HTML Course Notes

Compiled By: Dr. Gaurav Sehgal

Associate Professor
School of Business Studies
Central University of Jammu, Jammu
J&k State - INDIA

HTML Course Notes

Compiled By: Dr. Gaurav Sehgal

Contents

- ✓ ***The Basics of HTML***
  - ➢ *What Is It*
  - ➢ *Abstraction*
  - ➢ *How To Write It*
  - ➢ *Basic Formatting*
  - ➢ *Framework*
  - ➢ *Paragraphs*
  - ➢ *Headings*
  - ➢ *Breaks And Rules*
- ✓ ***Directory Structure & Links***
  - ➢ *Directory Structure*
  - ➢ *Links*
- ✓ ***Lists***
  - ➢ *Lists*
  - ➢ *Closing Comments*
- ✓ ***Text Formatting in HTML***
  - ➢ *Bold And Italics*
  - ➢ *Tags With Attributes*
  - ➢ *Text Sizes*
  - ➢ *Font Tag*
  - ➢ *Alignment*
  - ➢ *List Attributes*
  - ➢ *Closing Observations*
- ✓ ***Images***
  - ➢ *Image Types*
  - ➢ *Image Tag*
  - ➢ *Attributes For Images*
  - ➢ *Text Images*
  - ➢ *Linked Graphics*
  - ➢ *More Image Attributes*
  - ➢ *Background Formatting*
- ✓ ***Tables***
  - ➢ *Basic Tables*
  - ➢ *Properties Of Tables And Cells*
  - ➢ *Coding Table wise Cell Properties*
  - ➢ *Coding Cell Properties*
  - ➢ *Individual Cell Properties*
  - ➢ *Row span and Column span*
  - ➢ *Advanced Uses*
  - ➢ *Table Headers*

- ✓ ***HTML – A Quick Reference***

These HTML Course Notes have been compiled taking in view the requirements of the students of UG as well as PG….

The students are further advised to refer to the prescribed Reference Books also…

The Basics of HTML

What is it?

HTML = HyperText Markup Language is the *markup language* used for *hypertext* (yeah, that says very little...)

A **markup language** is a system of *tags* that describe how to format a document. There's usually some kind of *interpreter* that reads a document in the markup language and displays it in a nicely formatted way.

Hypertext is the technical term for web pages, i.e. text with formatting and which the user can interact with (e.g. clicking links, hence the "hyper" part). The interpreter for hypertext is a *web browser*.

Tags and HTML Files

HTML files are plain text files --> no formatting

We need ways to tell the web browser when to start formatting text in certain ways, insert things like horizontal rules and links, etc. These are the **tags**. They're code that is understood by the writer and replaced by the interpreter with appropriate formatting.

Putting it in Perspective

HTML is not the only markup language. Just a few other examples:

- **LaTeX** -- a markup language used for typesetting mathematics; interpreter eventually produces a printed document
- **MathML** -- another markup language for math that's designed to be integrated with HTML and read by web browsers (Update: it's supported by Firefox, Safari, Chrome, Camino, and Opera now.)
- **XML** -- eXtended markup language, also used for the web, but the theory is to markup *what* something is and not how to format it; formatting handled separately

HTML & Abstraction

Abstraction is one the most fundamental concepts of computer science.

Definition: Ignoring the details of a complicated system to focus on the essentials needed to use the system. Focusing on WHAT we're doing, not HOW it is done.

Where do we use abstraction in our daily lives?

Why is this important?

HTML is a great example of abstraction in action. When we look at a web page in a browser, we are viewing the same thing as the HTML code, but at a different (higher) level of abstraction.

How do we write it?

HTML files are plain text files, so a simple *plain text editor* will work fine. The simple one in Windows is **Notepad**. The simple one in Mac OS X is **TextEdit**.

Notepad

It's usually found under **Start > Accessories > Notepad**.

Very limited set of commands, but if you can use Office, you can certainly learn Notepad.

When you save in Notepad, the default file type is **txt**. For HTML, you want the extension **html** or **htm**. To achieve this, type the filename in double quotes in the **Save As** dialog, e.g. **"index.html"**

Text Edit

Double-click the **Macintosh HD** icon and go to the **Applications** folder. Look for **TextEdit**.

I'd recommended right-clicking the TextEdit icon in the dock and choosing **Keep In Dock** so you don't have to hunt it down every time.

We want plain text, so we need to go to **Format > Make Plain Text** to achieve this.

Some first-time setup to make life much easier:

1. Go to the **TextEdit** menu > **Preferences**.
2. While **New Document** is selected, change the **Format** to **Plain Text**.
3. Go to the **Open and Save** section of the Preferences. Here,
    - *Un*check **Add ".txt" extension to plain text files**.
    - *Un*check **Ignore rich text commands in HTML files**.
4. Close the Preferences window.

Now, saving HTML files will always be easy. For HTML, you want the extension **html** or **htm**, so just type the filename in the **Save As** dialog, e.g. **index.html**.

A few simple tags: formatting

Most HTML tags travel in pairs.

Bold example

HTML	As displayed in browser
This is a sentence with ****some bold text****.	This is a sentence with **some bold text**.

There's a **** to say "start bold formatting here." The **** says "stop bold formatting here."

Notice that the tags are the same, except that there's a forward slash (/) in the closing tag.

Italics

Similarly, the tags for italics are **** and ****.

Example:

HTML	As displayed in browser
some kind of ****interpreter**** that reads	some kind of *interpreter* that reads

Problem

Write the HTML for the blocks of text on the right.

HTML	As displayed in browser
???	**HTML = HyperText Markup Language** is the *markup language* used for *hypertext*
???	is the ***markup language*** used for *hypertext*

Caution!

- Watch that you have a closing tag to go along with the opening tag.
- Watch that the closing tag has the /.

The HTML file itself is all wrapped within an outer set of tags common to all HTML files. Here's the basic framework (you could feasibly copy and paste this each time you get started):

```
<html>

   <head>
     <title>The title of your page goes here.</title>
   </head>

   <body>
     The text that makes up your page goes here.
   </body>

</html>
```

Observations

- Notice that blank lines (whitespace) are ignored. It's good style to include whitespace because the HTML needs to be readable.
- Notice that spaces are ignored if there's more than one of them at a time. You could put 20 spaces between words and it's the same as putting one. Tabs mean nothing.
    - There's a way around that called a non-breaking space, but there really isn't any place you'd need to -- or *should* -- use it.
    - Common mistake I've seen: *Do NOT* try to indent paragraphs with a bunch of non-breaking spaces. Paragraphs in HTML were never intended to be indented. It looks sloppy.

Paragraphs

Paragraphs in HTML are wrapped inside ** ** and ** ** tags. This line of text is an example of one paragraph. The next is a new paragraph.
Paragraphs (and most elements of HTML) are separated by blank lines in the browser.

Headings

HTML provides levels of **headers** or **headings** for organizing a document.
There are six levels of headers:

Heading 1	**Heading 2**	**Heading 3**	**Heading 4**	**Heading 5**	**Heading 6**

Note the use of headers in this web page: Heading 1 for the main title, Heading 2 for titles of major sections, Heading 3 for subtitles. I've set all the headers off in green today as an example.

The tags for headings:

HTML	**As displayed in browser**
** **Heading 1 Example** **	**Heading 1 Example**

Heading 2 Example	**Heading 2 Example**
Heading 3 Example	**Heading 3 Example**

The tags for h4, h5, h6 are similar, although going below h3 generally isn't a good idea.

Line breaks and horizontal rules

All of the tags we've seen so far have opening a closing tags, to signal the beginning and end of formatting. Two exceptions are horizontal rules, <hr>, and line breaks, <br>.

HTML	As displayed in browser
first line second line	first line second line
first line <hr> second line	first line --- second line

Note: Hardcore HTML purists will tell you to format these tags as <hr /> and <br /> to keep the notion of opening and closing tags consistent.

Using them

Generally, HRs (typical to abbreviate elements by their tags like this) are used to divide up content. As you've seen, my philosophy is to use HRs to divide up content at higher levels of hierarchy (e.g. before each H2 in the notes) and use lower-level headings (e.g. H3 in the notes) to divide up information at lower levels of hierarchy. HRs are also handy to separate content from contact info, disclaimers, etc.

Use BRs within other HTML elements, e.g. table cells and list items where a line break is necessary. Break up paragraphs with the paragraph tag -- the whitespace provided fosters readability.

Directory Structure & Links

Directory structure

In a WYSIWYG editor, you can use the GUI to find the paths of links and images. In HTML, you need to refer to the exact *path* of the file.

HTML is designed to work with web servers. Web servers are generally based on Unix operating systems. Hence, we use forward slashes, /, to denote changes in directories. (Microsoft stole the idea but changed it ever so slightly to backslashes, \, so what you see in Windows is slightly different.)

File paths are always relative to the current directory. The notation **..** (two dots) is used to denote "one level up."

Some general forms and examples:

To reference...	General form	Example
a file in the current directory	filename.extension	schedule.htm
a subdirectory of the current directory	directory_name/	images/
the index.html file of a subdirectory of the current directory	directory_name/	cmpsc100/
a file in a subdirectory of the the current directory	directory_name/filename.extension	images/picture.jpg
a file in a subdirectory two levels down	directory_name/directory_name/filename.extension	images/fall/pic.jpg
the index file of a directory one level up (the parent directory)	../	../
a file in the parent directory	../filename.extension	../filename.extension

Problems

Suppose we have a directory structure like this: (filenames are given in italics directly under directory names):

- www
  index.html
  univ_logo.jpg
  - departments
    - **cs**
      - classes
      - programs
        undergrad.html
      - contacts
    - math
      - classes
      - programs
        undergrad.html
      - contacts

- registrar
  - schedules
    - fall_03
      index.html
    - spr_04

Suppose we're currently in the **cs** directory noted in blue and bold. We'd like to express the path of certain files relative to this directory.

1. There's a file called **undergrad.html** in this **programs** subdirectory. What is its path?

2. There's a file called **undergrad.html** in the programs directory of the **math** directory. What is its path?

3. We need to reference the **fall_03** schedule that lives in the registrar's directory. It's found in the file **index.html** of that directory. What is its path?

4. We want to refer to the **univ_logo.jpg** file in the top level **www** directory. What is the path?

5. We want to refer to the **index.html** file in the top level **www** directory. What is the path?

Links

The set of tags for a link has the following syntax:
****Linked Text(what is blue and underlined)****
The path takes the same form as discussed in the last section. The URL can be used for files that aren't in your current directory structure.

Examples

HTML
More information is found in ****yesterday's lecture****
****The schedule from another course****
****Google**** is an example of a search engine.

Lists

Lists

We need tags to signify the beginning and end of the list:

- For a bulleted or **unordered list**, **
** and **
**
- For a numbered or **ordered list**, **
** and **
**

We also need tags to signify the beginning and end of each **list item**: **
** and **
**

Examples

HTML	As displayed in browser

** **
  ** **first** **
  ** **second** **
** **
	• first
	• second

** **
  ** **first** **
  ** **second** **
** **
	1. first
	2. second

** **
  ** **first
  ** **
  ** **sublist first** **
  ** **sublist second** **
  ** **
  ** **
  ** **second** **
** **
	• first
	  1. sublist first
	  2. sublist second
	• second

Some closing comments

- There are a lot of heuristics given here and in prior notes. They are just that, heuristics, i.e. guidelines. There are certainly places where they can be broken, but for your purposes, don't.
- Some of the tags we've looked at here can be given *attributes* that add more specifications. For example, we can add to the paragraph tag to specify alignments.

Text Formatting in HTML

Basic Toggled Formats

First, recall the **** and **** tags for bold and italics...

Bold example

HTML	As displayed in browser
This is a sentence with ****some bold text****.	This is a sentence with **some bold text**.

There's a **** to say "start bold formatting here." The **** says "stop bold formatting here."

Notice that the tags are the same, except that there's a forward slash (/) in the closing tag.

Italics

Similarly, the tags for italics are **** and ****.
Example:

HTML	As displayed in browser
some kind of ****interpreter**** that reads	some kind of *interpreter* that reads

Caution!

- Watch that you have a closing tag to go along with the opening tag.
- Watch that the closing tag has the /.

An Alternate View: strong and em

We know the **** and **** tags for bold and italics, but there's another way to achieve the same visual formatting:

- The **** ... **** tag pair produces bold text.
- The **** ... **** tag pair produces italicized text.

These tag pairs are more strongly encouraged than the **** and **** tags today. The reason is accessibility: screen readers that speak the content of web pages can work with these tags for the visually-impaired, yet they still display the bold and italic text we know in the browser.

Underlining

There is a tag for underlining, but **NEVER, NEVER, NEVER USE UNDERLINED TEXT ON WEB PAGES!!!** Why?

Tags with Attributes

With the exception of the **<a>** tag, all of the tags we've worked with have simply contained the name of the tag between the angle brackets. We now add a slight complication into the mix...
Some tags can have **attributes** that can be set to specify certain properties. The attributes are set in the opening tag. The general form of the syntax is:

<tag-name
attribute_1="argument**" attribute_2="**argument**" ... attribute_n="**argument**">**

Note that the closing tag does **not** include the attributes. It just needs the slash and the name of the tag:

</tag-name>

Text Sizes

Before we write code to format text, we need to understand web font sizes.

How Text is formatted on the Web: Font Sizes

Web font sizes aren't the same as the font sizes in applications intended for print or viewing on one user's computer.
There are seven web font sizes:

Font Size 1	Font Size 2	Font Size 3	Font Size 4	Font Size 5	Font Size 6	Font Size 7

However, if you don't have a reason to set the font size, let it alone and let the browser decide.

How Text is Formatted on the Web: Headers

Recall that HTML provides levels of **headers** for organizing a document.
There are six levels of headers:

Header 1	**Header 2**	**Header 3**	**Header 4**	**Header 5**	**Header 6**

Important Heuristics

- **Don't set specifics about fonts**. Leave that up to the browser!
- **Use headers** rather than changing font size settings. Why?

Fonts

The **<font>** tag needs attributes to be useful. Specifically, it can take any of these attributes:

- size - takes arguments of specific HTML font sizes 1-7 (usu. bad) or relative font sizes like "+1" or "-1" (good)
- color - takes arguments of *hexadecimal* codes for colors (good, but you have to look them up) or color names (okay if they're known by the browser)
- face - takes arguments of font faces (dangerous, assumes all users have the font installed) or comma separated lists of font faces (better)

Example

HTML	As displayed in browser
`This part isn't formatted.`	This part isn't formatted.
`<font size="+1">This part is a bit bigger.</font>`	This part is a bit bigger.
`<font size="-1">This part is a bit smaller.</font>`	This part is a bit smaller.

`<font color="#336600">This part is green.</font>`	This part is green.

`<font face="Arial, Helvetica, sans-serif">`
`  This part is in Arial, Helvetica, or sans-serif.`
`</font>`
	This part is in Arial, Helvetica, or sans-serif.

`<font color="#336600" size="+2">`
`  This part is green and a bit bigger.`
`</font>`
	This part is green and a bit bigger.

`<font color="#000099" size="1">`
`  This part is blue and in size 1 and <b>this is bold</b>.`
`</font>`
	This part is blue and in size 1 and **this is bold**.

Alignment

We learned about the **<p>** tag to create paragraphs. Alignment is a property of a paragraph, so it is set as an attribute. The attribute name is **align**. It can take on one of four values:

- left
- center
- right
- justify

Example

HTML	As displayed in browser
`This part isn't formatted and longer than the rest.`	This part isn't formatted and is longer than the rest.
`<p align="center">This part is centered.</p>`	This part is centered.
`<p align="right">This part is right-aligned...</p>` `<p align="center">...but this part is centered.</p>`	This part is right-aligned... ...but this part is centered.

`<p align="right">`
`  <font color="#336600">This part is green and right-aligned.</font>`
`</p>`
	This part is green and right-aligned.

Notes

- The default alignment is left. If the align attribute is omitted, text is left-aligned.
- The justify alignment is a newer addition to the language and is not supported by all browsers. Do some trials to see which ones *do* support it.
- Note the nesting of tags in the last example.

Attributes and Ordered Lists

Recall that the **<ol>** tag starts an ordered list. Where did this list start counting?

We can add attributes to this tag as well. Here are the two that you may want to use:

- type - takes arguments of "A" or "a" or "i" or "I" for different types of ordered lists
- start - takes arguments in the form of integers in quotation marks to say where to start counting

Notes

- The default type uses Arabic numeral and starts counting at 1.
- Because lists that are "numbered" with letters are essentially the same thing as lists that are truly numbered, we call these lists ordered lists.

Example

HTML	As displayed in browser

`<ol>`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	1. first item
	2. second item
	3. third item

`<ol start="0">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	1. first item
	2. second item
	3. third item

`<ol type="a">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	a. first item
	b. second item
	c. third item

`<ol type="A">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	A. first item
	B. second item
	C. third item

`<ol type="I">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	I. first item
	II. second item
	III. third item

`<ol type="A" start="4">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	D. first item
	E. second item
	F. third item

`<ol type="i" start="7">`
`  <li>first item</li>`
`  <li>second item</li>`
`  <li>third item</li>`
`</ol>`
	vii. first item
	viii. second item
	ix. third item

Observations worth mentioning again

- Notice that blank lines (whitespace) are ignored. It's good style to include whitespace because the HTML needs to be readable.

- Notice that spaces are ignored if there's more than one of them at a time. You could put 20 spaces between words and it's the same as putting one. Tabs mean nothing.
    - There's a way around that called a non-breaking space, but there really isn't any place you'd need to -- or *should* -- use it.
    - Common mistake I've seen: *Do NOT* try to indent paragraphs with a bunch of non-breaking spaces. Paragraphs in HTML were never intended to be indented. It looks sloppy.

Images

Image Types

Images on the web should be in one of the following two file formats:

- GIF (Graphics Interchange Format) - *.gif
- JPEG (Joint Photographic Experts Group) - *.jpg

Prepare your images first. Put them in the same directory where you're making your web page. One standard practice is to have an **images** subdirectory where you store all of your graphics.

Additionally, the PNG format is supported by some, but not all browsers.

A Word on Web-Friendly Filenames

Some guidelines for the names of all files you create for use on the web:

- Restrict filenames to **lowercase** letters, hyphens (better), underscores (worse), and digits. In other words, never use capital letters in spaces.
- **NEVER** use any spaces in filenames. Use hyphens to denote changes in words.
- All files need proper extensions to work.

These conventions come from history and the behavior of web servers. Web servers are case sensitive. Some have problems with uppercase letters. Just be clean and stick to lowercase. Also, spaces can be problematic. Finally, spaces add unnecessary complication to programming that can avoided by smart file-naming.

Image Tag

The set of tags for an image has the following syntax:

<**img src="**path_or_URL**">**

As this tag doesn't require a closing tag, we could also use this form:

<**img src="**path_or_URL**" />**

Attributes of the Image Tag

Like many of the other tags we've studied, the image tag could take any number of attributes. We'll look at these in turn.

Alt text

It's generally a good idea to set the **alternative text** (generally called "alt text") for each image that conveys critical information:

- If the person viewing your page has set his/her browser preferences so that images are not displayed, the browser will display this text in its place.
- If a person is "viewing" your page with a screen reader, the alt. text is read whenever there ought to be images in the page.
- If an image is purely decorative, don't set alt text.

To set the alt text, we use the alt attribute of the img tag:

<img src="path_or_URL" **alt="**description of image**">**

Title Attribute

The title attribute sets text that shows up as a tooltip when the user hovers the mouse over the image. Use this for more information. It can also be a good place to give credit to photographers, sources, etc.

To set the title tool tip text, we use the title attribute of the img tag:

<img src="path_or_URL" **title="**description of image**">**

Image Size

The height and width attributes of the img tag can be used to set the size of images. These attributes are measured in pixels. (Keep in mind that everyone doesn't have the same screen resolution you do, especially on the newer machines we have in our lab.)
General form:

<img src="path_or_URL" **height="**# of pixels**" width="**# of pixels**">**

If the image size is omitted, the image is displayed full size. However, even if you want to display an image full size, the height and width attributes are recommended. This is because including these attributes tells the browser how much space to leave for an image before it loads. Over a slow connection, your page will take on the proper form as it loads.
To find the size of an image in Mac OS X, open it in Preview and go to **Tools > Get Info** or use the shortcut **Apple + I**.

Image Size Issues

It's best to size your images BEFORE you upload them, rather that uploading a large image and scaling it with attributes of the img tag.

Align attribute

Like the paragraph tag, the img tag has an align attribute. This attribute's value is in allowing text to wrap around an image. The two particularly meaningful values are left and right.
Here's a breakdown:

<img src="path_or_URL" **align="**left**">**

inserts the image at the left side of the page and wraps text around it to the right.

<img src="path_or_URL" **align="**right**">**

inserts the image at the right side of the page and wraps text around it to the left

This sometimes does not give you complete control of the situation. When we learn about tables in the next Section, we'll learn how to make sure text and images never overlap in bad ways.
If align is omitted, the image acts like a very large character. If more text fits to its right, it fits. If not, it goes to the next line. This can, of course, be control with tags like <br> and ... .

Attribute Closing Comment

Note that any of these attributes can be joined together. Here's an example:

<img src="my-logo.jpg" height="200" width="300" alt="Company Logo"
align="left">

Text Images and Capturing Graphics

There are times on web pages when we'd like to put text in interesting fonts or achieve special formatting, such as for logos and the like. Since these fonts may not be available on all systems, using them in the HTML is dangerous. Instead, it's better to build the text in the format we want it and save it as an image instead.
There are many professional graphics tools out there to create graphics the right way. Photoshop is perhaps the best one. Unfortunately, it's not installed on our lab computers. Note that iPhoto provides some Photo Editing capabilities.

As for making graphics from text, we'll experiment with this using the tools we do have: the Office software. While it's not particularly ideal, it is possible to create interesting text effects in programs like Word and PowerPoint and take a picture of it to insert on our pages. We call what results a *screen shot*; we'll learn how below.
There are places where using screen shots may be the best way to achieve certain formats on the web. For example, it's not easy and it's sometimes impossible to format mathematical equations that are even slightly interesting in HTML. If we don't want to go to the trouble of learning MathML, a quick way to get a professional-looking equation on a site is to create it in Word using Equation Editor and take a screen shot. There is a drawback: the image version of the text is static, so it cannot be resized in the browser.

Taking Screen Shots

In Mac OS X, there are two commands for taking screen shots:

- **Apple + Shift + 3** takes a screen shot of the entire screen.
- **Apple + Shift + 4** brings up a tool that allows you to select part of the screen to take a picture of. Position the mouse where you want the top left to be and drag it to the desired bottom right position.

Both commands save the result to the desktop in **png** format. Never fear! You can double-click these files to open them in **Preview**. There, you can use the **Save As** command to give them web-friendly, descriptive filenames and convert them to **jpg** format. (Remember to clean up the desktop when you're done.)

In Windows, there are also two commands for taking screen shots:

- **PrintScreen** takes a screen shot of the entire screen.
- **Alt + PrintScreen** takes a screen shot of just the active window.

Both commands store the screen shot on the clipboard. You'd need to paste it into photo editing software to save it.

A Word on Ethics

Files on the web aren't free for the taking; there are ownership issues. **Do not copy and use images from other people's sites without their permission**. There are some sources of images that are in the public domain. For such images, it's a good idea to cite the source. (Alt text is a fine place.) You must be absolutely sure you're not violating any laws or the wishes of whoever created an image before posting it.

Linked Graphics

You may want to turn an image into a link. The process for doing this is logical: simply nest an image tag INSIDE a link tag:

```
<a href="path_or_URL_of_target"><img src="path_or_URL_of_image"></a>
```

With a text link, we put text for users to click on between the <a>... tag pair. Now we want users to click on an image, so we put the image there. It just requires more HTML.
If we wanted to make buttons, we'd use this idea.

More Image Attributes

Borders

You'll notice that when you put <a>... tags around an image, the image gets a blue border around it, somewhat like how a text link gets a blue underline. This is default

behavior, but it can be changed. Similarly, we can put borders around non-linked images. We change these settings using the border attribute:

<img src="path_or_URL_of_image" **border="**number_of_pixels_in_border**"**>

If the border attribute is omitted,

- For a nonlinked graphic, its value is 0 (no border).
- For a linked graphic, a thin blue border appears. (This can be turned off by adding border=0 into the img tag.)

Spacing around an Image

If you use the align attribute to wrap text around an image, you also have control over how close the text gets to the image. The image's **hspace** (horizontal space) and **vspace** (vertical space) tell how much space to put around the image:

- Use the hspace attribute to set how many pixels are between the image and the rest of the page contents on the left and right.
- Use the vspace attribute to set how many pixels are between the image and the rest of the page contents on the top and bottom.

Here's the general form:

<img
src="path_or_URL" **hspace="**number_of_pixels**" vspace="**number_of_pixels**"**>

Here's an example that surrounds **logo.jpg** with 4 pixels on the left and right and 2 pixels on the top and bottom:

<img src="logo.jpg" **hspace="**4**" vspace="**2**"**>

Background Formatting

It's possible to set some of the background properties of your page. These properties include the background color, text color, and the color of links. You can also set a background image. Before I tell you how to this, let me remind you to use background images very carefully. They can go from tasteful to problematic very quickly.

All of these properties are properties of the page, more specifically the body of the page. Therefore, they are all attributes of the body tag. First, here's a general form:

<body **text="**color name/code**" link="**color name/code**" vlink="**color name/code**" bgcolor="**color name/code**" background="**path or URL**"**>

It's possible to include any number of these attributes.

Here are details on each of the attributes:

- **text** sets the default color of all text on your page. You can always override this setting with the font tag's color attribute. By default, text color is black. Use a color name or hex code here.
- **link** sets the default color of all links on your page. By default, this color is blue. Use a color name or hex code here.
- **vlink** sets the default color of all visited links on your page. By default, visited links are purple. Use a color name or hex code here.
- **bgcolor** sets the background color of your page. By default, the background color is white. Use a color name or hex code here.
- **background** sets an image to be the background of your page. This image is tiled. Use the file path of the image here (or a URL).

Tables

Setting Up Basic Tables

The Basics

Tables are essentially another kind of container. Tables consist of **rows**, which run horizontally, and **columns**, which run vertically. Rows and columns intersect at **cells**. We build tables as collections of rows. In turn, we build rows as collections of cells.
We need tags to signify...

- the beginning and end of the table: **<table>** and **</table>**
- the beginning and end of the rows: **<tr>** and **</tr>**
- the beginning and end of the cells within rows ("table data" cells): **<td>** and **</td>**

Example

HTML	As displayed in browser

<table>
  <tr>
  <td>first**</td>**
  <td>second**</td>**
  </tr>
  <tr>
  <td>third**</td>**
  <td>fourth**</td>**
  </tr>
</table>
	first second
	third fourth

A Few Starting Notes

As you see above, the width of cells in tables is determined by their contents by default. (Later we'll see that there's more to the story.)
As you also see above, tables don't come with borders by default. Let's fix this problem by looking at one attribute right now. We can add **border="1"** to the opening <table> tag to display the borders between cells.
Consider the following change:

HTML	As displayed in browser
<table **border="1"**> <tr> <td>first</td> <td>second</td> </tr> <tr> <td>third</td> <td>fourth</td> </tr> </table>	first second third fourth

Properties of Tables and Cells

Before we go on, we must learn some of the language of tables.

Properties of Tables: Cell Properties

These properties apply to the individual cells, but are consistent across the whole table. The unit for all of these settings is the pixel (ranging over nonnegative integers).

- **Border**
  - The thickness of the lines dividing cells.
  - Drawn around the cells themselves indisputably and around the whole table.
  - Use 0 to have no visible borders.
  - Use values >= 1 for visible borders.
- **Cell padding**
  - How much space there is **within the cell** between the text and the border of the cell
  - Text is flush with the borders if cell padding is 0 or omitted
  - Text is padded from the cell walls if cell padding is >= 1
- **Cell spacing**
  - How much space there is **between adjacent cells**
  - Cells are side by side if cell spacing is 0 or omitted, i.e. borders can "overlap"
  - Cells are spaced if cell spacing is >= 1, i.e. there's a visible gap between them

Examples of Cell Padding and Cell Spacing

In all cases, the *border* is 3 pixels. The examples are arranged in a table with *border* 0 as an example of a "borderless" table.
The numbers are extreme on purpose so effects can be seen on the screen in class.

cell padding	cell spacing	example table
0	0	one two three four five six seven eight nine
0	5	one two three four five six seven eight nine
5	0	one two three four five six seven eight nine
5	5	one two three four five six seven eight nine

Properties of Tables: Table Properties

These properties apply to the table as a whole.

- **Alignment**
  - Refers to the horizontal alignment of the whole table on the page, NOT the alignment within cells
  - If "Default" (i.e. omitted), the browser or other properties of the page decide.
  - **Ex**: big table above has center alignment
- **Width**
  - A value can be set
  - Width is unspecified --> content determines width
  - Options of setting the width in pixels or percentage of the browser width
  - Be very careful with specifying widths. Why?
  - **Ex**: all table widths on this page are unspecified

A Word on Cell Size

By default,

- The width of cells is determined by: (width of their content) + (cell padding)
- The height of cells is determined by: (height of their content) + (cell padding)

The table width is the sum of all the cell widths and sum of all cell spacing.

Attributes of <table> for Cells

The **<table>** tag has some attributes that may be set:

- cellpadding
- cellspacing
- border

See above for details on what they mean and how they're measured.

Example

HTML	As displayed in browser
<table cellpadding="5" cellspacing="1" border="1"> **<tr>** **<td>**first**</td>** **<td>**second**</td>** **</tr>** **<tr>** **<td>**third**</td>** **<td>**fourth**</td>** **</tr>** **</table>**	first | second third | fourth

Attributes of <table> for Whole Table

Again, we set attributes of the **<table>** tag:

- align
- width

Valid options:

- align is set to left by default. We can choose center or right. Remember, this is the alignment of the whole table on the page.

- width can either take a number of pixels or a percentage. Note that the percentage tells what percentage of the width of the browser the table takes up, or, if the table is nested within something else, say another table cell, the percentage of the parent element.

Example

HTML	As displayed in browser

`<table cellpadding="5" cellspacing="1" border="1" align="center">`
`<tr>`
`<td>first</td>`
`<td>second</td>`
`</tr>`
`<tr>`
`<td>third</td>`
`<td>fourth</td>`
`</tr>`
`</table>`
	<table cellpadding="5" cellspacing="1" border="1" align="center"><tr><td>first</td><td>second</td></tr><tr><td>third</td><td>fourth</td></tr></table>

`<table cellpadding="5" cellspacing="1" border="1" align="right" width="120">`
`<tr>`
`<td>first</td>`
`<td>second</td>`
`</tr>`
`<tr>`
`<td>third</td>`
`<td>fourth</td>`
`</tr>`
`</table>`
	<table cellpadding="5" cellspacing="1" border="1" align="right" width="120"><tr><td>first</td><td>second</td></tr><tr><td>third</td><td>fourth</td></tr></table>

`<table cellpadding="5" cellspacing="1" border="1" align="right" width="90%">`
`<tr>`
`<td>first</td>`
`<td>second</td>`
`</tr>`
`<tr>`
`<td>third</td>`
`<td>fourth</td>`
`</tr>`
`</table>`
	<table cellpadding="5" cellspacing="1" border="1" align="right" width="90%"><tr><td>first</td><td>second</td></tr><tr><td>third</td><td>fourth</td></tr></table>

Setting Properties of Individual Cells

We can alter some properties of individual cells by setting attributes of the <td> tag:

- align - sets the horizontal alignment of the cell's contents: left, center, or right (left is default)
- valign - sets the horizontal alignment of the cell's contents: top, center, or bottom (center is default)
- width - sets the width of a cell; arguments in pixels or percentages just like the settings for the whole table

- height - sets the height of a cell; arguments in pixels

Two notes:

- Remember that, by default, the contents of a cell plus cell padding determine its size. If you don't have a good reason to change the size, I'd recommending letting it alone.
- The width of a cell affects how an entire column is rendered. If your widths aren't consistent, things can get tricky.

Rowspan and Colspan

The <td> tag has two attributes, rowspan and colspan, that allow you to make cells cross multiple rows and columns. Essentially, the idea is that cells could be merged together into one.

Both attributes take the number of rows or columns to span as the argument.

It's important to note that when a cell spans multiple columns, there are fewer cells in that row than in other rows. HTML simply fills in cells in a row until the row is closed. If there are too many cells defined in a row, they still show up, but the table takes on a deformed look. I recommend that you try putting too many cells in a row when you're first experimenting with colspan in lab to see what this looks like (so you know how to spot an error to debug it).

Here are some examples:

HTML	As displayed in browser
<table border="1"> <tr> <td **colspan="2"**>first</td> <td>second</td> </tr> <tr> <td>third</td> <td>fourth</td> </tr> </table>	*Notice the error here:* <table border="1"><tr><td colspan="2">first</td><td>second</td></tr><tr><td>third</td><td>fourth</td></tr></table>
<table border="1"> <tr> <td **colspan="2"**>first</td> </tr> <tr> <td>third</td> <td>fourth</td> </tr> </table>	<table border="1"><tr><td colspan="2">first</td></tr><tr><td>third</td><td>fourth</td></tr></table>
<table border="1"> <tr> <td>first</td> <td **rowspan="2"**>second</td> </tr> <tr> <td>third</td>	<table border="1"><tr><td>first</td><td rowspan="2">second</td></tr><tr><td>third</td></tr></table>

</tr> </table>	

Advanced Uses of Tables

There are many uses of tables beyond what you might think. Here are just a few ways you can use tables in more advanced ways:

- Structuring pages: vertical navigation bar in one column, contents in a second (1 row x 2 columns)
- Formatting
- Achieving multiple alignments on one line: left in one column, right in another
- Positioning pictures and keeping contents away from them
- Nesting tables within tables

The <th> Tag

In all of these examples, we've worked with building tables from <tr> and <td> tags. Of course, we need to format header rows and columns in bold or italic print to distinguish them from the actual data cells. We could do this by nesting formatting tags within the cells.

We started this way to keep things simple. There is also a <th> tag, which stands for "table header." We can use <th> in lieu of <td> when we want to mark cells as part of a header row or column. By default, the contents of these cells are formatted with bold text and centered, so applying the formatting is much simpler.

Consider the following two examples:

HTML	As displayed in browser
<table cellpadding="5" cellspacing="1" border="1"> <tr> <td> </td> <td align="center"> column 1 header </td> <td align="center">column 2 header </td> </tr> <tr> <td align="center">row 1 header </td> <td>first</td> <td>second</td> </tr> <tr> <td align="center">row 2 header </td> <td>third</td> <td>fourth</td> </tr> </table>	<table cellpadding="5" cellspacing="1" border="1"><tr><td></td><td>column 1 header</td><td>column 2 header</td></tr><tr><td>row 1 header</td><td>first</td><td>second</td></tr><tr><td>row 2 header</td><td>third</td><td>fourth</td></tr></table>

```
<table cellpadding="5" cellspacing="1"
border="1">
   <tr>
     <th>&amp;nbsp;</th>
     <th>column 1 header</th>
     <th>column 2 header</th>
   </tr>

   <tr>
     <th>row 1 header</th>
     <td>first</td>
     <td>second</td>
   </tr>

   <tr>
     <th>row 2 header</th>
     <td>third</td>
     <td>fourth</td>
  </tr>
</table>
```

	column 1 header	**column 2 header**
row 1 header	first	second
row 2 header	third	fourth

HTML – A Quick Reference

What is HTML?
Hypertext Markup Language (HTML) is a syntax used to format a text document on the web. These documents are interpreted by web browsers such as Internet Explorer and Netscape Navigator.
HTML can be created as standard ASCII text with "tags" included to pass on extra information about character formatting and page layout to a web browser. The fact that HTML is, in essence, ASCII text is what makes it so universally compatible. This fact also makes it easy to edit: almost all computers are equipped with a text editor that can be used to edit HTML.

How can I create an HTML page?
HTML pages can be created in a number of ways. An HTML page is essentially a text document. You can create one in the simplest of text editors such as Microsoft Notepad in Windows, SimpleText in Mac OS, or Pico in Unix. Using these tools, you will need to edit the HTML code and insert HTML tags where necessary. You can also create pages using WYSIWYG (What You See Is What You Get) editors which do most of the work of coding for you. With WYSIWYG editors, such as Macromedia Dreamweaver and Adobe GoLive, you can type in a page as you would in a word processor, and the software adds formatting tags where necessary. You can then look into the code for fine-tuning.
Note: There are some WYSIWYG editors that do not create good HTML code. For instance, Microsoft FrontPage and particularly Microsoft Word will add extra tags that can make an HTML document quite large. Larger documents take longer to download. While you may not notice this much while on campus at NIH, modem users at home can be inconvenienced by long waits.

To save a file as HTML, open the text you wish to edit or type it into Notepad, and choose "File...Save As..." from the menu bar. Under "File name" give your file a name and change its extension from ".txt" to ".html." Under "Save as type" switch to "All Files" then click "OK." Notepad will save your file as ASCII text, and the ".html" or ".htm" extension will allow your browser to recognize it as an HTML file. (Hint - When naming an HTML file, it is a good idea to use a name without any spaces or uppercase letters. So, if you wanted to name a file Test page, some options would be testpage or test_page.)

To open your file in Netscape, choose "File...Open Page..." from Netscape's menu bar, and click "Choose File." Navigate to the drive and directory in which you saved your HTML, and double click on the file.

To edit your HTML, go back to Notepad, make your changes, and choose "File...Save" from the menu bar. Then go back to Netscape, and click the "Reload" button to bring in your latest changes.

What are Tags?
Tags are what we use to structure an HTML page. Tags start with a '<', then the command, and end with a '>'. For example, the center tag is ' '. To stop centering something, we need an ending, or closing tag. Closing tags look exactly like

opening tags, except after the first '<' there is a '/'. In other words, the closing tag for center is ' '.

HTML Structure

An HTML document has a definite structure that must be specified to the browser. The HTML's beginning and end must be defined, as well as the document's HEAD (which contains information for the browser that does not appear in the browser's main window) and its BODY (which contains the text that will appear in the browser's main window). The use and order of tags that define the HTML structure are described below.

<html>	Marks the beginning of your HTML
<head>	Begins the heading section of an HTML document
<title> ... </title>	Gives an HTML document a title that appears on the browser menu bar, also will appear on search engines or bookmarks referencing your site (must appear between the <HEAD> ... </HEAD> tags; should be straight text, no tags
</head>	Defines the end of the heading
<body>	Defines the body of an HTML document (text contained within the <BODY> … </BODY> tags appears in the main browser window). Can be used with "BGCOLOR", "TEXT", "LINK", and "VLINK" attributes
</html>	Defines the end of your HTML document

Character Formatting and Page Layout

The following tags are used for character formatting (bold, italicized or underlined text, for instance) and page layout (where and in what context that text appears on the page). These tags are used in the body of an HTML document only.

<!--...-->	*Comment.* This is a note for you. It will not be visible on the web page.
...	*Heading* tag. <H1> through <H6> are valid. Can be used with the "ALIGN" attribute.
...	Sets a *paragraph* apart from other text. Adds a line break after. </P> is optional. Can be used with the "ALIGN" attribute.
 	Line break (new line). Can be used with the "CLEAR" attribute.
<hr>	*Horizontal rule.* Can be used with "SIZE", "WIDTH" and "NOSHADE" attributes.
...	Defines the beginning and end of an *ordered list* (numbered).
...	*Unordered list* (bulleted).
	List Item. Must appear before each item in any of the above lists to set it apart from other items.
...	*Centers* any item or group of items

...	*Emphasized* text (usually italic).
...	*Strong emphasis* (usually bold).
<code> ... </code>	A sample of *code* (usually courier font).
 ... 	Changes the appearance of the text in your page. Can be used with "SIZE", "COLOR" and "FACE" attributes.
<table> ... </table>	Creates *Table.* Can be used with "BORDER", "ALIGN", and "WIDTH" attributes.
<tr> ... </tr>	Table *row*
<th> ... </th>	Table *header.* Can be used with "ALIGN", "VALIGN", "COLSPAN", "ROWSPAN", and "WIDTH" attributes.
<td> ... </td>	Table *data.* Can be used with "ALIGN", "VALIGN", "COLSPAN", "ROWSPAN", and "WIDTH" attributes.

Links

<A HREF="http://some.web.server/Document.html"> ... </A>

Creates a hypertext link to another page.

<A HREF="http://some.web.server/Document.html#AnchorName"> ... </A>

Creates a link to an anchor in another web page.

<A HREF="AnchorName"> ... </A>

Creates an anchor within a document that can be linked to.

<IMG SRC="filename.ext">

Inserts a graphic into the web page.

"SRC" is a required attribute. "HEIGHT", "WIDTH", "ALT", "BORDER" and "ALIGN" are optional attributes.

Character Entities (escape sequences)

Some characters that have special uses or meanings for browsers may not always appear as you intend in your browser. To get a browser to display a character without applying that character's special use, "escape sequences" are used. Below are a few frequently used sequences.

Character	Numeric Entity	Character Entity	Description
"	"	"	Double Quotation Mark
#	#		Number Sign
%	%	&	Ampersand
<	<	<	Less Than
>	>	>	Greater Than
			Blank Space