Introduction to HTML programming

Web Programming

OUTLINE

- hypertext
- tags & elements
- text formatting
- lists, hyperlinks, images
- tables, frames
- cascading style sheets
  - inline, document, external

Hypertext & HTML

- HyperText Markup Language (HTML) is the language for specifying the *static* content of Web pages
  - *hypertext* refers to the fact that Web pages are more than just text
    - can contain multimedia, provide links for jumping within & without
  - *markup* refers to the fact that it works by augmenting text with special symbols (tags) that identify structure and content type

HTML evolution

- HTML is an evolving standard (as new technology/tools are added)
  - HTML 1 (Berners-Lee, 1989): very basic, limited integration of multimedia

    in 1993, Mosaic added many new features (e.g., integrated images)
  - HTML 2.0 (IETF, 1994): tried to standardize these & other features, but late

    in 1994-96, Netscape & IE added many new, divergent features
  - HTML 3.2 (W3C, 1996): attempted to unify into a single standard

    but didn't address newer technologies like Java applets & streaming video
  - HTML 4.0 (W3C, 1997): current standard

    attempted to map out future directions for HTML, not just react to vendors
  - XHTML 1.0 (W3C, 2000): HTML 4.01 modified to conform to XML standards

Web development tools

- many high-level tools exist for creating Web pages

  e.g., Microsoft FrontPage, Netscape Composer, Adobe PageMill,

  Macromedia DreamWeaver, HotDog, …

  also, many applications have "save to HTML" options (e.g., Word)

- *You can use any editor that does not support WYSIWYG.*
  - *PHP designer 2005 (Good for our course)*
  - *PsPad (almost for all the languages (C++, C#, HTML, PHP.)*

Tags vs. elements

- HTML specifies a set of *tags* that identify structure and content type
  - tags are enclosed in < >
    - `<img src="image.gif" />` specifies an image
  - most tags come in pairs, marking a beginning and ending
    - `<title>` and `</title>` enclose the title of a page
  - an HTML *element* is an object enclosed by a pair of tags

    `<title>My Home Page</title>` is a TITLE element

    `<b>This text appears bold.</b>` is a BOLD element

    `<p>Part of this text is <b>bold</b>.</p>`
    is a PARAGRAPH element that contains a BOLD element

HTML document is a collection of elements (text/media with context)

Structural elements

- An HTML document has two main structural elements
  - HEAD contains setup information for the browser & the Web page
    - e.g., the title for the browser window, style definitions, JavaScript code, …
  - BODY contains the actual content to be displayed in the Web page

```
<html>
<!-- Demo web page                    -->

<head>
  <title>Title for Page</title>
</head>

<body>
  Text that appears in the page
</body>

</html>
```

HTML documents begin and end with <html> and </html> tags

Comments appear between <!-- and -->

HEAD section enclosed between <head> and </head>

BODY section enclosed between <body> and </body>

Text layout

```
<html>
<!-- Demo web page            -->
<head>
  <title>Text Layout</title>
</head>

<body>
  <p>
  This is a paragraph of text<br/>
  made up of two lines.
  </p>

  <p>
  This is another paragraph with a
    GAP   between
  some of the words.
  </p>

  <p>
     This paragraph is<br/>
  indented on the first line<br/>
  but not on subsequent lines.
  </p>
</body>
</html>
```

view it

- For the most part, layout of the text must be left to the browser
  - every sequence of whitespace is interpreted as a single space
  - browser automatically wraps the text to fit the window size
- Can override some text layout
  - can cause a line break using the `<br/>` tag (no closing tag)
  - can specify a new paragraph (starts on a new line, preceded by a blank line) using `<p>...</p>`
  - can force a space character using the symbol for a non-breaking space: ` `

Separating blocks of text

```
<html>
<!-- Demo web page                -->
<head>
  <title>Blocks of Text</title>
</head>

<body>
  <h1>Major heading 1</h1>
  <p>
  Here is some text.
  </p>

  <h2>Subheading</h2>
  <p>
  Here is some subtext.
  </p>

  <hr/>

  <h1>Major heading 2</h1>
  <p>
  Here is some more text.
  </p>
</body>

</html>          view it
```

- Specify headings for paragraphs or blocks of text
  - `<h1>...</h1>` tags produce a large, bold heading
  - `<h2>...</h2>` tags produce a slightly smaller heading.
- Insert a horizontal rule to divide sections
  - `<hr/>` draws line across window
  - `<hr width="50%" />` sets width
  - `<hr size=10 />` sets thickness

Aligning text

```
<html>
<!-- Demo web page                    -->
<head>
  <title>Text Alignment</title>
</head>

<body>
  <h1 style="text-align:center">
Centered Heading</h1>
  <p>
  Here is some left-justified text
  (which is the default in HTML).
  </p>

  <p style="text-align:center">
  Here is some centered text.
  </p>

  <div style="text-align:right">
  <h2>Right-justified Heading</h2>
  <p>Here is some right-justified
text.</p>
  </div>
</body>

</html>          view it
```

- Specify how elements should be aligned (default is left-justified)
  - utilize STYLE attribute of tag
- Justify more than one element as a group, use DIV tags
  - all elements enclosed in DIV are formatted similarly

Text styles

```
<html>
<!-- Demo web page         -->

<head>
  <title>Text Styles</title>
</head>

<body>
  <p>
  Text can be emphasized using
  <b>bold</b>, <i>italics</i>, or
  even <big>resizing</big>.  <br/>

  The typewriter font is good for
  displaying code:

  <tt>sum = sum + i;</tt>  <br />
  And remember:
 <span style="color:red">
  <small>2<sup>10</sup></small> =
  1024</span>
  </p>

</body>
</html>      view it
```

- Specify styles for fonts
  - … specify bold
  - … specify italics
  - <tt>… </tt> specify typewriter-like (fixed-width) font
  - <big>… </big> increase the size of the font
  - <small>… </small> decrease the size of the font
  - _… specify a subscript
  - […] a superscript
  - … for paragraphs
  - … for inline text

More text grouping

```
<html>
 <head>
  <title>More Text Grouping</title>
</head>

<body>
  <p>
  <b><tt><pre>
    for (i = 0 , i < 10 , i++) {
        sum = sum + i;
    }
  </pre></tt></b>
  </p>

  <p>
  Remember:
  <blockquote>
 the above <tt> for </tt> loop is
not correct.
  </blockquote>
  </p>
</body>
</html>
```

- `<pre>…</pre>` specify text that is to be displayed as is (line breaks and spacing are preserved)
  - *useful for code or whenever you want text to fit a specific layout*

- `<blockquote>..</blockquote>` specify text that is to be indented on both margins
  - *useful for quotations or for indenting text in subsections*

Lists

```
<html>
<head>
  <title>Simple Lists</title>
</head>
<body>
  <p> <ol>
    <li>First thing.
    <li>Second thing.
    <li>Third thing.
  </ol>   </p>

<p> <ul>
    <li> One.
    <li> One.
    </ul> </p>

  <p> <dl>
    <dt>HTML
    <dd>HyperText Markup Language
    <dt>HTTP
    <dd>HyperText Transfer Protocol
  </dl>
  </p>
</body>
</html>      view it
```

- there are 3 different types of list elements
  - `<ol>...</ol>` specifies an ordered list (using numbers or letters to label each list item)
    - `<li>` identifies each list item
  - `<ul>...</ul>` specifies unordered list (using a bullet for each)
    - `<li>` identifies each list item
  - `<dl>...</dl>` specifies a definition list
    - `<dt>` identifies each term
    - `<dd>` identifies its definition

Hyperlinks

```
<html>
<!-- demo -->

<head>
  <title>Hyperlinks</title>
</head>

<body>
  <p>
<a href="http://www.gust.edu.kw">
  GUST</a>
  <br>

<a href="p7.html" target="_blank">
  Open page07 in a new window</a>
  </p>

</body>
</html>   view it
```

- perhaps the most important HTML element is the hyperlink, or ANCHOR
  - `<a href="`*URL*`">...</a>`
    - where URL is the Web address of the page to be displayed when the user clicks on the link
    - *If the page is accessed over the Web, must start with* `http://`
    - *if not there, the browser will assume it is the name of a local file*
  - `<a href="`*URL*`" target="_blank">...</a>`
    - causes the page to be loaded in a new window

Hyperlinks (cont.)

```
<html>
 <head>
  <title>Internal Links in a
Page</title>
</head>
<body>
  <p align="center">
  [ <a href="#HTML">HTML</a> |
    <a href="#HTTP">HTTP</a> |
    <a href="#IP">IP</a> |
    <a href="#TCP">TCP</a> ]   </p>
  <p>
  Computer acronyms:
  <dl>
    <a name="HTML"></a><dt>HTML
    <dd>HyperText Markup Language
    <a name="HTTP"></a><dt>HTTP
    <dd>HyperText Transfer Protocol
    <a name="IP"></a><dt>IP
    <dd>Internet Protocol
    <a name="TCP"></a><dt>TCP
    <dd>Transfer Control Protocol
  </p>
</body>
</html>
```

- for long documents, you can even have links to other locations in that document
  - `<a name="ident">...</a>`
    - where ident is a variable for identifying this location
  - `<a href="#ident">...</a>`
    - will then jump to that location within the file
  - `<a href="URL#ident">...</a>`
    - can jump into the middle of another file just as easily

Images

- Include images using `IMG`
  - by default, browsers can display GIF and JPEG files
  - other image formats may require plug-in applications for display
    - `<img src="filename" alt="alternate text" />`
    - again, if file is to be accessed over the Web, must start with http:// (if not, will assume local file)

```
<html>
 <head>
  <title>Images</title>
</head>

<body>
  <div style="text-align:center">
    <img src="http://eee.gust.edu.kw/elearning/logo.gif"
         alt="logo.gif" />
  <p>GUST logo</p>
  </div>
</body>
</html>
```

Tables

- tables are common tools for arranging complex layout on a Web page
  - a table divides contents into rows and columns
  - by default, column entries are left-justified, so provide for alignment

```
<html>
<head>
  <title>Tables</title>
</head>

<body>
  <table>
    <tr>
      <td>HTML</td> <td>XHTML</td>
    </tr>
    <tr>
      <td>XML</td> <td>PHP</td>
    </tr>
  </table>
</body>
</html>  view it
```

`<table>…</table>` specify a table element

`<tr>…</tr>` specify a row in the table

`<td>…</td>` specify table data (i.e., each column entry in the table)

Layout in a table

```
<html>


<head>
  <title>Table Layout</title>
</head>

<body>
  <table border=1>
    <tr align="center">
      <td>HTML<br>XHTML</td>
      <td valign="top">W3C</td>
    </tr>
    <tr>
      <td>XML</td>
      <td>PHP</td>
    </tr>
  </table>
</body>
</html>
```

- Get a border on tables using the BORDER attribute
  - `<table border=1>`
  - *increasing the number makes the border thicker*
- Control the horizontal & vertical layout within cells
  - `<td align="center">`
  - `<td align="right">`
  - `<td valign="top">`
  - `<td valign="bottom">`
- Apply layout to an entire row
  - `<tr align="center">`
  - `<tr valign="top">`

Table width

```
<html>


<head>
  <title>Table Width</title>
</head>

<body>
  <table width="100%">
    <tr>
      <td>left-most
      <td align="right">right-
most</td>
    </tr>
</table>
</body>
</html>
```

- by default, the table is sized to fit the data
- can override & specify the width of a table relative to the page
  - `<table width="60%">`

Other table options

```
<html>

<head>
  <title>Table Formatting</title>
</head>

<body>
  <table border=1 cellspacing=4 cellpadding=8>
    <tr>
      <th>HEAD1</th> <th>HEAD2</th> <th>HEAD3</th>
    </tr>
    <tr>
      <td>one</td> <td>two</td> <td>three</td>
    </tr>
    <tr>
      <td rowspan=2 align="center"> four </td>
      <td colspan=2 align="center"> five </td>
    </tr>
    <tr>
      <td> six </td> <td> seven </td>
    </tr>
</table>
</body>
</html>
```

- can control the space between cells & margins within cells
  - `<table ellspacing=5>`
  - `<table cellpadding=5>`
- Add headings
  - `<th>` is similar to `<td>` but displays heading centered in bold
- Have data that spans more than one column
  - `<td colspan=2>`
- Similarly, can span more than one row
  - `<td rowspan=2>`

Frames

- frames provide the ability to split the screen into independent pages
  - must define a FRAMESET that specifies the layout of the pages
  - actual pages to be displayed must be in separate files

```
<html>


<frameset cols="*,*">
  <frame src="page01.html">
  <frame src="page02.html">
</frameset>

</html>
```

can divide vertically

```
<frameset cols="50%,50%">
```

or, horizontally

```
<frameset rows="30%,*,*">
```

** causes the browser to divide the remaining space evenly*

by default, each frame scrollable

- can drag the border to resize
- can hide the border with frameset attribute `frameborder=0`

Frame controversy

- frames are probably the most controversial HTML feature
  - some people love them, some people hate them
- 2 reasonable uses for frames
  - as a navigational aid:
    - divide the screen into a static menu frame and the main frame for navigating a site.
  - as a means of separating program input from output:
    - divide the screen into a static man input form frame and the main frame for displaying output

Menu Frame

- To create a menu, need to be able to direct links to the main frame
  - name the frames in the FRAMESET
  - specify the frame name as TARGET in the link
  - specify `_top` as target to return to top level of browser

Menu frame

```
<html>

<head>
  <title>Demo Browser</title>
</head>

<frameset cols="30%,*">
    <frame src="menu16.html" name="menu">
    <frame src="page01.html" name="main">
</frameset>

</html>
```

```
<html>

<head>
  <title>Menu of Demos</title>
</head>

<body>
Links to demo pages

<p>
<a href="page01.html"
   target="main">Demo 1</a><br/>
<a href="page02.html"
   target ="main">Demo 2</a><br/>
<a href="page03.html"
   target ="main"> Demo 3</a><br/>
<a href="page04.html"
   target ="main"> Demo 4</a><br/>
<a href="page05.html"
   target ="main"> Demo 5</a><br/>
<a href="page06.html"
   target ="main"> Demo 6</a><br/>
<a href="http://www.gust.edu.kw"
   target="_top" >GUST</a>
</p>
</body>
</html>
```

Content vs. presentation

- most HTML tags define content type, independent of presentation
  - exceptions?
- style sheets associate presentation formats with HTML elements
  - CSS1: developed in 1996 by W3C
  - CSS2: released in 1998, but not fully supported by browsers
  - HTML style sheets are known as *Cascading Style Sheets*, since can be defined at three different levels
    1. *inline* style sheets apply to the content of a single HTML element
    2. *document* style sheets apply to the whole BODY of a document
    3. *external* style sheets can be linked and applied to numerous documents

    lower-level style sheets can override higher-level style sheets

Forms

- An HTML form is a collection of elements for handling input, output, and events in a page
- Most event-handling in "JavaScript" is associated with form elements

```
<form action=" " name="FormName">
…
</form>
```

- form elements include:

  for input: button, selection list, radio button, check box, password, …

  for input/output: text box, text area

Button Elements

- The simplest form element is a **button**
  - analogous to a real-world button, can click to trigger events

- Buttons Type:
  - Submit
  - Reset
  - JavaScript buttons

Buttons

- Submit Buttons
  - <INPUT TYPE="SUBMIT" …>
    - Use NAME if you have multiple buttons
    - Use VALUE to change button's label
- Reset Buttons
  - <INPUT TYPE="RESET" …>
    - Use VALUE to change button's label

```
<html>
  <head>
  <title> Buttons</title>
  </head>
 <body>
  <form action = “ “name="ButtonForm">
    <input type=“submit" name=“my_button” value=“OK "/>
    <input type=“reset" name=“my_button” value=“reset "/>
  </form>
 </body>
</html>
```

Button Elements

- JavaScript Buttons

```
<input type="button" value="LABEL"
 onClick="JAVASCRIPT_CODE" />
```

```
<html>
  <head>
  <title> Buttons</title>
  </head>

 <body>
  <form name="ButtonForm">
    <input type="button" value=“OK " onClick=“ ”  />
  </form>
 </body>
</html>
```

Checkbox Elements

- Format
  - <INPUT TYPE="CHECKBOX" NAME="…" …>
    - The CHECKED attribute makes it initially checked
    - Name/value pair sent only if checkbox is checked when form is submitted

```
<html>
  <head>
  <title> Buttons</title>
  </head>

 <body>
  <form name="ButtonForm">
 <P>
     <INPUT TYPE="CHECKBOX" NAME="noEmail" CHECKED>
     Check here if you do <I>not</I> want to subscribe
     </form>
 </body>
</html>
```

view it

Combobox Elements

- <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

```
<html>
  <head>
  <title> Buttons</title>
  </head>

 <body>
 <form name="ButtonForm">

    Favorite language:
    <SELECT NAME="language">
      <OPTION VALUE="c">C
      <OPTION VALUE="c++">C++
      <OPTION VALUE="java"
    SELECTED>Java
      <OPTION VALUE="lisp">Lisp
      <OPTION VALUE="perl">Perl
      <OPTION VALUE =“CGI">CGI


    </SELECT>
</form>
 </body>
</html>                    view it
```

Listbox

- Format
  - Identical to combo boxes, but specify `MULTIPLE`
- Example

```
Languages you know:<BR>
<SELECT NAME="language" MULTIPLE>
  <OPTION VALUE="c">C
  <OPTION VALUE="c++">C++
  <OPTION VALUE="java" SELECTED>Java
  <OPTION VALUE="lisp">Lisp
  <OPTION VALUE="perl" SELECTED>Perl
  <OPTION VALUE="smalltalk">Smalltalk
</SELECT>
```

viewit

Text Box

- a text box allows for user input

```
<input type="text" name="BOX_NAME"…>
```

- optional attributes:
  - `SIZE` : width of the box (number of characters)
  - `VALUE` : initial contents of the box

- As we'll see in chapter 4: JavaScript code can access the contents as `document.FormName.BoxName.value`

Text Box: example

```
<html>

 <head>  <title> Fun with Text Boxes </title>  </head>

 <body>
  <form name="BoxForm">
    Enter your name here:
    <input type="text"  name="userName" size=12 value="" />
    <br /><br />

       <input type="button" value="OK"
       onClick="alert('Thanks, ' + document.BoxForm.userName.value);"
     />
  </form>
 </body>
</html>
```

view it

Text Area

- a TEXT box is limited to one line of input/output
- a TEXTAREA is similar to a text box in functionality, but can specify any number of rows and columns

```
<textarea name="TextAreaName" rows=NumRows
cols=NumCols wrap="virtual">
 Initial Text
 </textarea>
```

  - a TEXTAREA has closing tag
  - initial contents of the TEXTAREA appear between the tags
  - WRAP="virtual" specifies that text in the box will wrap lines as needed
  - as with a text box, no HTML formatting of TEXTAREA contents