Publishing On the Web

Publishing information on the WWW is an activity that involves three major steps:

1. **Create a document**
   - HTML with any text editor
   - HTML editors

2. **Put the document on the Web**

3. **Validate the document**
   - Try the page with different browsers and Oss
   - HTML validators

Course Content

- Introduction
- Internet and WWW
- Protocols
- **HTML and beyond**
- Databases & WWW
- SGML / XML
- Managing servers
- Search Engines
- Web Mining
- CORBA
- Security Issues
- Selected Topics
- Projects

Objectives of Lecture 4

**HTML and beyond**

- Get an overview of the hypertext markup language used for publishing on the WWW.
- See some advanced features of the last version of the languages such as cascading style sheets
- After the Lecture, students will demonstrate a sound understanding of HTML standards.
Web Technologies and Applications

University of Alberta

© Dr. Osmar R. Zaïane, 2001

Outline of Lecture 4

• What is HTML?
• Basic HTML
• Building Lists
• Tables
• Frames
• Image Maps
• Validating HTML pages
• Cascading Style Sheets

Web Publishing with HTML

• HTML stands for HyperText Markup Language.
• Language format based on SGML.
• Non-proprietary language.
• Based on a set of tags that indicate how document-content should be rendered.
• There are many versions of HTML and different browsers have their own add-ons.

Web Publishing with HTML

• HTML stands for HyperText Markup Language.
• Language format based on SGML.
• Non-proprietary language.
• Based on a set of tags that indicate how document-content should be rendered.
• There are many versions of HTML and different browsers have their own add-ons.

Web Publishing with HTML

• HTML stands for HyperText Markup Language.
• Language format based on SGML.
• Non-proprietary language.
• Based on a set of tags that indicate how document-content should be rendered.
• There are many versions of HTML and different browsers have their own add-ons.

History of HTML

• HTML was originally developed by Tim Berners-Lee in CERN in 1990. HTML+ (1993) was later abandoned.
• HTML 2.0, now obsolete, was ratified by the IETF in 1994.
• HTML 3.0 was abandoned due to lack of compromise between major browsers
• HTML 3.2 (less than HTML 3.0) is the most used standard today (since 1997)
• HTML 4.0 (end of 1997) brought new enhancements.
• XHTML 1.0 is the latest recommendation of the W3C. It is a combination of HTML4 and XML.

Major Syntax Components

• <TAG parameters=values> text </TAG>
• <TAG parameters=values>
• You may have many parameters or no parameters at all.
• HTML is not case sensitive
• Values can be case sensitive
• Tags for changing appearance of text, tags for structuring sections, tags for embedding objects, etc.
Do We Have to Know HTML?

- There are many HTML editors and publishing software that generate HTML.
- We can create web pages without knowing HTML.
- However, HTML editors do not generate correct HTML and the generated HTML is not easy to update.
- Many HTML editors do not deal with style sheets and JavaScript that can add neat functionalities.
- In order to write web-based application one need to know HTML since server-side scripts and programs generate HTML.

Outline of Lecture 4

- What is HTML?
  - Basic HTML
  - Building Lists
  - Tables
  - Frames
  - Image Maps
  - Validating HTML pages
  - Cascading Style Sheets

Page Structure

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>title text here</title>
  </head>
  <body bgcolor="#00FF00">
    Text and tags come here
  </body>
</html>
```

What Comes in the Head?

- `<TITLE> ...</TITLE>` Title of page
- `<BASE HREF="...">` Starting location for relative URLs
- `<SCRIPT ...>` Embedded script programs
- `<STYLE ...>` Specify cascading style sheets
- `<LINK ...>` Creates associations between pages
- `<META ...>` Records document information
  - `<META Name="author" Content="O. Zaiane">`
  - `<META HTTP-EQUIV="Expires" Content="Sun, 31 Dec 2000 23:59">`
- And many others
What Comes in the Body?

Optional parameters for BODY tag:

- **BACKGROUND** Specifies URL of an image to display in the background
- **BGCOLOR** Specifies the color of the background
- **TEXT** Specifies the color of the text
- **LINK** Specifies the color of the unvisited hyperlinks
- **VLINK** Specifies the color of the visited hyperlinks
- **ALINK** Specifies the color of the links currently selected

Some HTML Tags

Headers:
- `<H1>` Header 1
- `<H2>` Header 2
- `<H3>` Header 3
- `<H4>` Header 4
- `<H5>` Header 5
- `<H6>` Header 6

Optional parameters for `<H1>` to `<H6>`:

- `ALIGN=left|right|center`

Horizontal line:
- `<HR>`
  - `SIZE=4`
  - `WIDTH=50%`

Some useful tags:

Line breaks and Paragraphs:
- `<BR>` Line 1 and Line 2
- `<P>` Paragraph
- `<P ALIGN=left|right|center>`...

Bold Text and Underlined Text:
- `<B>` Bold Text
- `<I>` Italic Text
- `<U>` Underlined Text
- `<STRIKE>` Strikethrough Text
- `<TT>` Teletype Text

Descriptive markup:
- `<ADDRESS>` Contact info
- `<BLOCKQUOTE>` some text
- `<CODE>` some code
- `<VAR>` myVar

Rendering depends upon browser:
- `<PRE>` Text is unchanged
- `<XMP>` Interpretation off

Special characters in HTML

More information can be found at:
Some HTML Tags – con’t

Some useful tags:

Hyperlinks:

- `<A HREF=http://www.ualberta.ca TARGET=_TOP>` This a link to UoF/A
- `<A NAME=section1>text section 1</A>

Font sizes and Colours:

- `<FONT SIZE=4>` this is size 4, while this is size 5 `<FONT>`
- `<FONT COLOR=red>` This is red `<FONT>`
- `<FONT COLOR=white>` and this is white `<FONT>`
- `<FONT COLOR=cyan SIZE=4 FACE=arial>` all parameters `<FONT>`

More on sizes:

- `<BIG>` this is bigger `<BIG>`
- `<SMALL>` this smaller `<SMALL>`

- `<BASEFONT SIZE=3>`

The 11 `<SUP>` th `<SUP>` is in superscript and x `<SUB>` 2 `<SUB>` is subscript

Embedding images:

- `<IMG SRC=MyFile.gif WIDTH=100 HEIGHT=200>`
- `<IMG SRC=MyFile.gif ALIGN=right>` text comes here
- `<IMG SRC=MyFile.gif ALIGN=left>` text comes here
- `<IMG SRC=MyFile.gif ALIGN=top>` text comes here
- `<IMG SRC=MyFile.gif ALIGN=bottom>` text comes here
- `<IMG SRC=MyFile.gif ALIGN=middle>` text comes here

- `<IMG SRC=MyFile.gif ALT="alternative text">`

Other parameters:

- `<HSPACE, VSPACE, BORDER, USEMAP, ISMAP, LOWSRC, NAME>

Embedding objects:

- Embedding video, audio, VRML, and other formats via Plug-ins `<EMBED SRC=“…” …> …</EMBED>`

  - with generic attributes such as WIDTH and HEIGHT as well as plug-in-specific attributes such as AUTOSTART, etc.

- Embedding applets `<APPLET CODE=“…” WIDTH=xxx HEIGHT=yyy …>`

  - `<PARAM NAME=“…” VALUE=“…”>`

  - `…` `<APPLET>`

Embedding ActiveX components `<OBJECT CLASSID=“…” …> … </OBJECT>`

(compromises security, good only for intranets)
Outline of Lecture 4

• What is HTML?
• Basic HTML
• Building Lists
• Tables
• Frames
• Image Maps
• Validating HTML pages
• Cascading Style Sheets

Some HTML – con’t

Some useful tags:

<table>
<thead>
<tr>
<th>Ordered lists:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;OL&gt;</td>
</tr>
<tr>
<td>&lt;LI&gt;element 1</td>
</tr>
<tr>
<td>&lt;LI&gt;element 2</td>
</tr>
<tr>
<td>&lt;LI&gt;element 3</td>
</tr>
<tr>
<td>&lt;/OL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unordered lists:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;UL&gt;</td>
</tr>
<tr>
<td>&lt;LI&gt;element 1</td>
</tr>
<tr>
<td>&lt;LI&gt;element 2</td>
</tr>
<tr>
<td>&lt;LI&gt;element 3</td>
</tr>
<tr>
<td>&lt;/UL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition lists:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DL COMPACT&gt;</td>
</tr>
<tr>
<td>&lt;DT&gt;definition term1</td>
</tr>
<tr>
<td>&lt;DD&gt;element 1 description comes here</td>
</tr>
<tr>
<td>&lt;/DD&gt;</td>
</tr>
<tr>
<td>&lt;DT&gt;definition term2</td>
</tr>
<tr>
<td>&lt;DD&gt;element 2 description comes here</td>
</tr>
<tr>
<td>&lt;/DD&gt;</td>
</tr>
<tr>
<td>&lt;/DL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition lists with image bullets:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DL&gt;</td>
</tr>
<tr>
<td>&lt;DT&gt;&lt;IMG SRC=&quot;soccer.gif&quot; ALT=&quot;*&quot;&gt;</td>
</tr>
<tr>
<td>element 1</td>
</tr>
<tr>
<td>&lt;DT&gt;&lt;IMG SRC=&quot;soccer.gif&quot; ALT=&quot;*&quot;&gt;</td>
</tr>
<tr>
<td>element 2</td>
</tr>
<tr>
<td>&lt;/DL&gt;</td>
</tr>
</tbody>
</table>
Outline of Lecture 4

• What is HTML?
• Basic HTML
• Building Lists
  • Tables
  • Frames
  • Image Maps
  • Validating HTML pages
  • Cascading Style Sheets

Basic Structure of Tables

Tables:
<TABLE BORDER=1 WIDTH=100%>
  <TR>
    <TH>Title1</TH><TH>Title 2</TH>
  </TR>
  <TR>
    <TD>data 1</TD><TD>data 2</TD>
  </TR>
  <TR>
    <TD>data 3</TD><TD>data 4</TD>
  </TR>
</TABLE>

Tables in Tables

<TABLE BORDER=1 WIDTH=100%>
  <TR>
    <TH>Title1</TH><TH>Title 2</TH>
  </TR>
  <TR>
    <TD>data 1</TD><TD>data 2</TD>
  </TR>
  <TR>
    <TD>data 3</TD>
    <TD><TABLE><TR><TD>1</TD><TD>2</TD></TR></TABLE></TD>
  </TR>
</TABLE>

Table Parameters

• BORDER
• ALIGN
• WIDTH
• CELLSPACING
• CELLPADDING
• BCOLOR
• <CAPTION ALIGN=...>…</CAPTION>
• BORDERCOLOR (non standard)
• FRAME (non standard)
Rows and Cells Parameters

- **TR**
- **ALIGN**
- **VALIGN**
- **BGCOLOR**
- **TH** and **TD**
- **COLSPAN**
- **ROWSPAN**
- **ALIGN**
- **VALIGN**
- **BGCOLOR**
- **WIDTH**
- **HEIGHT**
- **NOWRAP**

Tables and Page Layout

Borderless tables can be useful for web page layout.

```html
<TABLE BORDER=0 WIDTH=100%>
  <TR>
    <TD rowspan=2><IMG SRC=MyFile.gif></TD>
    <TD>Line1</TD>
  </TR>
  <TR>
    <TD>Line2</TD>
  </TR>
  <TR>
    <TD>data1</TD>
    <TD>data2</TD>
  </TR>
</TABLE>
```

The Frames

- Frames present a different template for HTML documents. They were introduced as a standard in HTML 3.2.
- Rather than **HEAD** and **BODY**, the document has **HEAD** and **FRAMESET**.
- The **FRAMESET** divides the document into a set of frames, each referring to a URL.
- Frames contain sub documents, however, while a frame is an HTML document per se, the set of frames is also a document. (Web document view)
The Frames

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>title text here</title>
</head>
<frameset rows=50,*>
  <frame src="MyPage.html">
  <frameset cols="*,*">
    <frame src="P1.html">
    <frame src="P2.html">
  </frameset>
</frameset><noframe>
  <body>You need another browser</body>
</noframe>
</html>
```

The Different Framesets

- **Having rows:**
  ```html
  <frameset rows=row1Size, row2Size, row3Size,…>
  </frameset>
  <frameset rows=10%,*>
  </frameset>
  <frameset rows=*,*>
    = <frameset rows=50%,50%>
  </frameset>
  ```

- **Having columns:**
  ```html
  <frameset cols=col1Size, col2Size,col3Size,…>
  </frameset>
  <frameset cols=*,*> Other parameters:
FRAMEBORDER, FRAMESPACING, BORDERCOLOR…
```

The Element FRAME

- A FRAME can contain another FRAMESET.
- A FRAME has a NAME and SRC.
- Other parameters: FRAMEBORDER, BORDERCOLOR, NORESIZE, SCROLLING, MARGINWIDTH, MARGINHEIGHT,…

- `<noframe>…</noframe>` is to display a message for browsers that do not support frames.

Targeting a Frame

- You can target a different frame to open an HTML document.
  ```html
  <frame src="myPage.html NAME=main">
  …
  <a href="NEWPAGE.html TARGET=main">
  ```

- There are predefined names _blank, _top, _parent and _self.
Why Avoid Frames?

- The concept of a web document changes.
- The meaning of the “Back” and “Forward” becomes confusing to some users.
- Poorly designed frames can get the user lost and frustrated.
- It is difficult to find the URL of a HTML file contained in a frame. ➔ difficult to bookmark, print, etc.
- Some browsers do not support frames.
- Framed documents are not validated by SGML.

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
  - Image Maps
  - Validating HTML pages
  - Cascading Style Sheets

What are Image Maps?

An image-map is an image that serves as a menu: Clicking different areas lead to different URLs.

Server-Side Image Map

- `<A href="mycgi.cgi">
  <IMG SRC="MyImage.gif" ISMAP>
</A>`
- ISMAP causes the image to be used as a server-side image map. The browser send the coordinates of the point clicked to the cgi on the server. The cgi determines what to do.
- Rarely used today (connection overhead).
Client-Side Image Map

- `<IMG SRC="MyImage.gif" USEMAP=mapname>`
- `<MAP NAME=mapname>…</MAP>`
- The MAP tag section identifies areas that could be clicked.
- `<AREA ALT="…" HREF="…"…>`
- Parameters could be:
  - SHAPE (RECT, CIRCLE, POLY), COORDS
  - Also NOHREF

Adding Comments

- Comments can be added to an HTML file.
- Comments are ignored by the HTML interpreter.

```html
<!-- These are comments -->
<!-- These are also comments -->
```

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

HTML Validators

- Browsers try to guess how to render pages in presence of incorrect HTML.
- Browsers don’t guess the same way.
- When there is incorrect HTML the result is unpredictable.
- With errors, some pages may still look fine with some browsers but not with others.
- Some browser specific HTML is non conform to the standards.
- W3C HTML validation service http://validator.w3.org
Outline of Lecture 4

• What is HTML?
• Basic HTML
• Building Lists
• Tables
• Frames
• Image Maps
• Validating HTML pages
• Cascading Style Sheets

What are Style Sheets?

• HTML contained tags to indicate how to render pages: tags for structure and tags for style.
• The idea of style sheets is to separate page content and page style (text vs. colour, font, format, etc.).
• Cascading Style Sheets (CSS) is a technology used as an addition to HTML and gives a sophisticated way to specify how content should be rendered (in style).

Cascade of Rules

• CSS are sets of style rules to customize HTML elements.
• Style rules can be: external (imported), inline (within an HTML element), or embedded (declared in a document).
• They could different styles applying for the same text section.
• There are general rules for determining the precedence (cascading) of the styles.

Specifying Style Rules

The syntax for specifying style properties is:

\[ \text{selector \{property : value\}} \]

or

\[ \text{selector \{property1 : value1; property2: value2; ... propertyN : valueN\}} \]

Notice it is \text{property:value} NOT \text{property=value}.
**Style Properties**

- There are many properties pertaining to: font, size, colour, background, margins, borders, width, height, alignment, text appearance, etc. (and even position as we shall see later)
- Not all properties are recognized by both major browsers Netscape and IE.
- Conform to W3C CSS-level1 and CSS-level2.

**Define Rules**

- The tag `<STYLE>` allows the definition of formatting rules

```
<STYLE>
<!--
Style rules
-->
</STYLE>
```

**Example**

```
<!--
BODY {font: 12pt Helvetica; color:blue; margin-left: 0.5in}
H1 {font: 18pt Palatino; color: red}
H2 {font-family: MeppDisplayShadow}
KBD {text-decoration: underline}
-->
</STYLE>
```

**Style Sheets Advantages**

- Separation of text content and displaying style
- Possibility to create external style templates
- Consistent rendering of style throughout site
- No need for new HTML tags for new styles
- The end of a war?

**External Style Sheets**

- Style sheets can be kept separately from the HTML document.
- Possible re-use of the same style sheets with different HTML documents.
- Use the `<LINK ...>` tag in the document HEAD.
- `<LINK REL=STYLESHEET HREF="mystyle.css" TYPE="text/CSS">`
- Style sheets file should be WWW-accessible.
Embedded Style Sheets

• We can add style information in the document HEAD.
• The formatting rules apply for the whole document.

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Style Sheets Demo 1</title>
<style>
H1 {text-align:center; color:blue; font-family:Arial}
H2, H3 {text-decoration:underline; font-style:italic}
</style>
</head>
<body>
...[HTML content]
</body>
</html>
```

Inline Style Sheets

• We can create style rules within a document directly inside an HTML element tag.
• The formatting rules apply the section.
• The selector in the rule is omitted.

```html
...<p style="font: 10 pt Arial; line-height: 12 pt; margin-left: 0.5in; margin-right: 0.5cm; color: green; font-weight: bold">This paragraph will be displayed as specified</p>
```

Applying the New Styles

• External and embedded style rules will automatically apply when HTML elements they are associated with are used.
• Inline style rules apply where they are defined.
• The application of the style rule on the content starts at the beginning of the opening tag and end at the closing tag.
• What about applying styles for non tag contained text?

The <SPAN> Tag

• When we want to apply a style to part of a document that is not contained between an opening and closing tag, we can use the <span> ... </span> tag.

```html
<ol type="A">
  <li>my first element</li>
  <li>my second element</li>
  <span style="font-style:italic; color:red">is element is normal</span>
</ol>```

Example
User Defined Classes

• One can create classes of selectors associate a style to them and then use them to apply the style on sections of the document.
• For example we could define an abstract paragraph type as being in italic, 10 point text with some left and right margins: 

```html
<style>
P.abstract {font-style:italic; font: 10pt Palatino; margin-left:0.5cm; margin-right:0.5cm} </style>
```

• The new class is used as follows:

```html
<p class="abstract"> this is the abstract</p>
```

Generic User Defined Classes

• One can also define a class that is not attached to a particular HTML element:

```html
c.blue {color: blue; font-weight: bold}
```

• The new class can be used with different HTML element or even with non contained text:

```html
<header class="cblue">My header is blue</header>
<span class="cblue">this text is also blue</span>
```

Cascading Rules

If multiple style rules apply to the same section of text, one rule has to be selected based on precedence:

1. Rules marked “important” have highest priority:
   - `H1 {font-style: normal !important; color:red}`
2. Author’s rules have precedence over reader’s rules.
3. More specific rules have precedence over less specific rules.
4. In case of a tie, the last rule specified has priority.

Positioning in HTML

• HTML allows the description of content, text images, etc, and provides some structure.
• CSS provides a wonderful way to define and maintain consistent rendering style.
• Content is positioned automatically by the browser in “natural” positions.
• Frames and borderless tables help better position objects in a page.
• Difficult and limited.
CSS-P for Positioning

- `LAYER` and `ILAYER` in the early beta release of Netscape 4.0
- `LAYER` was rejected by W3C but still in use with Netscape
- CSS-P (Cascading Style Sheets Positioning) is an extension to CSS that allows pixel-level control over the position of HTML elements.
- Based on the `LAYER` idea

What is a Layer?

- A Layer is an area that may contain text or objects and different layers can overlap.
- Layers are implemented differently in IE and Netscape.
- What follows is cross-browser valid
- Do not use `LAYER` tag, use `DIV` tag
- `DIV` tag, `DIV` element, `DIV` block, CSS-layer and `LAYER` are all synonyms and simply refer to the same thing: a layer.

Example 1: Content Positioning

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head><title>Style Sheets Demo 1</title></head>
<body bgcolor="#FFFFFF">
<div id="mydiv" style="position:absolute; left:100; top:50; width:80;">HTML goes here</div>
</body>
</html>
```
Example 2: Content Positioning

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Style Sheets Demo 2</title>
</head>
<body bgcolor="#FFFFFF">
<div id="mydiv" style="position:absolute; left:100; top:50; width:300; height:100; clip:rect(0,300,100,0); background-color:red; layer-background-color:red;">
</div>
</body>
</html>
```

What’s Next?

- After seeing HTML, the Cascading Style Sheets and the positioning capabilities, the next step is dynamic web pages, or **Dynamic HTML**.
- Dynamic HTML is a term used to describe HTML pages with dynamic content.
- There are three components in dynamic HTML:
  1. HTML
  2. Cascading Style Sheets (CSS)
  3. JavaScript
- The three components are glued together with **DOM**, the Document Object Model.