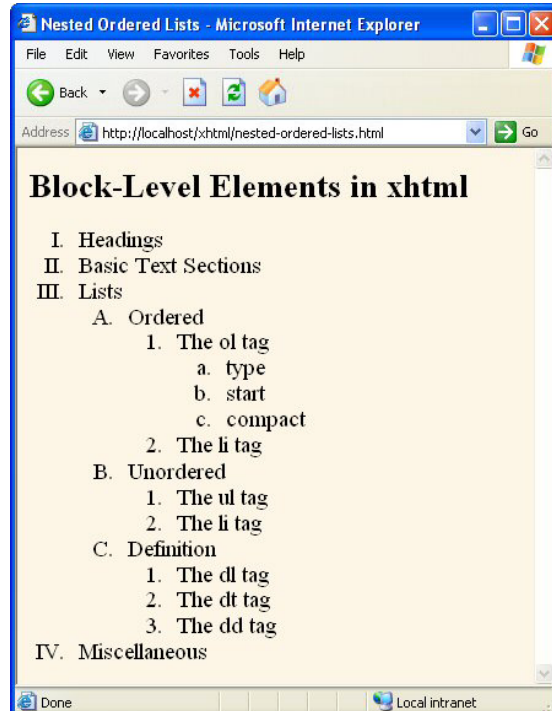- **List items: li**
  - Attributes: (When inside ol) value, type

```
<p>A sample list:</p>
<ol>
  <li>List Item One</li>
  <li>List Item Two</li>
  <li>List Item Three</li>
</ol>
```

A sample list:

1. List Item One
2. List Item Two
3. List Item Three

Nested Ordered Lists

```
<h2>Block-Level Elements in xhtml</h2>
<ol type="I">
  <li>Headings</li>
  <li>Basic Text Sections</li>
  <li>Lists
      <ol type="A">
        <li>Ordered
            <ol type="1">
              <li>The ol tag
                  <ol type="a">
                    <li>type</li>
                    <li>start</li>
                    <li>compact</li>
                  </ol></li>
              <li>The li tag</li>
            </ol></li>
        <li>Unordered
            <ol type="1">
              <li>The ul tag</li>
              <li>The li tag</li>
            </ol></li>
        <li>Definition
            <ol type="1">
              <li>The dl tag</li>
              <li>The dt tag</li>
              <li>The dd tag</li>
            </ol></li>
      </ol></li>
  <li>Miscellaneous</li>
</ol>
```

36

Block-Level Elements in xhtml

I. Headings
II. Basic Text Sections
III. Lists
  A. Ordered
    1. The ol tag
      a. type
      b. start
      c. compact
    2. The li tag
  B. Unordered
    1. The ul tag
    2. The li tag
  C. Definition
    1. The dl tag
    2. The dt tag
    3. The dd tag
IV. Miscellaneous

Done   Local intranet

UL: Unordered (Bulleted) Lists

- **ul Element**
  -
      …
      …
      ...

- **Attributes: type, compact**
  - type is disc, circle, or square
- **List items: li (type)**
  - type is disc, circle, or square

```
<p>A sample list:</p>
<ul>
  <li>List Item One</li>
  <li>List Item Two</li>
  <li>List Item Three</li>
</ul>
```

A sample list:

- List Item One
- List Item Two
- List Item Three

37

ul: Custom Bullets

```
<h2>Custom Bullets</h2>
<ul type="disc">
  <li>The ul tag
      <ul type="circle">
        <li>type
          <ul type="square">
            <li>disc</li>
            <li>circle</li>
            <li>square</li>
          </ul></li>
        <li>compact</li>
      </ul></li>
  <li>The li tag
      <ul type="circle">
        <li>type
          <ul type="square">
            <li>disc</li>
            <li>circle</li>
            <li>square</li>
          </ul></li>
        <li>value</li>
      </ul></li>
</ul>
```
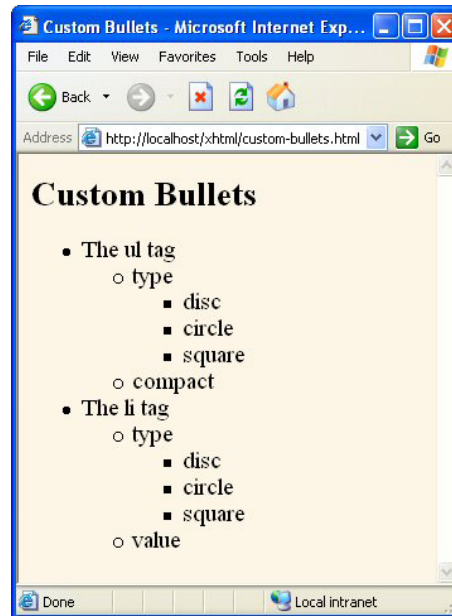

38

© 2011 Marty Hall

Inline (Text-Level) Elements (Font-Like Elements)

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2, PrimeFaces, Java 6 or 7, Ajax, jQuery, GWT, Spring, Hibernate, REST, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Inline Elements

- **Physical Character Styles**
  - b, i, tt, u, sub, sup, small, big, strike, s, blink
  - font
    - size
    - color
    - face
  - basefont
  - size
- **Logical Character Styles**
  - em, strong, code, samp, kbd, dfn, var, cite

Inline Elements (Continued)

- **Hypertext Links**
  - a
    - href, name, target, ...
    - See earlier section on URLs and links
- **Images**
  - img
    - src (required), alt, align, width, height, hspace, vspace, border, usemap, ismap
- **Misc. Inline Elements**
  - span (arbitrary style wrapped around text)
  - br (explicit line break)
  - area (client-side image map)
  - applet (Java)
  - ...
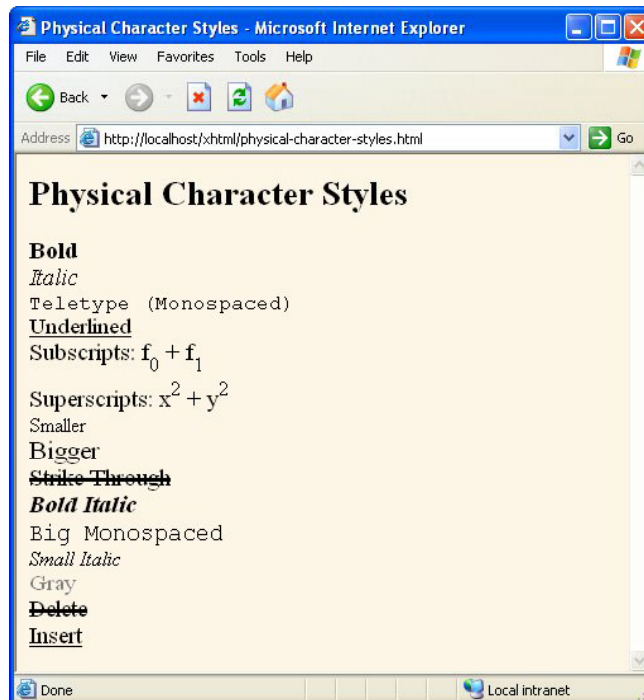
Physical Character Styles: Example

```
<body bgcolor="#fdf5e6">
<h2>Physical Character Styles</h2>
<b>Bold</b><br/>
<i>Italic</i><br/>
<tt>Teletype (Monospaced)</tt><br/>
<u>Underlined</u><br/>
Subscripts: f<sub>0</sub> + f<sub>1</sub><br/>
Superscripts: x<sup>2</sup> + y<sup>2</sup><br/>
<small>Smaller</small><br/>
<big>Bigger</big><br/>
<strike>Strike Through</strike><br/>
<b><i>Bold Italic</i></b><br/>
<big><tt>Big Monospaced</tt></big><br/>
<small><i>Small Italic</i></small><br/>
<font color="gray">Gray</font><br/>
<del>Delete</del><br/>
<ins>Insert</ins><br/>
</body>
```

Physical Character Styles: Result

Logical Character Styles: Example (Code)
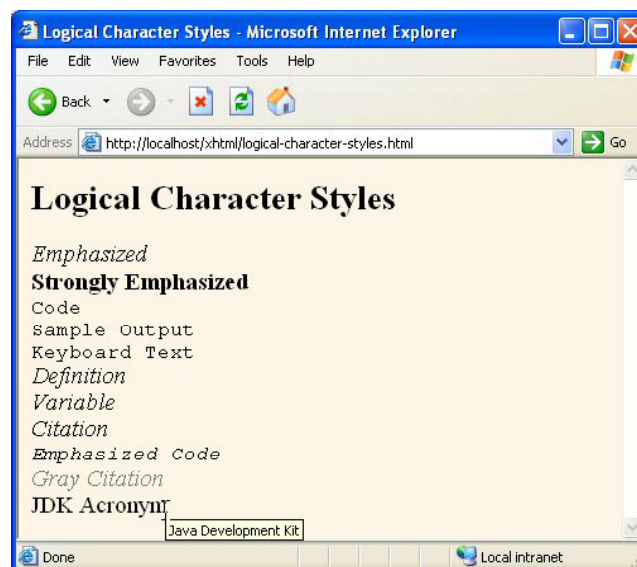
```
...
<h2>Logical Character Styles</h2>
<em>Emphasized</em><br/>
<strong>Strongly Emphasized</strong><br/>
<code>Code</code><br/>
<samp>Sample Output</samp><br/>
<kbd>Keyboard Text</kbd><br/>
<dfn>Definition</dfn><br/>
<var>Variable</var><br/>
<cite>Citation</cite><br/>
<em><code>Emphasized Code</code></em><br/>
<font color="gray"><cite>Gray Citation</cite></font><br/>
<acronym title="Java Development Kit">JDK Acronym</acronym>
...
```

44

Logical Character Styles: Example (Result)

45

img: Embedding Images

- **Example**

  ```
  <img src="Rover.gif" alt="My Dog"
       width="400" height="300"/>
  ```

- **Attributes:**
  - src (required)
  - alt (technically required)
  - align (see <br clear="all"/>)
  - width, height
  - hspace, vspace
  - border
  - usemap, ismap

Image Alignment: Example (Code)
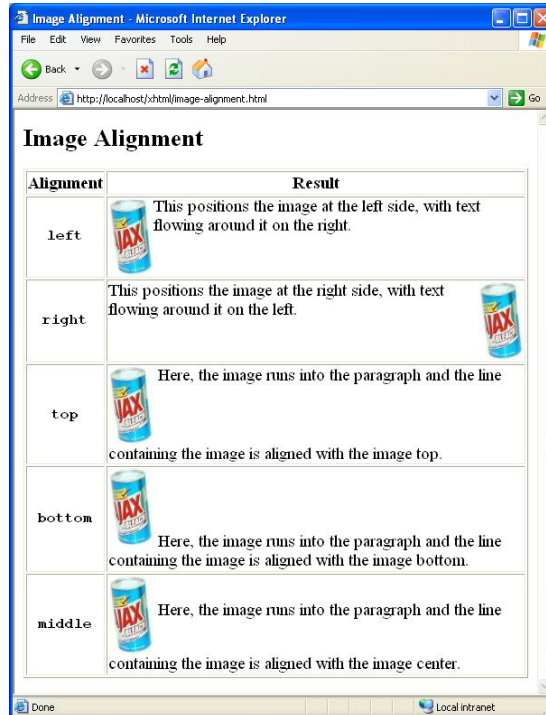
```
<table border="1">
  <tr><th>Alignment</th>
      <th>Result</th></tr>
  <tr><th><code>left</code></th>
      <td><img src="ajax.jpg" align="left"
              alt="Ajax" width="50" height="87"/>
          This positions the image at the left side,
          with text flowing around it on the right.</td></tr>
  <tr><th><code>right</code></th>
      <td><img src="ajax.jpg" align="right"
              alt="Ajax" width="50" height="87"/>
          This positions the image at the right side,
          with text flowing around it on the left.</td></tr>
  <tr><th><code>top</code></th>
      <td><img src="ajax.jpg" align="top"
              alt="Ajax" width="50" height="87"/>
          Here, the image runs into the paragraph
          and the line containing the image is
          aligned with the image top.</td></tr>
...
</table>
```

Image Alignment: Example (Result)

© 2011 Marty Hall

Tables

Basic Template

Borders are usually on for real tabular data, but tables are used even more widely for multi-column text, usually with borders off.

Many developers omit "caption" and use normal HTML markup to make captions.

```
<table border="1">
  <caption>Table Caption</caption>
  <tr><th>Heading1</th>      <th>Heading2</th></tr>
  <tr><td>Row1 Col1 Data</td><td>Row1 Col2 Data</td></tr>
  <tr><td>Row2 Col1 Data</td><td>Row2 Col2 Data</td></tr>
  <tr><td>Row3 Col1 Data</td><td>Row3 Col2 Data</td></tr>
</table>
```
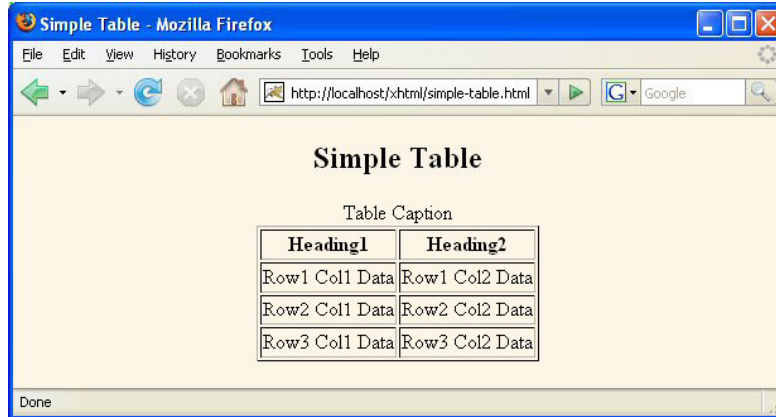
---

table Element Attributes

- **align**
  - The horizontal alignment of the table as a whole (left, right, center). Default is left.
  - Text flows around the table for left and right alignments
- **border**
  - The width in pixels of the border around the table (default: 0)
  - This is in addition to the border around each cell (the cellspacing).
- **bgcolor**
  - The background color of the table (e.g., "yellow" or "#rrggbb"). Also legal for tr, td, and th.
  - Tables are widely used to make colored sidebars or columns of text, but stylesheets are often a better way to specify the colors.
- **background**
  - The background image for the table. Will be tiled to fit table size. Also legal for td and th (but not tr).

table Element Attributes (Continued)

- **width, height**
  - This specifies the width or height of the table, either in pixels (<table width="250">) or, for width only, as a percentage of the current browser window width (<table width="75%">)
- **cellspacing**
  - The space in pixels between adjacent cells. Drawn as a 3D line if border is nonzero, otherwise empty space in the background color is used
  - The default is usually about 3
- **cellpadding**
  - The empty space, in pixels, between the cell's border and the table element
  - The default is usually about 1

table Element Attributes (Continued)

- **rules**
  - Specifies which inner dividing lines are drawn
  - All are drawn if this attribute is omitted
  - Legal values are none, rows, cols, and all
- **frame**
  - Specifies which outer borders are drawn
  - All four are drawn if this attribute is omitted
  - Legal values are border or box (all), void (none), above (top), below (bottom), hsides (top and bottom, despite the somewhat confusing name), vsides (left and right), lhs (left), and rhs (right)

tr: Table Row

- **tr is used to define each row in the table**
  - Each row will then contain th and/or td entries
- **align**
  - The default horizontal alignment for table cells. Legal values: left, right, or center
- **valign**
  - The default vertical alignment for table cells. Legal values: top, bottom, or middle
- **bgcolor**
  - The row color. Overrides bgcolor of the table as a whole.
  - Tables with rows that alternate colors are widely used, but again stylesheets are often used instead

Table Cells: th and td

- **Where legal**
  - th (table header) and td (table data cell) can be used anywhere in the table (inside tr). In particular, there is no restriction that th is only at the top.
- **Difference between th and td**
  - th uses bold font and center alignment by default. td uses normal font and left alignment by default.
  - Otherwise they are the same.
- **Real headers (and footers)**
  - For real headers, you can enclose one or more of the topmost rows inside thead. This does not change appearance in browsers, but when printed, the header will be repeated if the printout spans multiple pages. Similarly, bottom rows can go inside tfoot.

- **align**
  - left, right, center, justify and char.
  - E.g., the following aligns entries on a decimal point
    - `<td align="char" char=".">`
- **valign**
  - top, bottom, middle
- **width, height**
  - Values in pixels only (no percentages officially allowed)
- **bgcolor, background**
  - Background color and image (tiled)
- **nowrap**
  - Disables word wrapping. Use with caution

56

Table Cells: th and td – Attributes (Continued)

- **colspan**
  - Defines a wide cell that straddles more than one column
- **rowspan**
  - Defines a tall cell that straddles more than one row

- **Example**

```
<table border="1">
  <tr><th colspan="2">Col 1&amp;2 Heading</th>
      <th>Col3 Heading</th></tr>
  <tr><td>Row1 Col1 Data</td>
      <td rowspan="2">Row1&amp;2 Col2 Data</td>
      <td>Row1 Col3 Data</td></tr>
  <tr><td>Row2 Col1 Data</td>
      <td>Row2 Col3 Data</td></tr>
</table>
```

57

Forms

The form Tag

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>A Sample Form Using GET</title>
</head>
<body bgcolor="#fdf5e6">
<div align="center">
<h2>A Sample Form Using GET</h2>
<form action="http://localhost:8088/SomeProgram">
   First name:
   <input type="text" name="firstName" value="J. Random"/><br/>
   Last name:
   <input type="text" name="lastName" value="Hacker"/><p/>
   <input type="submit"/> <!-- Press button to submit form -->
</form>
</div>
</body></html>
```
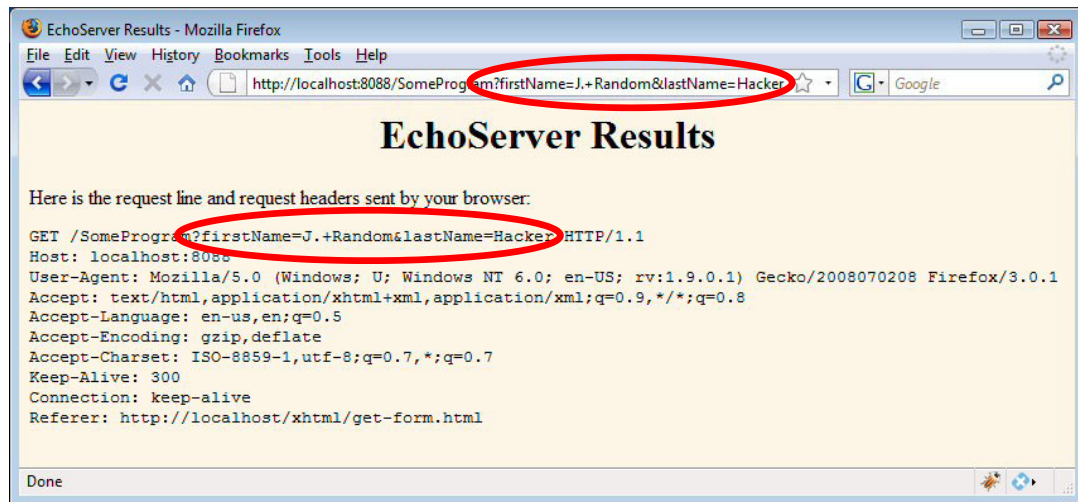
You normally use a relative URL for the ACTION. This URL is just for testing because I am running a test server on port 8088 that echoes the data it receives.

59

GET Form: Initial Result

GET Form: Submission Result (Data Sent to EchoServer)

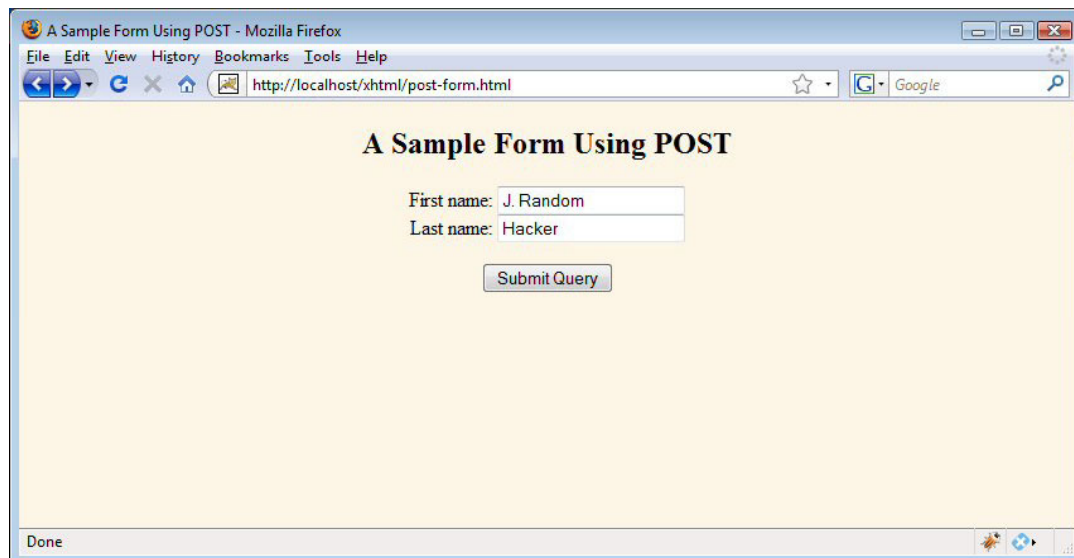# Sending POST Data

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>A Sample Form Using POST</title>
</head>
<body bgcolor="#fdf5e6">
<div align="center">
<h2>A Sample Form Using POST</h2>
<form action="http://localhost:8088/SomeProgram"
      method="post">
  First name:
  <input type="text" name="firstName" value="J. Random"/><br/>
  Last name:
  <input type="text" name="lastName" value="Hacker"/><p/>
  <input type="submit"/> <!-- Press button to submit form -->
</form>
</div>
</body></html>
```

The default method is GET. So, if a form says method="get" or it has no method at all, GET is used.

62

POST Form: Initial Result

A Sample Form Using POST - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/xhtml/post-form.html

Google

A Sample Form Using POST

First name: J. Random
Last name: Hacker

Submit Query

Done

63

POST Form: Submission Result (Data Sent to EchoServer)

EchoServer Results - Mozilla Firefox

File   Edit   View   History   Bookmarks   Tools   Help

http://localhost:8088/SomeProgram

G· Google

EchoServer Results

Here is the request line and request headers sent by your browser:

```
POST /SomeProgram HTTP/1.1
Host: localhost:8088
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://localhost/xhtml/post-form.html
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

firstName=J.+Random&lastName=Hacker
```

Done

Text Controls

- **Textfields**
  - <input type="text" name="…" …/>
    - `value` can give original value

    Textfield: Regular Text

- **Password Fields**
  - < input type="password" name="…" …/>
    - *Always* use `POST`

    Enter Password: ••••••

- **Text Areas**
  - <textarea name="…"
              rows="…" cols="…">
    …
    </textarea>

    Post your feedback here:
    What did you think of this article? Post your comments here then send it to our editors.

    Send to Editors

    - Interpretation of regular HTML tags turned off between `<textarea…>` and `</textarea>`

Push Buttons

- ## Submit Buttons
  - <input type="submit" …/>

    Submit Query
    - Use `name` if you have multiple buttons
    - Use `value` to change button's label
- ## JavaScript Buttons
  - <input type="button"
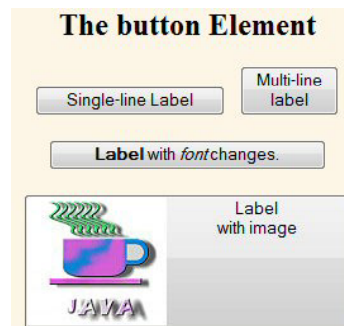   onclick="someJavaScriptFunction()" …/>
    - Widely used with Ajax
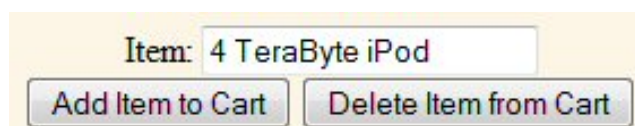- ## Reset Buttons
  - <input type="reset" …/>

    The button Element

    Single-line Label   Multi-line label

    Label with *font* changes.

    Label with image

- ## Fancy Buttons
  - <button type="submit" ...>
    html
    </button>

Multiple Submit Buttons

- ## Button names
  - Submit buttons don't normally need a name attribute, but if you have more than one button and want the server to identify which one was pressed, give them names
    - Used more with standard Web apps than with Ajax

```
Item:
<input type="text" name="Item" value="4 TeraByte iPod"/>
<br/>
<input type="submit" name="Add"
       value="Add Item to Cart"/>
<input type="submit" name="Delete"
       value="Delete Item from Cart"/>
```

Item: 4 TeraByte iPod

Add Item to Cart   Delete Item from Cart

Check Boxes

- **Format**
  - <input type="checkbox" name="…" …>
    - Use `checked="checked"` to make it initially checked
    - Name/value pair sent only if checkbox is checked when form is submitted
- **Example code**
  ```
  <p/>
  <input type="checkbox" name="noEmail" checked="checked"/>
  Check here if you do <i>not</i> want to
  get our email newsletter
  ```
- **Example result**

  ☑ Check here if you do *not* want to get our email newsletter

Radio Buttons

- **Format**
  - <input type="radio" name="…" value="…"…>
    - All radio buttons in a group should have same `NAME`
    - Only one button in a group can be pressed; pressing a different one causes previous one to pop out
- **Example**
  ```
  <dl>
    <dt>Credit Card:</dt>
    <dd><input type="radio" name="creditCard" value="visa"/>
        Visa</dd>
    <dd>…
    <dd><input type="radio" name="creditCard" value="java"
              checked="checked"/>
        Java Smart Card</dd>
    …
  </dl>
  ```

Credit Card:
- ◉ Visa
- ◉ Master Card
- ◉ Java Smart Card
- ◉ American Express
- ◉ Discover

Combo Boxes

- ## Format
  - select gives name
  - option gives value

- ## Example
  ```
  Favorite language:
  <select name="language">
     <option value="c">C</option>
     <option value="c++">C++</option>
     <option value="java" selected="selected">Java</option>
     <option value="javascript">JavaScript</option>
     <option value="perl">Perl</option>
     <option value="ruby">Ruby</option>
  </select>
  ```
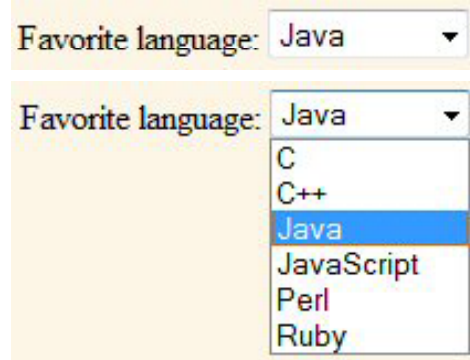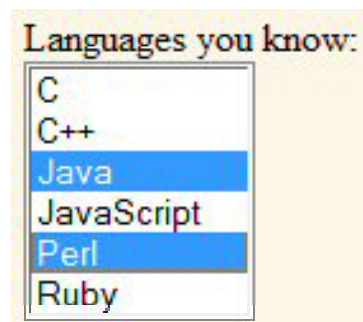
List Boxes

- ## Format
  - Similar to combo boxes, but specify multiple
  - Give size if you want to see all entries without scrollbars

- ## Example
  ```
  Languages you know:<br/>
  <select name="language" multiple="multiple" size="6">
     <option value="c">C</option>
     <option value="c++">C++</option>
     <option value="java" selected="selected">Java</option>
     <option value="javascript">JavaScript</option>
     <option value="perl" selected="selected">Perl</option>
     <option value="ruby">Ruby</option>
  </select>
  ```

Grouping Form Entries: label, fieldset, and legend

- **label**
  - If you use the label tag for prompts associated with fields, clicking on the label transfers focus to the input field
  - You can either use the "for" attribute or enclose the field within the label
    - <label for="fname">First name:</label>
      <input type="text" name="userFirstName" id="fname"/>
    - <label>First name:
        <input type="text" name="userFirstName"
      </label>

- **fieldset and legend**
  - Grouping all or part of a form inside fieldset draws attention to it and separates it from the rest of the page
  - Using style sheets for the legend is particularly useful

72

Grouping Form Entries: Example

- **HTML**

```
<fieldset>
  <legend>ajax:updateField</legend>
  <form ...>
    <label for="f">Enter temperature in Fahrenheit:</label>
    <input type="text" id="f"/>
    <input type="button" id="convertButton" value="Convert"/>
    <hr width="500" align="left"/>
    <label for="c">Temperature in Celsius:</label>
    <input type="text" id="c"/>
    <label for="k">Temperature in Kelvin:</label>
    <input type="text" id="k"/>
  </form>
</fieldset>
```

- **CSS**

```
legend {
  font-weight: bold;
  color: black;
  background-color: white;
  border: 1px solid #cccccc;
  padding: 4px 2px;
}
```

73

Other Controls and Options

- **File upload controls**
  - <input type="file" …/>
  - Lets user select a file and send it to the server

  Enter data file below:
  [   ] [Browse...]

- **Server-side image maps**
  - User clicks on an image and form gets submitted.
  - Form data gets sent as *name*.x=*x-pos*&*name*.y=*y-pos*
- **Hidden fields**
  - Preset `name` and `value` sent with form submission..
- **Tab order control**
  - `tabindex`

74

Thumbnail Review of CSS

Loading an External Style Sheet (Most Reusable Approach)

- **styles/my-styles.css**

  p { color: blue; }

  .note { font-weight: bold; background-color: red; }

- **my-page.html**

  <head>

      <link href="styles/my-styles.css"

          rel="stylesheet" type="text/css" />

  The type attribute is officially optional in HTML 5. In other HTML versions, although it is ignored by browsers, it should be included for HTML validators.

  …

  </head>

  <body>

  <p>Blue text</p>

  <p class="note">Blue bold text with red background</p>

  </body>

76

Embedding a style sheet

- **Embedding an internal style sheet**

  <head>

  Again, the type can be omitted in HTML5

  <style type="text/css">

    p { color: blue; }

    .note { font-weight: bold; background-color: red; }

  </style>

  …

  </head>

- **Listing CSS styles inline**

  – <h1 style="color: red; background-color: blue">…</h1>

77

CSS Selectors

```
#some-id  {
      Styles that apply to <h1 id="some-id">
}
p {
      Styles that apply to <p>
}
.note {
      Styles that apply to <div class="note">
}
p span {
      Styles that apply to <p>…<span>…</span>…</p>
}
h1, h2, td li {
      Styles that apply to <h1> and also to <h2> and also to <td>…<li>…</li>…</td>
}
```

78

Wrapup

References

- **Books**
  - *HTML, XHTML, and CSS, Sixth Edition*
    by Elizabeth Castro
  - *Head First HTML with CSS & XHTML*
    by Eric Freeman and Elisabeth Freeman
- **Online References**
  - xhtml specification: http://www.w3.org/TR/xhtml1/
  - xhtml cheat sheet (pocket reference):
    http://floele.flyspray.org/htmlcheatsheet.pdf
  - xhtml reference: http://xhtml.com/en/xhtml/reference/
  - xhtml tutorials: http://www.w3schools.com/xhtml/
  - Search on Google for "HTML tutorial[s]" and adapt
    syntax for xhtml (use validator to be sure)

Summary

- **Template**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Some Title</title></head>
<body>
...
</body></html>
```

- **Differences from HTML 4**
  - Use lowercase for tag names, attribute names, predefined
    attribute values (<h1 align="center">)
  - Always include end tags (<p></p> or <p/>)
  - Always use quotes around attribute values
    (<table border="1">)
  - Follow these rules for HTML inserted by Ajax

Questions?