

# CSC 551: Web Programming

Fall 2001

## Basic HTML

- hypertext
- tags & elements
- text formatting
- hyperlinks
- images
- style sheets

## Hypertext & HTML

HyperText Markup Language (HTML) is the language for specifying the 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 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 attempts to map out future directions for HTML, not just react to vendors

## 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)

*for most users who want to develop basic, static Web pages, these are fine*

assembly language vs. high-level language analogy

so, why are we learning low-level HTML using a basic text editor?

- may want low-level control
- may care about size/readability of pages
- may want to "steal" page components and integrate into existing pages
- may want dynamic features such as scripts or applets

## Tags vs. elements

HTML specifies a set of *tags* that identify structure and content type

- tags are enclosed in `< >`  
`` 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>
<!-- Dave Reed  html01.html  8/16/01  -->
<!-- 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>

[view page in browser](#)

## Text layout

```
<html>
<!-- Dave Reed  html02.html  8/16/01  -->
<!-- Demo web page          -->

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

<body>
  This is a whole lot of text that
  goes on and on and
  on
  and

  on
  .
  .
  .
</body>

</html>
```

the BODY can contain multiple  
lines of text

- text layout and spacing is pretty  
much ignored by the browser
  - every sequence of whitespace is  
interpreted as a single space
  - browser automatically wraps the  
text to fit the window size
- ➔ can layout text in an HTML  
document for readability, will not  
affect how it is viewed

[view page in browser](#)

## Overriding default layouts

```
<html>
<!-- Dave Reed  html03.html  8/16/01  -->
<!-- Demo web page          -->

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

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

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

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

for the most part, layout of the text must be left to the browser  
**WHY?**

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: &nbsp;

[view page in browser](#)

## Separating blocks of text

```
<html>
<!-- Dave Reed  html04.html  8/16/01  -->
<!-- Demo web page          -->

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

can specify headings for paragraphs or blocks of text

- <h1>...</h1> tags produce a large, bold heading
- <h2>...</h2> tags produce a slightly smaller heading
- ...
- <h6>...</h6> tags produce a tiny heading

can insert a horizontal rule to divide sections

- <hr> draws line across window
- <hr width="50%"> sets width
- <hr size=10> sets thickness
- <hr color="red"> sets color

[view page in browser](#)

## Aligning text

```
<html>
<!-- Dave Reed  html05.html  8/16/01  -->
<!-- Demo web page          -->

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

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

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

  <p align="right">
    Here is some right-justified text.
  </p>
</body>
</html>
```

can specify whether elements should be centered

- `<center>... </center>`

*Note: the CENTER element is deprecated in HTML 4.0  
→ still can be used, but a preferred method exists*

HTML 4.0 recommends using paragraph tags with an alignment attribute

- `<p align="center">...</p>`
- `<p align="right">...</p>`

[view page in browser](#)

## Example Web page

Consider the [CSC 551 Home Page](#)

- page title and section information is centered
- paragraphs/sections for different topics  
(course description, required work, collaboration policy, schedule)
- headings to identify each section
- breaks and spaces used to align text, improve layout

## Text styles

```
<html>
<!-- Dave Reed  html06.html  8/16/01 -->
<!-- Demo web page          -->

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

<body>
  <p>
    Text can be emphasized using
    <b>bold</b>, <i>italics</i>, or even
    <big>resizing</big>. <br>
    <u>Underlining</u> text is not
    generally recommended since it looks
    too much a like a hyperlink. <br>
    The typewriter font is good for
    displaying code:
    <small><tt>sum = sum + i;</tt></small>
  </p>
</body>

</html>
```

[view page in browser](#)

### can specify styles for fonts

- `<b>... </b>` specify bold
- `<i>... </i>` specify italics
- `<u>... </u>` specify underlined
- `<tt>... </tt>` specify typewriter-like (fixed-width) font
- `<big>... </big>` increase the size of the font
- `<small>... </small>` decrease the size of the font

*Note: if elements are nested, the order of opening/closing is important! (must be LIFO)*

## More text styles

```
<html>
<!-- Dave Reed  html07.html  8/16/01 -->
<!-- Demo web page          -->

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

<body>
  <p>
    <font color="red">Subscripts</font>
    (e.g., x<sub>1</sub>) and
    <span style="color:red">superscripts
    </span> (e.g., 2<sup>10</sup>)
    can be embedded directly in text.
  </p>

  <p>
    In order to avoid affecting line
    spacing, usually it should be made
    smaller (e.g.,
    <small>2<sup>10</sup></small>).
  </p>
</body>

</html>
```

- `<sub>... </sub>` specify a subscript
- `<sup>... </sup>` specify a superscript
- `<font color="red">... </font>` change the font color  
*constants are defined for many colors*  
*the FONT tag is deprecated in HTML 4.0, instead they recommend:*
- `<p style="color:red">...</p>` for paragraphs
- `<span style="color:red">...</span>` for inline text

[view page in browser](#)

## More text grouping

```
<html>
<!-- Dave Reed  html08.html  8/16/01  -->

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

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

  <p>
    Eagleson's Law states that:
    <blockquote>
      Any code of your own that you haven't
      looked at for six or more months
      might as well have been written by
      someone else.
    </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*

[view page in browser](#)

## Lists

```
<html>
<!-- Dave Reed  html09.html  8/16/01  -->

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

<body>
  <p>
    <ol>
      <li>First thing.
      <li>Second thing.
      <li>Third thing.
    </ol>
  </p>

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

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

*can set type of ordering, start index*

- `<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

[view page in browser](#)

## Hyperlinks

```
<html>
<!-- Dave Reed  html10.html  8/16/01  -->

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

<body>
  <p>
    <a href="http://www.creighton.edu">
      Creighton University</a>
    <br>
    <a href="html09.html" target="_blank">
      Open html09 in a new window</a>
    </p>
</body>

</html>
```

[view page in browser](#)

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>
<!-- Dave Reed  html11.html  8/16/01  -->

<head>
  <title>Title for 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
    </dl>
  </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

[view page in browser](#)

## Hyperlinks (cont.)

```
<html>
<!-- Dave Reed  html12.html  8/16/01  -->

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

<body bgcolor="white" link="blue"
  alink="red" vlink="darkblue">

<p>
  <a href="http://www.cnn.com">CNN</a>
</p>

</body>
</html>
```

[view page in browser](#)

### By default,

- hyperlinks appear as blue
- while you are clicking, turns red
- after you have visited, turns purple

### can redefine these colors by specifying attributes in the BODY tag

- BGCOLOR is background color
- LINK is the unvisited link color
- ALINK is the active link color
- VLINK is the visited link color

## Images

```
<html>
<!-- Dave Reed  html13.html  8/16/01  -->

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

<body>
  <center>
    
  </center>
</body>
</html>
```

[view page in browser](#)

### Can include images using <IMG>

- by default, browsers can display GIF and JPEG files
- other image formats may require plug-in applications for display
-   
again, if file is to be accessed over the Web, must start with http://  
if not, the browser will assume it is a local file

more on images next week

## Example Web page revisited

Consider the [CSC 551 Home Page](#) again

- bold, italics, and font size changes are used throughout to embellish text
- does not have colored fonts (use sparingly!)
- uses an unordered list to present the course goals
- uses BODY attributes to specify link colors
- has links to University page, department page, instructor page
- links to lecture notes bring up a new window

## Style sheets

modularity is key to the development and reuse of software

- design/implement/test useful routines and classes
- package and make available for reuse
- saves in development cost & time
- central libraries make it possible to make a single change and propagate

HTML 4.0 emphasizes the use of *style sheets* in order to package and reuse style formats

- idea is to separate structure/content (HTML) and presentation (Style Sheets)
- central libraries make it possible to make a single change and propagate

## Inline style sheets

```
<html>
<!-- Dave Reed  html14.html  8/16/01  -->

<head>
  <title>Title for Page</title>
  <style type="text/css">
    <!--
      body {background-color : white}
      a:link {color : blue}
      a:visited {color : purple}
      a:active {color : red}
    -->
  </style>
</head>

<body>
  <a href="http://www.cnn.com">CNN</a>
</body>
</html>
```

[view page in browser](#)

Using `<style>` tags, can move formatting specifications into the HEAD

- specify element, then sequence of attribute:value pairs

*Careful:* current browsers do not support all style sheet features

## Inline style sheets (cont.)

```
<html>
<!-- Dave Reed  html15.html  8/16/01  -->

<head>
  <title>Title for Page</title>
  <style type="text/css">
    <!--
      h1 {color : blue;
          text-align : center}
      p.indented {margin-left : 1in}
    -->
  </style>
</head>

<body>
  <h1>Center Title</h1>

  <p>This is a normal paragraph.</p>

  <p class="indented">
    This is an indented paragraph.</p>

</body>
</html>
```

Can define extensions to existing elements

- p.indented defines a new class of paragraphs
  - inherits all defaults of `<p>`
  - adds new features

to specify this newly defined class, place `class="ID"` attribute in tag

[view page in browser](#)

## Careful with style sheets

```
<html>
<!-- Dave Reed  html16.html  8/16/01 -->

<head>
  <title>Title for Page</title>
  <style type="text/css">
    <!--
      a {color : red;
        text-decoration : none;
        font-size : larger}
      a:hover {color : blue}
    -->
  </style>
</head>

<body>
  Visit the
  <a href="http://www.cnn.com">CNN</a>
  web site.
</body>
</html>
```

[view page in browser](#)

Can completely change the look of HTML elements

- not usually a good idea to diverge too much from user expectations

*Careful:* current browsers do not support all style sheet features

## Modularity & style sheets

```
<!-- myStyle.css  Dave Reed  8/16/01 -->
hl {color : blue; text-align : center}
p.indented {margin-left : 1in}
```

```
<html>
<!-- Dave Reed  html17.html  8/16/01 -->

<head>
  <title>Title for Page</title>
  <link rel="stylesheet"
        type="text/css"
        href="myStyle.css"
        title="myStyle">
</head>

<body>
  <h1>Center Title</h1>

  <p>This is a normal paragraph.</p>

  <p class="indented">
    This is an indented paragraph.</p>

</body>
</html>
```

To take full advantage of style sheets, can place style definitions in separate file, link to that file

- any changes can be localized to that one style file

ideally, place all formatting options in a central style sheet

all Web pages link to that same style sheet for a uniform look

- simplifies Web pages since only need to specify structure/content tags

[view page in browser](#)

## Style sheets summary: 3 options

### 1) if need to set the style of a single element

- place style attribute in corresponding tag

```
<span style="color:red">red text </span>
```

### 2) if need to set the style for similar elements in a page

- place style definitions with STYLE tags in the HEAD

```
<style type="text/css">
  <!--
    a {color : red;
      text-decoration : none;
      font-size : larger}
    a:hover {color : blue}
  -->
</style>
```

### 3) if need to set the style for similar elements across pages

- place style definitions in separate file, link in using tag in HEAD

```
<link rel="stylesheet"
      type="text/css"
      href="myStyle.css"
      title="myStyle">
```

## Web rules of thumb

HTML provides for lots of neat features,  
but just because you can add a feature doesn't mean you should!

don't add features that distract from the content of the page

- use color & fonts sparingly and be careful how elements fit together  
e.g., no purple text on a pink background, no weird fonts
- use images only where appropriate  
e.g., bright background images can make text hard to read  
e.g., the use of clickable images instead of buttons or links can slow access
- don't rely on window or font size for layout  
e.g., font size may be adjusted by viewer, window constrained
- don't be annoying  
e.g., no pop-up windows, excessive advertising, silly music
- break large document into smaller or provide a menu to internal links
- stick to standard features and test using both IE and Netscape

## Next week...

### advanced HTML

- tables
- more on style sheets
- frames
- images

### Read Chapters 7-8, 10-11

### As always, be prepared for a quiz on

- today's lecture (moderately thorough)
- the reading (superficial)