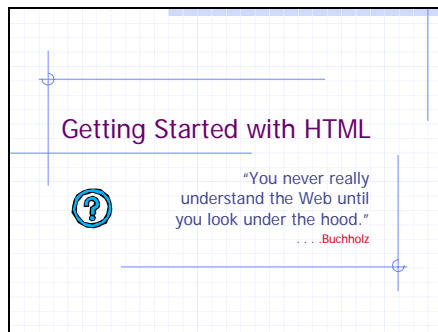


Getting Started with HTML

Dr. William James Buchholz
Professor and Chairman
Information Design and Corporate Communication
Bentley College
175 Forest Street
Waltham, MA 02452-4705
wbuchholz@bentley.edu
781-891-2216

The complete PowerPoint presentation is available as a Web slideshow at
http://cyber.bentley.edu/faculty/wb/presentations/html_files/frame.htm

Slide 1



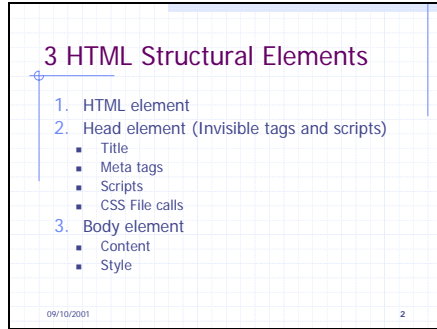
You often hear that HTML is easy to learn, that you can master all you really need to know in twenty-four hours—even two hours if your goals are not set too high. Well, I suppose there is some truth to these claims. That, after all, is why Tim Berners-Lee developed HTML: to quickly produce unadorned text pages that could be moved easily over the internet. Content—not design, graphics, interactivity, multi-media, or e-business—ruled the internet in 1990.

Today on the World Wide Web, however, most people are not happy with just a simple page of text. They want to make visual statements. They want multifunctional, interactive, database driven, world-class designed Web sites.

But . . . you don't achieve that overnight. It takes Web designers years of experience, and patient craftsmanship, to produce the elegant and impressive sites that you visit on the Web. In fact, professional and commercial sites are usually designed, implemented, and maintained by teams of professionals: designers, writers, programmers, database designers, graphic artists, HTML specialists, and information architects. In BC 370, you will learn a little about each of these.

This course, however, will start you primarily on the road to page design, and, time permitting, introduce you to some elements of site design. But remember, BC 370 is only an introduction to the possibilities of the Internet. We will just begin to scratch the surface of this extraordinary potential for communication: the World Wide Web.

Slide 2



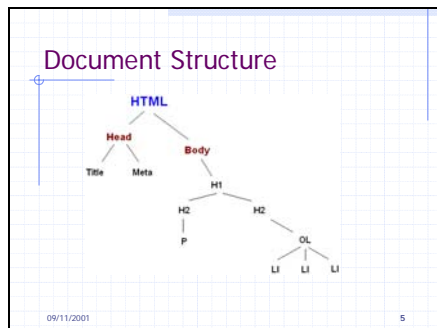
Let's start modestly. If you understand the fundamentals and build upon them, you could well become a first-rate creator of Web pages.

So, what is HTML? *HyperText Markup Language* is a specially selected set of markup tags that tell a browser how to render text, images, program code, and applications in your computer. You need both the markup language and the browser to produce what you see on the Web.

The first step in understanding HTML is to learn a little about its structure. HTML is highly structured—in fact, Berners-Lee selected the language (from SGML) so that it could easily render information hierarchically.

Three elements make up any page on the WWW. First, the page begins with a declaration that it is tagged in HTML. Second, the page contains the head element. Usually the most important item in the head is the title of the page. All the interactive JavaScripting, DHTML, and animated fireworks that you've grown so fond of also have core pieces of code lodged in the head element. And third, the page contains the body element—everything that makes up the substance of a Web page: text, images, hyperlinks, interactive forms, applications, programs, streaming video, sound—you name it, the body contains it.

Slide 3

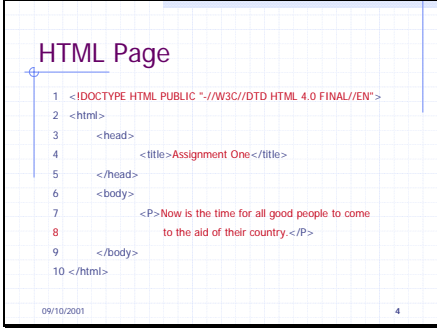


The HTML document structure is a hierarchy that we often refer to as **root/branch** or **parent/child**. The root element from which all else grows is HTML. The primary branch from the root (what we might call the trunk) is the Body element. Branching from the trunk are the various HTML elements, each with their own potential branches. Given this metaphor, can you define the “leaf”?

Now switch the metaphor to ancestry. The Body element becomes a child to its HTML parent. The H1 is seen as a **child** to its **parent** the Body element and its grandparent, the HTML element. LI has OL as its parent, H2 as its grandparent, H1 as its great-grandparent, BODY as its great-great-grandparent, and HTML as its great-great-great-grandparent (the primal **ancestor**). LI1 is a **sibling** to LI2 and to LI3. LI1 is a **descendant** of OL, which is a descendant of H2, a descendant of H1, a descendant of Body, and finally, as are all elements, a descendant of HTML.

The HTML document model simply reflects hierarchical relationships. Thus, every object in a Web page bears a namable relationship to

every other object. Because of this relative relationship, HTML descendant elements can inherit traits from their parents. The idea of **inheritance** becomes very important when you are formatting a page using Cascading Style Sheets (CSS).

Slide 4A slide titled "HTML Page" showing a 10-line HTML document structure. The code is as follows:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 FINAL/EN" >
2 <html>
3   <head>
4     <title>Assignment One</title>
5   </head>
6   <body>
7     <p>Now is the time for all good people to come
8     to the aid of their country.</p>
9   </body>
10 </html>
```

The slide also includes a date "09/10/2001" in the bottom left and a small number "4" in the bottom right.

Here is a brief look at the HTML source of a simple Web page. Note that line one—the doc type declaration—declares the document type, stating that it conforms to the W3C (World Wide Web Consortium) Document Type Definition (DTD) finalized 4.0 version of HTML in English. With this declaration, your browser knows what kind of document it is dealing with. The line is optional, however, because browsers are programmed by default to render HTML.

HTML source is unformatted ASCII* text. It is plain vanilla, with nothing graphically fancy. The whole purpose of the HTML source is simply to tell the browser how to render the page graphically. HTML uses the bracketed items, known as tags, to send these instructions to the browser. As you see here, tags usually come in pairs that set off portions of text. Tag pairs have an opening and a closing tag. Everything in between the opening and closing tags is rendered specifically by the way the browser interprets HTML.

What page would the browser render based on this 10-line HTML source?

***ASCII:** (American Standard Code for Information Interchange) Pronounced "ask-ee." A binary code for text as well as communications and printer control. It is used for most communications and is the built-in character code in most minicomputers and all personal computers. (From Computer Desktop Encyclopedia, <http://www.techweb.com/encyclopedia/>)

Slide 5

Head Element (Invisible)

```

1 <html>
2 <head>
3 <title>Assignment One</title>
4 <meta name="description" content="Information design offers
  advice and various resources for presenting information effectively in
  written, oral, and Web communication. Please use any of the
  PowerPoint presentations, tutorials, scholarly articles, site addresses, or
  course materials for your own purposes.">
5 <meta name="keywords" content="information, design, internet,
  www, w3c, intranet, XML, HTML, Buchholz, Bentley">
6 </head>

```

09/10/2001 5

Note that the head element is paired; it must have an opening tag `<head>` and a closing tag `</head>`. The `<title>` is also a paired element: `<title> . . . </title>`.

Meta tags, however, are not paired, that is, they do not set off material that is rendered on the page. The meta tag simply consists of an open bracket (`<`) and a close bracket (`>`). The material between the brackets is important, of course, but not for rendering, *per se*.

Slide 6

Body Element

```

1 <body bgcolor="#FFFFFF">
2 <table border="1">
3 <tr>
4 <td>
5 <h2>Abstract</h2>
6 </td>
7 </tr>
8 </table>
9 </body>

```

Rendered Page

Abstract

09/10/2001 6

The body element consists of the rendered content of the Web page. The body tag in this example is declaring a white background for the page. The HTML source contains all the tags necessary to produce a one-cell table, with a one-pixel border, containing the word “Abstract” formatted as a level-2 heading. (Note the rendered page illustrated on the right side of the slide.)

Slide 7

Anatomy of a Block Element

Element

- Start Tag and End Tag
- Attribute and Value

```

Attribute Value
<a href="http://cyber.bentley.edu/faculty/wb/courses/380/index.html">BC 380</a>
Start tag End tag
A (anchor) Element

```

09/10/2001 7

HTML is a markup language consisting of elements with contained tags. Block elements have start and stop tags that enclose the material they are marking up. So-called empty elements have a single tag (for example: BR, HR, IMG). Elements can have various characteristics that we call “attributes.” Attributes, when they are assigned meaning are said to have “values.” Thus, in the anchor element example here, the attribute is “href” and its value is “http://cyber.bentley.edu/faculty/wb/courses/380/index.html”.

The start tag consists of everything before the phrase being blocked: BC 380. The end tag consists of everything after the phrase. End tags are usually very short, as in this example. All this markup is kept from the viewer. The browser will render this simply as the blue underlined hyperlink with which we have all become so familiar on the Web: BC 380

Slide 8

Block-level Elements

- html and body
- div
- h1-h6
- p
- br
- blockquote
- hr
- ol
- ul

09/18/2001 8

Listed here are the most commonly used block-level elements. These are referred to as “block-level” because of the way they affect the formatting of the text and segregate material inside the opening and closing tags.

Slide 9

In-line Elements

- a
- b
- i
- big
- small
- img
- span

09/10/2001 9

The text-level elements produce common format changes that affect text rendering. For example, here is how the word “Bentley” would be tagged in HTML to render in bold text:

```
<B>Bentley</B>
```

Text can be rendered in Italics `<I></I>` as well. Or it can be made relatively larger `<big>big</BIG>` or smaller `<Small>little</smALL>`

Note: HTML is not case-sensitive. It does not matter if your tags consist of lower-case or upper-case letters—or any mix of the two. XHTML is case-sensitive. Therefore, I recommend that you keep your HTML tags consistently in lower-case, especially as your server may be case-sensitive (UNIX). By keeping case consistently low, you will never have a problem in establishing links to files and images on the Bentley server.

Slide 10

Hyperlinks: Named Anchors

- > HTML named anchor:
`Bentley`
- > Example:
[Bentley](http://www.bentley.edu)
- > Key:
 - `<a` = anchor
 - href = hypertext reference
 - http = hypertext transfer protocol
 - Domain name (server name) ``

09/18/2001 10


Hyperlinks are the *piece de resistance* of the WWW. The whole power of the Web resides in hyperlinks. The ability to surf—to proceed in a nonlinear fashion through page after page, site after site anywhere in the world at any time—is possible because of the simple HTML anchor tags that you see here.

Translated, the HTML source is saying that the text between the opening and closing anchor tags is a hypertext reference to the Bentley domain accessed through the *HyperText Transfer Protocol* (HTTP) via the *Uniform Resource Locator* (URL) `www.bentley.edu`

Slide 11

Hyperlinks: Image Links

- > HTML named anchor:


```
<a href="http://erebus.bentley.edu/empl/c/rcrooks/toolbox/html/contents.html">
</a>
```
- > Example: 

09/10/2001 11

Of course, images that can actually perform a function make the Web more exciting. Everyone who creates a Web page wants hyperlinked images (probably animated). Here is how you hyperlink an image.

The `<a href>` opening tag precedes the call to an image (the `img src` tag). The image source is a file named `html.jpg` residing in a folder named `images370`. This folder is accessed outside the folder containing the document with this HTML source call, hence the need to use `../`. If this last part is a little confusing, don't worry. I'll explain this further when talking about absolute and relative URLs. Right now, let's continue the analysis of the image call.

Some alternative text is given that will appear before the image is rendered (or instead of the image if the visitor has the browser image rendering toggled off). The alternative text will also appear in the browser's tool tip (the little box that appears momentarily when you place a cursor over an image). The image itself is borderless (0 pixels), 180 pixels wide by 31 pixels high.

Note the use of = signs and quotation marks. The = sign indicates that what follows is an *attribute* (a declared value) of an element. Declared values must reside within quotation marks.

All this HTML source results in one tiny hyperlinked button!

Slide 12

Valid HTML

- > Three structural elements
- > Complete block tags (open & close)
 - `<p> . . . </p>`
 - ` . . .`
- > Proper nesting of block tags
 - `<p>my name <i>is Bill</i></p>`
 - `<p>my name <i>is Bill</p></i>`
- > Quotation marks around attribute values
 - `align="center"`
 - `border=0`

09/10/2001 12

To be valid, an HTML page must have all three structural elements, completed block tags (that is, both the opening and the closing tags), all block tags properly nested, and both opening and closing quotation marks around attribute values.

The final sub-bullet in each example on the slide is incorrect. Do you see why?

Slide 13

Tips

- Learn the HTML basics so that you can peek under the hood when you're stranded.
- Keep your HTML clean: it's the key to tomorrow's |Web:
 - CSS
 - DHTML
 - XML
 - Content management systems in business

09/10/2001 13

Throughout the entire semester, I will advise you in performing your troubleshooting activities. At some point, however, I expect that you will become fairly independent in troubleshooting. That is our goal this semester: to have you achieve a comfort level and degree of competence that will make you a technically proficient creator of excellent Web content.

Slide 14

What's the Future of HTML?

- W3 Consortium: "[HTML 4.01](#) is a revision of the XHTML 4.0 Recommendation first released on 18th December 1997."
- "The [XHTML 1.0](#) spec relies on HTML 4.01 for the meanings of XHTML tags." Released as a recommendation January 2000.
- XHTML 1.1 Released as a recommendation May 2001.
- Thus, HTML is no more, and much has been deprecated. Enter XHTML and CSS.

09/10/2001 14

It's a little disheartening for students to realize that in learning HTML they are studying a "dead" language. Somewhat an exaggeration to call it "dead," nevertheless, it is true that HTML will have no further incarnations. XHTML 1.0 has replaced it. HTML pages are now part of the legacy Web. XHTML is the future Web, the *semantic Web*. Don't despair, though. Your efforts to learn HTML won't be fruitless. XHTML is really HTML cleaned up and validated, dependant upon CSS for stylistic and formatting effects.

XHTML is an XML application, which means that the future Web will adhere to stricter standards of control through *semantic* or *logical* markup rather than *physical* markup. Content and style have now become largely separated so that content can be repurposed (reused). This means that information will be more readily accessible across all platforms, becoming truly universally accessible.

Slide 15

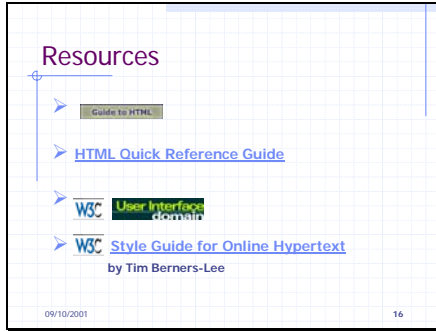
What is Deprecation?

- Certain elements and attributes in HTML 4.01 have been declared obsolete or "deprecated." CSS in most cases should achieve their "stylistic and formatting effects."
- A list of deprecated:
 - [Elements](#)
 - [Attributes](#)

09/10/2001 15

In this migration from HTML to XHTML and CSS, certain HTML elements and attributes have been deprecated, declared "dead." For now and in the near future, however, browsers will continue to render these obsolete structures. But the future holds their ultimate disappearance. Among your favorite elements now deprecated: *font* and *center*. Of the elements: *align*, *alt*, *background*, *face*, *size*, and *color*.

Slide 16



Familiarize yourself as soon (and as often) as possible with these recommended resources. You will be putting this material into effect immediately.