



Intro to HTML



Topics

- ◆ Background & naming documents
- ◆ Anatomy
- ◆ Headings, font size, alignment
- ◆ Linking
- ◆ Images
- ◆ Tables
- ◆ XML, metadata
- ◆ Resource sites



HTML

- ◆ short for **hypertext markup language**
 - **Hyper** is the opposite of linear. HTML does not hold to the old linear pattern but allows the person viewing the Web page to go anywhere, any time
 - **Text** is what you will use
 - **Mark up** is what you will do
 - **Language** it's a language, really -- but the language is plain English



Writing HTML

- ◆ by hand
 - using tools such as NotePad on Windows
 - OS X users can use TextEdit on the Mac
- ◆ using an HTML assistant program
 - easier than by hand
 - but harder to understand HTML because the program does all work



HTML documents

- ◆ must be text only
- ◆ when you save an HTML document, you must save only the text, nothing else.
- ◆ HTML browsers can only read text
 - They don't understand anything else



Learning from others

- ◆ look at others' Web pages
- ◆ When you find a page you like, click on VIEW at the top of the screen
- ◆ Choose DOCUMENT SOURCE from the menu. Sometimes it only reads SOURCE.
- ◆ The HTML document will appear on the screen.
- ◆ It may look like chicken-scratch but later you'll be able to find exactly how a certain HTML presentation was performed



HTML annotations

- ◆ Web page authors may write notes or describe what is happening within the HTML document
 - these notes show in the HTML source, not the Web page display
 - they write their notes in the following format:
- ◆ *<!-- start of syllabus and definitions -->*



Naming HTML documents

- ◆ Basic format--name and suffix
- ◆ Follow this format to name your document:
 - Choose a name. Anything.
 - Add a suffix. For all HTML documents, you will add either ".htm" or ".html".
- ◆ Name the file anything you want
 - Tree followed by .html so the file name is tree.html.
- ◆ *.html* tells the computer that this file is an HTML document
- ◆ All files used on the Web will follow the format of "name.suffix."



Basic HTML anatomy

- ◆ HTML works in a very simple, very logical, format
- ◆ reads from top to bottom, left to right
- ◆ HTML is written with TEXT
- ◆ Tags are used to set certain sections apart and to specify their format (as bigger text, smaller text, bold text, underlined text)



Basic HTML anatomy--tags

- ◆ tags are like commands
- ◆ To make a line of text bold put a tag at the exact point you want the bold lettering to start and another tag where you want the bold lettering to stop
- ◆ To make a word italic place a start italic tag at the beginning of the word and an end italic tag at the end of the word



Basic HTML anatomy--tags

- ◆ All tag formats are the same
 - They begin with a less-than sign: < and end with a greater-than sign: >
 - What goes inside the < and > is the tag
 - Learning HTML is learning the tag to perform whatever command you want to do
- ◆ The tag for bold lettering is "B". Here's what the tags look like to turn the word "Sam" bold:
`Sam`



Basic HTML anatomy--tags

- ◆ What happened "behind the scenes"
 - `` is the beginning bold tag
 - "Sam" is the word being affected by the `` tag
 - `` is the end bold tag. Notice it is exactly the same as the beginning tag except there is a slash in front of the tag command
 - what the bold tags above produce in a Web browser: **Sam**



Basic HTML quick points

- ◆ The end tag is simply the begin tag with the added slash `<a> `
- ◆ Not all tags will show up on a page, because the commands are placed inside the `<` and `>` marks
 - the tag is used to alter the text, but unless you view the page source the code is hidden from view.
- ◆ The command inside the `<>` does not have to be a capital letter; the browser doesn't care
 - for ease when coding keep the tag in caps



More HTML quick points

- ◆ not everything on a web page needs to have tags
- ◆ if you want to format the text with italics or underlines or bold, or if you want to center text, etc., you **will** need to use tags
- ◆ if you forget to add an end tag it will be obvious when you view the document in your browser
 - the entire document after the point where you forget the end tag will be affected
 - to fix, go back into the document, add the slash, save, and then reload



More HTML quick points

- ◆ two tags can affect text at the same time
 - make sure to begin and end both
 - `<I>Bold and Italic</I>` gives you ***Bold and Italic***
- ◆ when using more than one tag:
 - set the beginning and end tags at the same time
 - always placing them on the farthest end of the item being affected
 - note above that the Bold tags are on the far ends, with the Italics next
 - set commands at the farthest ends each time you add them



And the last HTML quick points

- ◆ start every page with this tag: `<HTML>`
 - next tags will always be these: `<TITLE>` and `</TITLE>`
 - whatever is between these two tags will show up in the title bar way at the top of the browser
- ◆ end every page you write with this tag: `<HTML>`
 - you started the page with `HTML` and you will end the page with `/HTML`



Headings

- ◆ Heading commands are used to create headings
 - there are six (6) heading commands: <H1> through <H6>
 - <H1> is the largest and <H6> is the smallest
 - heading commands create nice, bold text with a simple H# and /H# command
- ◆ When using a heading command you set the text alone
 - the heading commands carry a pseudo <P> command with them
 - other text cannot sit right up against a heading



Font size

- ◆ heading commands great for the top of the page
- ◆ get a little more control over your text size via the `` commands
 - there are twelve (12) font size commands available: +6 through +1 *and* -1 through -6
 - +6 is the largest (it's huge); -6 is the smallest (it's a little too small)



Aligning text

- ◆ Default alignment is left-justified
- ◆ To center text you surround the text you want centered with simple `<CENTER>` and `</CENTER>` commands:

```
<CENTER> All text in here will be centered </CENTER>
```
- ◆ To align text on the right, set the text aside as a separate paragraph using the `<P>` command plus an attribute:

```
<P ALIGN="right">Text in the paragraph is pushed to the right</P>.
```



Creating a link

- ◆ links to another page are a set tag format
`text
you want to display on the web page`
 - **A** stands for Anchor. It begins the link to another page.
 - **HREF** stands for Hypertext REFerence. That says to the browser, "This is where the link is going to go."
 - **URL** of the web site is the FULL ADDRESS of the link. Also notice that the address has an equal sign in front of it and is enclosed in quotes, because it's an attribute of the Anchor tag, a command inside of a command



Creating a link, cont.

- ◆ "text you want to display on the web page" is where you write the text you want to appear on the page
- ◆ What is in that space will appear on the page for the viewer to click--write something that denotes the link.
- ◆ /A ends the entire link command.



Adding an e-mail link

- ◆ known as mailto: commands
- ◆ follows the same coding scheme as a link
- ◆ places wording on the screen that people can click to send you a piece of e-mail
- ◆ pattern is:

```
<A HREF="mailto:e-mail  
address">text to display</A>
```
- ◆ same format as a link except you write "mailto:" in place of the "http://" and your e-mail address in place of the page address
 - still need the tag at the end
 - note there is NO SPACE between the colon and the e-mail address



Images

- ◆ format for placing an image:

```
<IMG SRC="filename.gif">
```

- **IMG** stands for "image" and tells the browser that an image will go here on the page wherever you write in the image tag
- **SRC** stands for "source" and this is an attribute, a command inside a command, that tells the browser where to go to find the image
- **filename.gif** is the name of the image, and this file name follows the same format as HTML docs
 - name (of the image file) then a dot
 - then there is a suffix (gif) or .jpg or .bmp



Image info

- ◆ place image files in the same directory as the page
 - you can call for the image by name alone
 - otherwise you'll have to start adding directories and sub-directories to the SRC attribute
 - some place all their images in an image directory; that can cut down on the confusion
- ◆ be consistent on where you locate images or else the image won't display



Image file types

- ◆ three basic image formats on the Web and they have different suffixes
 - **.gif** This is generally pronounced "gif" (hard "G"), an acronym for Graphics Interchange Format that browsers can handle quite easily
 - **.jpeg** or **.jpg** (pronounced "j-peg") an acronym for Joint Photographic Experts Group, and this format uses compression after it's been created
 - **.bmp** (pronounced "bimp") or a "bitmap." Internet Explorer browsers allow images as bitmaps (images a computer produces and places for you, such as a counter)



Clickable images

- ◆ An image where if you click on it you activate a hypertext link to another web page
- ◆ The format is:

```
<A HREF="http://URL of the web page"><IMG SRC="filename.gif"></A>
```
- ◆ Places an image tag where normally there would be words
- ◆ entire image is "clickable," or active



Image attributes

- ◆ use the "alt" attribute to provide alternate text to display when you hover over any image
- ◆ alternate text is especially important for users:
 - browsing with a text-only browser
 - that cannot see and are using audible readers with their browser to surf the web
- ◆ format is `<alt=brief description of image>` and follows after the IMG SRC tag:
``



Tables

- ◆ very useful for presentation of tabular information
- ◆ Useful to creative HTML authors who use the table tags to present their regular Web pages
 - tables can control page layout



Table format

- ◆ `<TABLE>`
- ◆ `<!-- start of table definition -->`
- ◆
- ◆ `<CAPTION>` caption contents `</CAPTION>`
- ◆ `<!-- caption definition -->`
- ◆
- ◆ `<TR>`
- ◆ `<!-- start of header row definition -->`
- ◆ `<TH>` first header cell contents `</TH>`
- ◆ `<TH>` last header cell contents `</TH>`
- ◆ `</TR>`
- ◆ `<!-- end of header row definition -->`
- ◆ `<TR>`



Table format, continued

- ◆ *<!-- start of first row definition -->*
- ◆ `<TD> first row, first cell contents </TD>`
- ◆ `<TD> first row, last cell contents </TD>`
- ◆ `</TR>`
- ◆ *<!-- end of first row definition -->*
- ◆
- ◆ `<TR>`
- ◆ *<!-- start of last row definition -->*
- ◆ `<TD> last row, first cell contents </TD>`
- ◆ `<TD> last row, last cell contents </TD>`
- ◆ `</TR>`
- ◆ *<!-- end of last row definition -->*
- ◆
- ◆ `</TABLE>`
- ◆ *<!-- end of table definition -->*



Publishing a Web page

- ◆ use an FTP program (File Transfer Protocol)
 - a small program that allows you to place files from your computer to your service provider's computer
 - get an FTP program and directions from your ISP
 - or check out utilities



XML

- ◆ acronym for Extensible *Markup Language*
- ◆ a set of rules
- ◆ published by the W3C (World Wide Web Consortium)
- ◆ for building new languages
 - languages in question are not written or spoken
 - they're intended to simplify information sharing among software and humans
 - languages, and the documents in which they're "written," share common characteristics
- ◆ XML delimits blocks of content with intelligible, structure-defining markup to *add meaning* to the content



XML documents

- ◆ add meaning with plain-text contents
- ◆ scatters little verbal signposts among the content
- ◆ impose a structure which is immediately understandable even if *what* is being structured is not obvious
- ◆ signposts are called *markup*
- ◆ special characters come into play
 - no XML document does *not* include the most important characters
 - the $<$, $>$, and $/$ (less-than, greater-than, and slash)



XML example

- ◆ the markup is in boldface between and angle brackets:

```
<sentence><clause>Benedict Arnold  
didn't<punctuation type="apostrophe" />  
cross the Delaware<punctuation  
type="semi-  
colon" /></clause><clause>he crossed his  
country<punctuation  
type="period" /></clause></sentence>
```



Legible XML example

- ◆ **<sentence>**

- <clause>**Benedict Arnold
didn't cross the
Delaware**<punctuation
type="apostrophe" />**t cross the
Delaware**<punctuation type="semi-
colon" />****</clause>**

- <clause>**he crossed his
country**<punctuation
type="period" />****</clause>**
</sentence>



XML quick points

- ◆ each clause is subordinate to the overall sentence
- ◆ within a clause there may be a mixture of the plain text and punctuation
- ◆ punctuation could have been left as literal text, rather than defined via markup
- ◆ the markup itself is human-readable
 - an elementary understanding of English grammar for the meaning of "sentence," "clause," and "punctuation"



XML quick points

- ◆ XML is all about ***well-formedness***
- ◆ well-formedness are the specific rules with which all XML documents must comply in order to be minimally legitimate XML

A vertical image on the left side of the slide shows a dark metal key with a circular bow and a notched bit, resting on a light-colored, textured surface. The key is oriented vertically, with the bow at the top and the bit at the bottom.

More XML well-formedness

- element and attribute names are case-sensitive (a SENTENCE element is not the same as a sentence element)
 - so is the corresponding markup
- attribute values must be enclosed in single or double quotation marks
- the nesting of one element within another, as defined by the placement of tags, is precise
 - *Every* start tag must be balanced with *one* end tag
 - no overlap of the boundaries between one element and the next is permitted



More XML points

- ◆ each well-formed XML document has one and only one "outermost element," within which all the others are nested
 - the *root element*.
- ◆ an XML document is a string of plain text
 - delimited by markup
 - in a well-structured form including a single root element and others, nested inside one another
 - XML was designed to describe data and to focus on what data is
 - HTML was designed to display data and to focus on how data looks



Metadata

- ◆ data about data
- ◆ the background information that describes:
 - the content
 - the quality
 - the condition
 - and other appropriate characteristics of the data
- ◆ paper maps contain metadata as part of the map legend
 - For the paper map, metadata is readily apparent and easily transferred between map producers and map users
 - For digital map data, metadata development and maintenance often requires a more conscious effort on the part of data producers and the chain of users who may modify the data

Questions?

