



HTML

Qiang Yang
Simon Fraser University

Thanks: Francis Lau, HKU.
Thanks: Ethan Cerami, New York University

HTML for Web Publishing

- HTML stands for **HyperText Markup Language**
- The *lingua franca* for publishing hypertext on the World Wide Web
- It is a non-proprietary format based upon SGML
- HTML uses tags such as `<h1>` and `</h1>` to structure text into headings, paragraphs, lists, hypertext links etc.
- XHTML 1.0** is W3C's recommendation for the latest version of HTML
 - A reformulation of HTML 4.01 in XML, and combines the strength of HTML4 with the power of XML

1/12/01 HTML 2

History

- Originally developed by TBL while at CERN, and popularized by the NCSA Mosaic browser
- HTML 2.0 developed under the aegis of the Internet Engineering Task Force (IETF) in 1994
- HTML+ (1993) and HTML 3.0 (1995) proposed much richer versions of HTML
- World Wide Web Consortium's HTML Working Group's effort resulted in HTML 3.2 (January 1997)
- HTML 4.0** (December 1997)
 - Style sheets, scripting, frames, embedding objects, improved right to left and mixed direction text, richer tables, and enhancements to forms, for people with disabilities

1/12/01 HTML 3

HTML and HTML Editors

- Why should I care about how HTML works if I could use FrontPage, Netscape Composer, Dreamweaver, GoLive, etc. to create my pages?
 - There are certain neat features you can include in your pages that are only accessible through the code (JavaScript elements for example)
 - HTML is defined by an external group; it is possible for HTML to have features defined that aren't supported by your HTML editor (or by your browser)
 - Web page editors don't necessarily generate correct HTML code, nor clean code
 - Server-side scripts need to generate HTML code

1/12/01 HTML 4

An Example

```
<html>
<head>
<title>A simple HTML document</title>
</head>
<body>
<h1>A demonstration of simple HTML</h1>
<p>Here is a simple paragraph.
<p>Here is the <b>second</b> paragraph.
<address>Francis Lau / fcmlau@csis.hku.hk</address>
</body>
</html>
```

1/12/01 HTML 5

Kinds of Markup

- Structural** markup
 - `<p>`, `<h1>`
- Stylistic** markup
 - ``
- Descriptive (semantic)** markup
 - `<title>`, `<address>`

1/12/01 HTML 6

■ Creating Links

- A Tags for creating hyperlinks
 - href: URL for the hyperlink
 - that's it!
- An Image Link:
- A Text Link:

```
<A HREF="http://www.wired.com"></a>
<A HREF="http://www.wired.com/wired/archive/
3.06/xanadu.html?person=tod_nelson&topic_set=wiredpeople">
Full Text of Article</A>
```

1/12/01

HTML

7

■ Font Tags

```
<FONT FACE=ARIAL SIZE=6>
  <B>The Curse of Xanadu</B>
</FONT>
<FONT FACE=ARIAL SIZE=3>
  by By Gary Wolf, <I>Wired Magazine</I>
</FONT>
```

- Font tags:
 - face: Arial, Courier, etc.
 - size: e.g. 3, 6
 - color: e.g. "RED", "GREEN", etc.

1/12/01

HTML

8

■ <P> v.
 Tags

-
: Break
- <P>: Paragraph tag. Creates more space than a BR tag.
- <HR>: Creates a Horizontal Rule

1/12/01

HTML

9

■ Lists

- HTML supports two types of Lists:
 - Ordered Lists (OL): e.g. 1,2,3
 - UnOrdered Lists (UL): e.g. bullets.
- Basic Syntax:

```
<UL>
  <LI>Item 1
  <LI>Item 2
</UL>
```

1/12/01

HTML

10

■ Embedding Audio

- For Internet Explorer, you can use BGSOUND:

```
<BGSOUND src="http://www.3-
cities.com/~yogi/Starwars/impressive.wav">
```
- Netscape doesn't support BGSOUND, so you need to use the EMBED tag:

```
<EMBED SRC="http://www.3-
cities.com/~yogi/Starwars/impressive.wav"
controls="console">
```

1/12/01

HTML

11

■ Tables: Basic Tag Structure

```
<TABLE>
<TR>
  <TH>Ticker</TH>
  <TH>Price</TH>
</TR>
<TR>
  <TD>MSFT</TD>
  <TD>71 1/16</TD>
</TR>
<TR>
  <TD>KO</TD>
  <TD>46 15/16</TD>
</TR>
</TABLE>
```

TR: Table Row

TH: Table Heading

TD: Table Data

Every <TD> must have a matching <TD>. Every <TR> must have a matching <TR>.

1/12/01

HTML

12

HTML Frames

- Enables you to divide a page into parts.
- Each part can be served by the same or different web server.
- Very simple to use.
- Use with Caution:
 - Some browsers don't support frames
 - Reduces screen "real estate"
 - Much better way: use DHTML

1/12/01

HTML

13

Frame Example 1.0

```
<HTML>
<HEADER><TITLE>Framesets Example</TITLE></HEADER>
<FRAMESET ROWS="60%,*">
  <FRAME src="http://www.onvia.com">
  <FRAME src="http://www.office.com">
</FRAMESET>
  You are using a browser that does not support frames.
</NOFRAMES>
</FRAMESET>
</HTML>
```

Divides the screen horizontally

Specify Individual Frames

Used by old browsers.

1/12/01

HTML

14

HTML Colors

- You have two options for specifying colors:
 - Specify the color name, e.g. blue, maroon, pink.
 - Specify the RGB value.
- RGB Values indicate the amount of Red, Green and Blue within each color.
- These are specified as Hexadecimal numbers.
 - First Two Digits: Amount of Red
 - Next Two Digits: Amount of Green
 - Last Two Digits: Amount of Blue

1/12/01

HTML

15

HTML Forms

Forms Overview

- Every form must have a start `<form>` tag and an end `</form>` tag.
- ```
<FORM>
...
</FORM>
```
- Note that the form tag also has two attributes:
    - Method
    - Action

1/12/01

HTML

17

```
<HTML>
<HEAD>
<TITLE>Form Example 1.0</TITLE>
</HEAD>

<BODY>
<CENTER>
...
<FORM ACTION="http://www.someone.com/servlet/FormServlet"
METHOD="POST">
First Name:
<INPUT TYPE="TEXT" NAME="first" SIZE=20 MAXLENGTH=20>

Last Name:
<INPUT TYPE="TEXT" NAME="last" SIZE=20 MAXLENGTH=20>

Password:
<INPUT TYPE="PASSWORD" NAME="password" SIZE=20 MAXLENGTH=20>

<INPUT TYPE="SUBMIT" VALUE="Submit">
</FORM>
</CENTER>
</BODY>
</HTML>
```

Start of Form Tag

End of Form Tag

1/12/01

HTML

18

## More on the Form Method

- Historical meaning:
  - POST: Used to "post" news messages.
  - GET: Given an ID, go "get" the new message.
- This is now confusing, since you can use either one. So, why use one over the other? Let's look at another example....

1/12/01

HTML

19

```
<HTML>
<HEAD>
<TITLE>Form Example 2.0</TITLE>
</HEAD>
<BODY>
<CENTER>
...
<FORM ACTION="http://www.someone.com/servlet/
FormServlet" METHOD="GET">
First Name:
<INPUT TYPE=TEXT NAME=first SIZE=20 MAXLENGTH=20>

Last Name:
<INPUT TYPE=TEXT NAME=last SIZE=20 MAXLENGTH=20>

Password:
<INPUT TYPE=PASSWORD NAME=password SIZE=20 MAXLENGTH=20>

<INPUT TYPE=SUBMIT VALUE="Submit">
</FORM>
</CENTER>
</BODY>
</HTML>
```

HTML

20

## URL Encoding with GET

- When submitting data for Example 2.0, note that the user data is now appended to the URL:
- <http://www.ecerami.com/servlet/FormServlet?first=Eth&last=Cerami&password=blue>
- URLEncoding:
  - Path and first piece of user data are separated with a ? character.
  - After that, each piece of user data is separated with an & character.

1/12/01

HTML

21

## GET v. POST

- If GET and POST end up with the same result, why choose one over the other?
  - Some servers limit the **length** of GET requests to 240 characters. Hence, use POST for large forms.
  - If **security** is an issue, use POST. This will not encrypt the user data, but at least you cannot easily view the data by inspecting the URL.
  - If you want to create **links** that act like forms, use GET.

1/12/01

HTML

22

## History of HTML Standards

- HTML first created by Tim Berners Lee.
- HTML standard is controlled by the World Wide Web Consortium
- <http://www.w3c.org>
- Current Official Versions of HTML: XHTML 1.0, HTML 4.0
- Unfortunately, W3C has tended to lag behind the Browser Developers.

1/12/01

HTML

23

## Browser Wars

- First Browser: Created by Tim Berners Lee
  - First browsers were primarily text based.
- NCSA Mosaic: First popular web browser
  - Created at the University of Illinois
  - Created by Marc Andreesson, who went on to found Netscape Communications.
  - First browser to include embedded images.
- Netscape: First commercial browser.
  - originally had 90% of market
- Microsoft Internet Explorer
  - now: ~85% of market

1/12/01

HTML

24

## HTML Validation

- What is an HTML validator?
  - Checks that your HTML adheres to the W3C standards.
  - Checks HTML syntax.
    - Example: do all your start table tags have an end table tag?
    - Example: did you include an end `</BODY>` tag?
  - Makes sure that all embedded links work.
  - Checks for cross-browser support.
  - Depending on the service, other validators provide spell checking, performance analysis, etc.

1/12/01

HTML

25

## Some Free Validators

- <http://validator.w3.org>
- <http://www.netmechanic.com>
  - Great service for: checking HTML, spell checking, cross-browser support, link checking, even performance analysis.
- Let's try them out...

1/12/01

HTML

26

## Web Accessibility

## Disabilities and Access

- The Web was created for everybody, but 8% people have disability.
- People with certain disabilities, however have difficulty accessing web pages.
- World Wide Web Consortium has a working group to address issues of disabilities and the web.
- Web Accessibility Initiative:  
<http://www.w3.org/WAI/>
- It may soon be law that any company that receives federal funding provide Web accessibility.

1/12/01

HTML

28

## Most Important Tips

- Web Accessibility Initiative has created a series of HTML tips for making your sites more accessible.
- Most tips focus on making pages comprehensible to blind users who use voice browsers / screen readers.

1/12/01

HTML

29

## Accessibility Tip #1

- **Images & animations**
  - Provide ALT tags, describing the content of the visual elements.
  - For example:  
`<IMG ALT="W3C logo" BORDER=0 SRC="logo.gif">`
  - The more important and/or complex the image, the more descriptive your ALT tag should be.

1/12/01

HTML

30

## ■ Accessibility Tip #2

- **Check your work.**
  - Nothing throws off a screen reader more than non-standard, incomplete or incorrect HTML tags.
  - Hence, use an HTML Validator.

1/12/01

HTML

31

## ■ Accessibility Tip #3

- **Organize your pages well.**
  - Use headings (<H1>, <H2>, <H3> etc), lists and consistent structure.
  - Where possible, break up long lists into smaller sublists, with appropriate labels or headings.
  - This also helps screen readers organize content more easily.

1/12/01

HTML

32

## ■ Accessibility Tip #4

- **Hypertext links**
  - Links should be descriptive of their destination.
  - For example, avoid "click here."

1/12/01

HTML

33

## ■ Accessibility Tip #5

- **Frames**
  - Use frames cautiously, always providing NOFRAMES content and giving titles to each frame.
  - Remember to keep your NOFRAMES content up-to-date.

1/12/01

HTML

34

## ■ Accessibility Tip #6

- **Preview in various browsers**
  - including old versions and a text only browser, such as Lynx to ensure that your pages are still intelligible.
  - If the page is still intelligible in Lynx, it's probably still intelligible in a screen reader.

1/12/01

HTML

35

## ■ Some More Tips

- **Multimedia:**
  - Provide captioning and transcripts of audio, and descriptions of video.
- **Scripts, applets, & plug-ins.**
  - Provide alternative content in case the features are inaccessible or unsupported.
- For a full list of tips, go to:  
<http://webaim.org/Design/quick-tips>

1/12/01

HTML

36



## HTML and Web Performance

---

### ■ Performance Tip #1

- Validate your HTML
  - Missing ends tags can delay rendering of an entire page.
  - Particularly acute in Netscape Navigator.

1/12/01

HTML

38

### ■ Performance Tip #2

- Add height and width attributes to all your images.
- Example:
  - ``
- Given width and height tags, the browser is able to render the HTML page before actually downloading the images.
- Speeds page rendering significantly.

1/12/01

HTML

39

### ■ Performance Tip #3

- Reduce the number and size of the images on your page.
- The fewer images you have, the faster the site.
- For example: compare the speed of Yahoo to that of CNN.com.
- To reduce the size of your images, try <http://www.netmechanic.com/accelerate>

1/12/01

HTML

40

### ■ Performance Tip #4

- Reuse images as much as possible.
- If you place the same image, e.g. a bullet image on four different pages, the browser will only download it once.
- This takes advantage of browser caching.

1/12/01

HTML

41

### ■ Performance Tip #5

- Reduce the number of embedded tables.
- Some web pages have 3,4, even 5 levels of embedded tables.
- The fewer levels of embedding, the faster the page will render.

1/12/01

HTML

42

## ■ Performance Tip #6

- Pick a page weight standard and stick to it.
- For example, all pages, including embedded images should be 20K.
- For example: check out any page on Yahoo.