HTML Basics

Instructor: Lisa Fiedor
E-mail: lisa_fiedor@ncsu.edu
Phone: 513-4616

DELTA Instructional Services
http://delta.ncsu.edu
Supporting you in using technology for teaching and learning.

Overview:

HTML editors such as Dreamweaver make it easy to create Web pages without knowing how to write HTML code. However, if you understand the basics of how HTML works, you can more easily create and edit your pages, regardless of whether you use editing software or you (gasp!) code by hand. Knowing how HTML works can also be useful for instructors who want additional design control in the Vista environment. This two-hour hands-on workshop will cover HTML document structure and basic HTML tags; by the end of the class, participants will be able to create and edit their own basic HTML documents using any text editing application. Some attention will be given to using HTML in Dreamweaver and in Vista. Participants should be comfortable working on a PC in a Microsoft Windows environment and should be familiar with basic functions of the operating system, such as opening, closing, moving, copying, and saving files.

Objectives:

Skills developed in this workshop include:

- To learn basic HTML tags.
- To learn about HTML attributes.
- To develop a basic Web page by hand coding an HTML document in a text editor.

Table of Contents:

What is HTML?

HTML stands for Hypertext Markup Language. But it's not really a "language;" it's a special kind of word processing that uses plain ordinary text, with some special characters called "tags" that you add to achieve certain results. Once you understand how tags work, you can create pages in HTML. When you use an HTML editor like Dreamweaver to create a Web page, the tags are still being created, but the editor hides them from you unless you instruct it not to.

Though most Web developers these days use an HTML editor like Dreamweaver, it is possible to create and modify an HTML document with any program that can edit a plain text file, such as Notepad (Windows) or TextEdit (Mac). However, editing HTML code directly can be a tedious and laborious process. Editors like Dreamweaver are designed to edit and manage HTML documents, and they make the process of creating and modifying Web pages much more efficient.

HTML pages are typically *viewed* with a browser, such as Mozilla or Internet Explorer. A browser simply interprets the HTML tags and displays the HTML page accordingly.

Exercise: View the HTML source code of a Web page.
1. Open a Web browser and navigate to a website of your choice.
2. Under the View menu, select "Source" (this might be labeled differently depending on your browser).

Tags are all you really need to know!

The heart of creating HTML code is the use of tags, which indicate how you want different parts of your page to display. Web browsers "read" the tags and display the page according to the instructions found in the tags. The tags themselves are not displayed when you view the HTML page in a browser; they are only used by the browser.

- **Tags are created by using angle brackets (< and >), also known as the "less than" and "greater than" signs.**

Here are some examples; note that in each example, there is both a "start tag" and an "end tag."

```
<strong>affected text</strong> = displays text in boldface
<em>affected text</em> = displays text in italics
<p>A very short paragraph.</p> = these tags set off a paragraph.
```

- **The start tag contains a command, such as "p" for paragraph, and the end tag contains the same command with a forward slash (/) in front of it.**

In most cases, to create an effect on a Web page, you simply put the appropriate tags before and after the text you want to be affected by the command in the tag.

For example, if you want a line of text to be bold, you would put the start strong tag, `<strong>`, at the point you want the bold lettering to start and the end strong tag, `</strong>`, where you want the bold lettering to end. If you want certain text to appear in italics, you would use the emphasis tags, `<em>` and `</em>`. In most cases, both a start tag and an end tag are required, but we will see some exceptions to this later.

Here are some examples of how tags might look in your HTML code:

```
<strong>This whole sentence will appear in
bold.</strong>
But here, only a single <em>word</em> is in italics.
```

- **You can also apply multiple tags to a section of text.**

For example, what if I wanted a word to appear in both bold AND italics? I have to open and close both tags, making sure the pairs are nested rather than overlapping:

```
        The meeting starts at 7:30
<strong><em>sharp!</strong></em>
```

List of all tags

http://www.w3schools.com/tags/default.asp
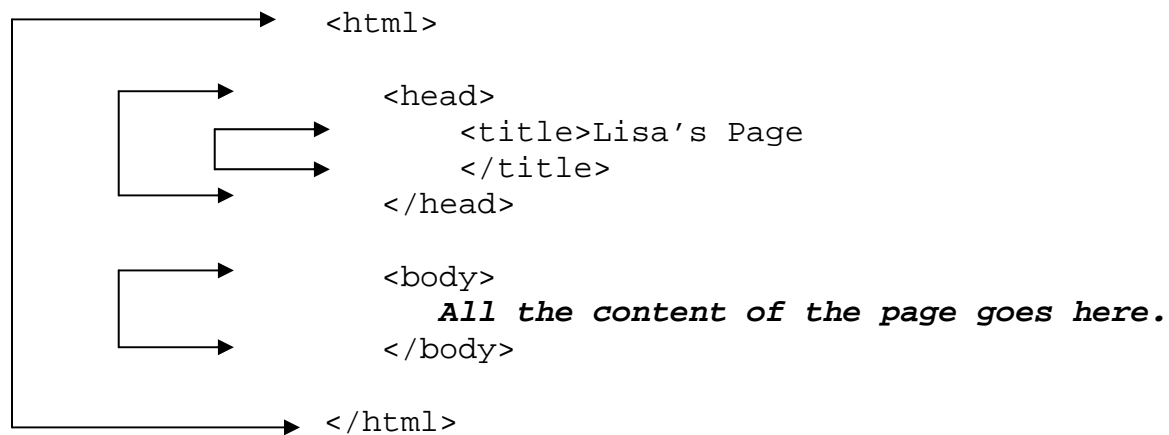
Document structure tags

These are special tags that are used by the Web browser to determine the structure and organization of your document. In other words, they tell the browser (1) that this is an HTML document, and (2) where the basic parts of the document will be found.

These document structure tags are:

```
<html> ... </html> (encloses entire document)
<head> ... </head>  (encloses HEAD section)
<title> ... </title> (encloses document title)
<body> ... </body> (encloses BODY section)
```

- **The four sets of tags above are important ones; you should include them in every HTML page you make.** An editor such as Dreamweaver adds these tags automatically when you create a new document.

This example shows how these tags should be arranged, and how they help define the structure of your HTML document. Note that the HTML tags are at the beginning and end of the document, the HEAD and BODY sections are separate, and the TITLE is found inside the HEAD section. All the content of your page (everything that will appear in the browser) goes inside the BODY section.

```
<html>

    <head>
        <title>Lisa's Page
        </title>
    </head>

    <body>
        All the content of the page goes here.
    </body>

</html>
```

A note about spacing: Blank spaces are irrelevant in HTML code. The tags above are indented to illustrate their nested relationship (and many people indent their pages this way as they create them). However, you could make everything flush left, or put it all in one long line of text if you wanted. The browser will ignore all spaces (or lack of spaces) in your code when it displays your page.

One more thing … Attributes and their values

Attributes are commands *located inside of tags,* which give additional instructions to the browser displaying the HTML page. Many tags will work just fine without using any attributes, but in some cases attributes provide a lot more power and flexibility, and in other cases using attributes is required for the tag to mean anything.

The way these always look is `attribute="value"` Here are some examples of tags with attributes; in each case the attribute and its value are in bold.

```
<p align="center"> . . . </p>
<body bgcolor="#0000ff" link="#ff0000"> . . . </body>
<img src="logo.gif" alt="NC State University" />
```

In-class activity

We will create a simple HTML document, and edit the document by adding a series of tags. After each change we make, we will save the document, and then view the changes in a Web browser.

1. Open a blank document in Notepad

Notepad is normally found in the Start Menu under Programs → Accessories.

2. Add document structure tags

Add the document type tag at the top (<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">) which will tell the browser to which language and rules your page is adhering, in this case XHTML Transitional. Then, to create the basic structure of the document, add the HTML, HEAD, TITLE, and BODY tags, following the model on page 3 of this handout. Note that the TITLE tags should go inside the HEAD tags, and both the HEAD and BODY sections go inside the HTML tags. At this point, there will be nothing inside the BODY tags. Within the TITLE tags, give the page a name of your choice (you can always change it later).

3. Save As HTML

Save the file *as an HTML file*. Normally, Notepad will save as plain text, so we have to tell it not to do this. Select File → Save As, and change "Save as type" from "Text document" to "All files." Type a file name, such as "syllabus.html", in the "File name" box. You can give your file an .htm or an .html extension. Save the file in the "html_basics" folder on the desktop.

4. Open the HTML page in a browser

If you haven't already done so, open a Web browser. Locate the HTML file you just saved and open it (Use File → Open, for example). *Does the browser display anything?*

5. Adding content

The page content should all go between the BODY tags. You can type in a couple of paragraphs if you want, or to save time, you can copy/paste the text from the "sample_text.txt" document located in the "html_basics" folder. After you make the changes, SAVE YOUR HTML PAGE (File Menu → Save; or Ctrl-S on the keyboard), then switch to your browser and Refresh (or Reload) the page. You should see the changes you made take effect.

*6. Paragraphs <p> and Line Breaks
*

Add paragraph tags `<p>  </p>` to blocks of text to indicate the presence of paragraphs. Paragraphs in HTML pages display with a blank line between them. To create a simple line break, use the `<br  />` tag at the point you want the break to occur. *Note:* the `<br  />` tag does

NOT require a separate end tag, but it does require the / before the > to close it).   Save the file, switch to your browser, and refresh the page.

7.   Text formatting tags \<strong\> \<em\> and the CENTER attribute

Add some **bold** and some *italics* to selected text on your page. Use `<strong> ... </strong>` for bold and use `<em> ... </em>` for italics. Center one of the paragraphs using the `align="center"` attribute.

8.   Heading tags \<h1\>, \<h2\>, etc.

Make a top-level heading (the largest size) by using the Heading1 tags <h1> ... </h1>, with the text of the heading between them. Save, switch to browser, Refresh. Play around with adding a different level heading to another paragraph if you want.

9.   Horizontal Rule \<hr /\>

The HR tag is used to create a horizontal line (or "rule") across the page. This tag does NOT require a separate end tag; a single HR tag will create the line.

The ALIGN and WIDTH attributes can also be used as follows:

```
<hr align="center" width="50%" />
```

The above tag creates a horizontal line positioned in the center of the page, with a width that is 50% of the browser window.

10. Adding Images \<img\>

The image tag is `<img>`, and as in the previous case, you don't need to use a separate end tag. You do, however, need to use attributes to make the IMG tag meaningful. The SRC attribute (which means "source") is used to tell the browser what image to display when it finds an IMG tag.

Add the following tag at the very beginning of the BODY section (but make sure it's between the start and end BODY tags, since that's where all our page content goes):

```
<img src="lis456_banner.gif">
```

You can add multiple attributes inside of tags. Here is the same image tag with several more possible attributes added. Note that the value of each attribute is enclosed in quotation marks.

```
<img src="lis456_banner.gif" width="600" height="100" alt="LIS 456 banner">
```

Note: It's good practice to include the ALT attribute whenever you use an image in a Web page. It provides descriptive information about the image without requiring that the image itself be visible.

11. Hyperlinks <a href= " ">

There are two things to consider when making a link. You need to know what are you linking to, and you need to know what a user will click on to activate the link. Here is the basic structure of what a hyperlink looks like in HTML code:

```
<a href="links.html">Visit the links page.</a>
```

In this example, the words, "Visit the links page" will be the link text that will be clickable, and the file "links.html" is the page the browser will open when the link is clicked.

To link to pages outside your site on the Web, you must use the entire URL of the site you're linking to. Note the second example below where the clickable link is an image rather than text.

```
<a href="http://www.google.com/">Click here for Google</a>
<a href="http://www.ncsu.edu"> <img src="ncsulogo.gif" alt=NCSU
Logo"> </a>
```

To create an e-mail link, use the following format:
```
<a href="mailto:lisa_fiedor@ncsu.edu">click here to e-mail
me</a>
```

Add the appropriate links for the text on you page.

12. Tables <table> <tr> <td> <th> <caption>

A table is a collection of information (words, numbers, etc.) arranged in rows and columns in boxes known as cells. To create a table in HTML, there are a few sets of tags that you need to know about. Other table tags do exist, but these are the basics. TR means "table row," TH means "table header," and TD means "table data."

Creating tables by hand coding HTML can be a very tedious process. Here is a table with one header row and two rows of data:

```
<table>
  <caption>This is my simple table</caption>
  <tr>
    <th>First Header Cell</th>
    <th>Second Header Cell</th>
  </tr>
  <tr>
    <td>data here</td>
```

```
      <td>more data here</td>
    </tr>
    <tr>
      <td>data here</td>
      <td>more data here</td>
    </tr>
</table>
```

Convert the evaluation percentages into a table.

**12. Lists, ordered and unordered <ol> **

HTML allows you to create two kinds of lists: ordered (numbered) and unordered (bulleted). For either type, use the <li> tag to mark your list items. Here are two lists, one of each type.

ordered (numbered) list: unordered (bulleted) list:

```
<ol>                                        <ul>
  <li>First</li>                              <li>Item</li>
  <li>Second</li>                             <li>Item</li>
  <li>Third</li>                              <li>Item</li>
</ol>                                       </ul>
```

Convert the lists in sections 2 and 3 to HTML lists.

13. Summary

In this class, we have learned what HTML is, and how tags and attributes can be used to create effects in HTML documents. We have also learned about the basic structure of an HTML document. We have taken a blank text document and created an HTML page from scratch, adding document structure tags, paragraphs and line breaks, text formatting (bold, italics, and centering), headings of various sizes, horizontal rules, images, hyperlinks, tables, and lists.

Moving on from here…

This class has involved the creation of a very simple HTML page, but you can do much more complex things in HTML simply by learning about different tags and attributes. There are lots of ways to do this; here are a few to consider:

1. **Get a book to refer to.** There are lots of HTML books out there, written for people of all different levels of experience. Browse through a few until you find one that seems to make sense to you. It can be handy to keep a book nearby to refer to when you are working on your pages. There are also "pocket" reference guides to HTML that include lists of tags. The last three pages of this handout describe several basic HTML tags.

2. **Examine other sites you like.** Use your browser's "View Source" ability to examine the HTML code of other sites. Doing this can help you understand how various effects are created in HTML, and how to duplicate these effects on your own pages.

3. **Work through a Web tutorial.** There are numerous Web sites that offer tutorials and other tips and tricks related to writing HTML. You can go through these tutorials at your own pace and learn a lot by doing them. Here are a couple:

> HTML Goodies: http://www.htmlgoodies.com/tutors/
> Webmonkey: http://hotwired.lycos.com/webmonkey/teachingtool/index.html
> W3 Schools http://www.w3schools.com/html/default.asp

4. **Feel free to contact me.** If you think of a question later, I'll be glad to help. My contact information (and DELTA's) is on the front page of this handout. You may call me or any of the DELTA staff (learntech@ncsu.edu) and we will be happy to help you figure something out.

Reference

Document Structure Tags	
`<html>` `</html>`	Indicates the beginning and end of your entire HTML document
`<head>` `</head>`	Indicates the beginning and end of the HEAD section of the document. The TITLE must go in this section.
`<title>` `</title>`	Indicates what title will appear in the title bar of the browser when the page is viewed; must go in the HEAD section
`<body>` `</body>`	Indicates the beginning and end of the BODY section of the document. The content of your page goes in this section. Common attributes used inside this tag: BGCOLOR = sets background color; TEXT = sets color of normal text; LINK = sets color of hyperlinks (often blue); VLINK = sets color of links already visited (often purple); BACKGROUND = sets an image file as a tiled background image. See color section for more color information.

Text Formatting Tags	
`<strong>` `</strong>`	Makes text in **bold**
`<em> </em>`	Makes text in *italics*

Separators	
`<h1> </h1> ... <h6> </h6>`	Creates headings; number sets size between 1 (largest) and 6 (smallest)
` `	Creates a break at the end of a line of text. The browser will continue the text on the next line. There is not a separate end tag

`<p> </p>`
 	
`<p align="right"> </p>`
	Sets the beginning and the end of a paragraph. Browsers typically display a blank line between paragraphs automatically. The "ALIGN" attribute can be used to align the entire paragraph to the left, center, or right.

`<hr />`
 	
`<hr width="30%" align= "center">`
	Creates a line or "horizontal rule" across your page; there is not a separate end tag.
	
	Some attributes that can be used with HR include "width" and "align".

delta

Lists

`<ol> </ol>`	Creates an "ordered list" (numbered)
`<ul> </ul>`	Creates an "unordered list" (bullets)
`<li> </li>`	Creates a list item; use with either or

Anchors (Links)

`<a href="URL">Hotspot</a>`
 	
example:
`<a href="http://www.cnn.com">Click for CNN</a>`
	Anchor linking "Hotspot" to a particular URL, so when Hotspot is clicked, browser opens that URL

`<a href="mailto:lisa_fiedor@ncsu.edu">E-Mail Me!</a>`	Anchor linking to an e-mail address; will normally open a blank e-mail message when clicked

Images

`<img src="mypicture.jpg">` (This tells the browser to look in the same directory that the HTML file is in to find the image file. The SRC attribute will be different if the image is located elsewhere.)	Inserts image into document; the two most common image types are .gif and .jpg (or .jpeg) Common attributes: alt = alternate text height = height of image in pixels width = width of image in pixels border = width of border in pixels, can also be set to 0.

Tables

`<table> ... </table>`	Defines the entire table; all of your table's information goes between these tags.
`<tr> ... </tr>`	Defines a table row; each row is enclosed in a set of these tags
`<td> ... </td>`	Defines a data cell; each cell is enclosed in a set of these tags
`<th> ... </th>`	Similar to <td> but defines a header cell (usually displays in bold).
`<caption> ... </caption>`	Defines a caption which displays below the table.

How to define colors in attributes

Many current browsers can interpret names of colors, so attributes like BGCOLOR="RED" will work, but for best results across all browsers, you may choose to use hexadecimal codes to define colors. These codes consist of the pound sign (#) and six characters, all enclosed together within quotation marks. The characters can be any number from 0 to 9 and any letter from A to F. The first two characters indicate Red, the second set of two indicates Green, and the third set of two represents Blue. Combining these three values produces all the Web-safe colors.

```
<body bgcolor="#000000">
```
means the background color is black (all 3 values set to zero)
```
<body bgcolor ="#FF0000">
```
means the background color is red (first value set to Full)
```
<body bgcolor ="#0000FF">
```
means the background color is blue (third value set to Full)
```
<body bgcolor ="#FFFFFF">
```
means the background color is white (all 3 values set to Full)

Web sites for more information about hexadecimal color codes:
```
        http://www.zspc.com/color/index-e.html
        http://hotwired.lycos.com/webmonkey/reference/color_code
        s/
```

Please fill out our evaluation at http://delta.ncsu.edu/workshops/evaluation

Notes