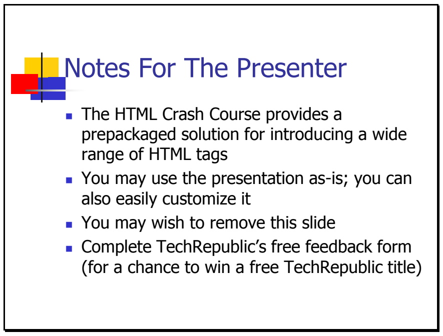
Slide 1



Notes For The Presenter

- The HTML Crash Course provides a prepackaged solution for introducing a wide range of HTML tags
- You may use the presentation as-is; you can also easily customize it
- You may wish to remove this slide
- Complete TechRepublic's free feedback form (for a chance to win a free TechRepublic title)

Notes For The Presenter

The HTML Crash Course presentation provides a prepackaged solution for teaching attendees the basics of HTML code by focusing on some of the most common HTML tags and showing how they are used. The HTML Crash Course Cheat Sheet is a hand out for attendees that summarizes the tags covered in the presentation and can be used as reference material.
The HTML Crash Course presentation assumes:
The attendees have experience surfing the Web with a browser and are familiar with the fact that documents on the Web are formatted with HTML, but have little or no direct HTML coding experience.
The attendees are motivated to learn how HTML works and will use the information in this presentation as a stepping stone for further study and experimentation
You may wish to remove this slide prior to presenting your Lunch and Learn session. Pre-print the speaker's notes (Notes Pages view) for the presenter. Pre-print copies of each slide, as well as a copy of the HTML Crash Course Cheat Sheet, for distribution to each attendee.
Once you've reviewed the HTML Crash Course presentation, please take a moment to complete TechRepublic's short feedback form. Once a month, TechRepublic will randomly award one respondent a free selection from TechRepublic's extensive books, CDs, laminated charts and premium electronic downloads library.

Thank you for trusting TechRepublic to help meet your organization's HTML training needs.

Slide 2

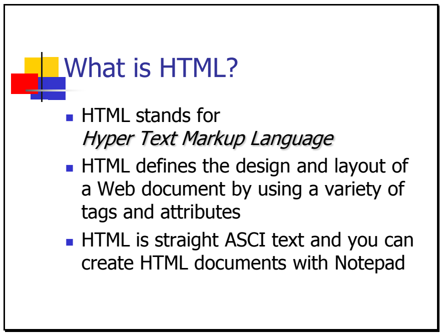
HTML Crash Course

Understanding HTML Tags

As you may know, there are a number of readily available HTML editors, both commercial and shareware, that provide you with a WYSIWYG (What You See Is What You Get) interface for creating HTML documents. As such, it’s very possible that you could create a Web page without ever having to learn about HTML tags. However, having a solid understanding of what HTML tags are and how they work will come in handy should you have to troubleshoot formatting problems with a Web page that you’re creating.

In this presentation, we’ll take a well-rounded look at some common HTML tags. However, keep in mind that we’ll just scratch the surface here but will build a foundation on which you can delve deeper into the details of HTML tags.

Slide 3



What is HTML?

- HTML stands for *Hyper Text Markup Language*
- HTML defines the design and layout of a Web document by using a variety of tags and attributes
- HTML is straight ASCI text and you can create HTML documents with Notepad

HTML stands for *Hyper Text Markup Language* and is essentially the programming language used for creating documents on the World Wide Web. More specifically, HTML defines the structure and layout of a Web document by using a variety of tags and attributes. These tags and attributes range from the basic to very complex and are used to define a number of elements in a document including how text is formatted, how and where pictures are to appear, how links work, and much more.

HTML is straight ASCI text and you can easily create HTML documents with a simple text editor such as Notepad.

Slide 4

What are HTML Tags?

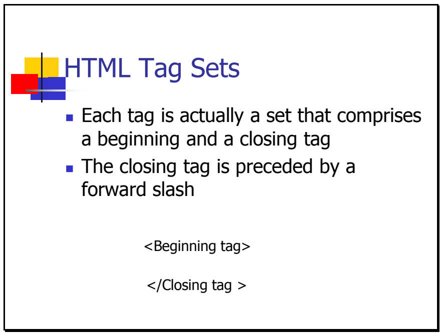
- HTML tags are essentially commands that Web browsers use to determine how to display a document
- HTML tags are enclosed in brackets

<HTML tag>

HTML tags are essentially commands that Web browsers, such as Internet Explorer, Opera, or Firefox, and email clients, such as Outlook or Eudora, use to properly display a document on the Internet or received via email. Like all computer commands, HTML tags have a set of rules, or a syntax, that you must use in order for the tag to function properly.

All HTML tags are enclosed in brackets, which makes it easy to distinguish the tags from the text in an HTML document.

Slide 5



HTML Tag Sets

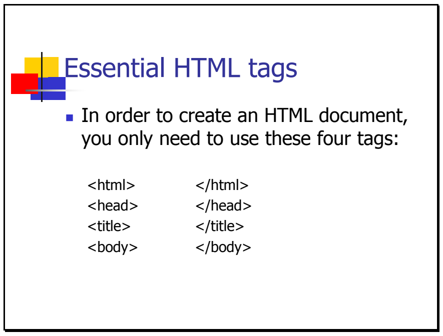
- Each tag is actually a set that comprises a beginning and a closing tag
- The closing tag is preceded by a forward slash

<Beginning tag>

</Closing tag >

To begin with, most of the common HTML tags are actually paired sets that consist of a beginning and a closing tag. The opening and closing tags are essentially the same except that the closing tag is preceded by a forward slash. Anything that falls in between the beginning and the closing tags takes on the properties, or attributes, specified by the tag.

Slide 6



Essential HTML tags

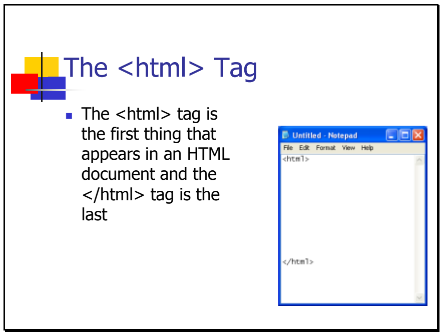
- In order to create an HTML document, you only need to use these four tags:

<html>	</html>
<head>	</head>
<title>	</title>
<body>	</body>

In order to create a very basic HTML document, you only need to use four HTML tags. These tags are: <html>, <head>, <title>, and <body>. Each one is accompanied by a matching closing tag as shown here.

Let’s take a closer look at each one of these tags.

Slide 7



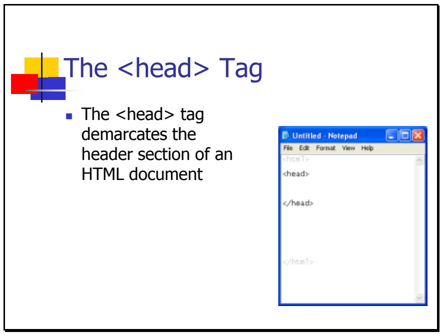
The <html> Tag

- The <html> tag is the first thing that appears in an HTML document and the </html> tag is the last

```
<html>

</html>
```

The <html> tag is the very first tag that appears in an HTML document and specifies that anything that falls between the opening and closing tag is considered the HTML document. As you can see in this example, the closing tag is the very last tag in an HTML document.

Slide 8



The <head> Tag

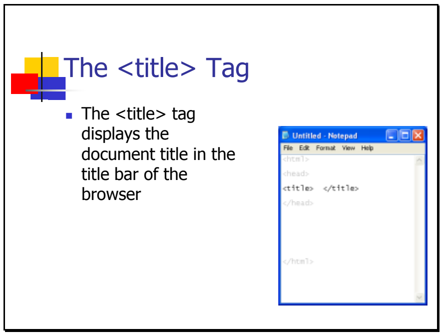
- The <head> tag demarcates the header section of an HTML document

```
<html>
<head>

</head>



</html>
```

The <head> tag is the second tag that appears in a HTML document and specifies that anything that falls between the opening and closing tags is considered part of the header. The header typically contains information used for advanced HTML operations and the majority of this information isn’t displayed by the Web browser.

We’ll cover some of the advanced information that appears in the header later in this presentation.

Slide 9



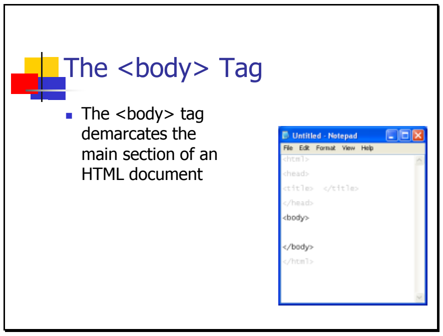
The <title> Tag

- The <title> tag displays the document title in the title bar of the browser

```
<html>
<head>
<title>  </title>
</head>



</html>
```

The <title> tag is the third tag that appears in a HTML document and it actually appears inside the header. The text that appears in the <title> tag represents the title of the HTML document and will be displayed in the title bar of the browser.

Slide 10



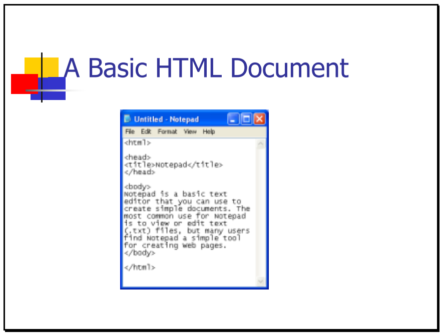
The <body> Tag

- The <body> tag demarcates the main section of an HTML document

```
<html>
<head>
<title>  </title>
</head>
<body>

</body>
</html>
```

The <body> tag is the fourth tag that appears in a HTML document and is used to demarcate the body section. Anything that falls between the opening and closing <body> tag is considered to be the main portion of the HTML document.

Slide 11

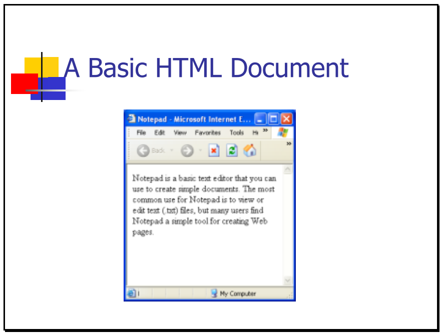


A Basic HTML Document

```
<html>

<head>
<title>Notepad</title>
</head>

<body>
Notepad is a basic text
editor that you can use to
create simple documents. The
most common use for Notepad
is to view or edit text
(.txt) files, but many users
find Notepad a simple tool
for creating web pages.
</body>

</html>
```

Using these four tags as a framework and some text on a particular topic is all that is needed to create a basic HTML document, as shown in this example.

Slide 12

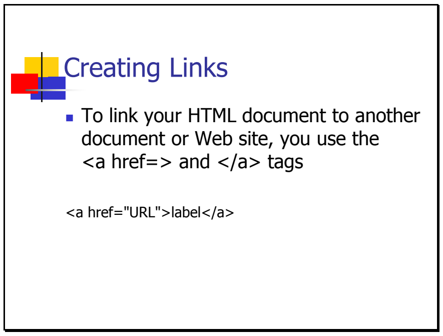

Notepad - Microsoft Internet E...
File Edit View Favorites Tools He
Back
My Computer

A Basic HTML Document

Notepad is a basic text editor that you can use to create simple documents. The most common use for Notepad is to view or edit text (.txt) files, but many users find Notepad a simple tool for creating Web pages.

Just save the file with an .htm extension and open it up in your Web browser. This example shows how the basic HTML document appears in a Web browser.

Slide 13



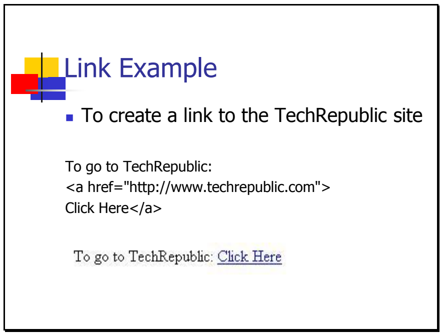
Creating Links

- To link your HTML document to another document or Web site, you use the <a href=> and </a> tags

<a href="URL">label</a>

As you create HTML documents, chances are that you'll want to set up links to other HTML documents or Web sites. In order to specify a link, you use the <a href=> tag. This special tag actually contains the address, or URL (Uniform Resource Locator), of the destination as well as a label that will represent the link in HTML your document. The closing tag is </a>.

Slide 14



Link Example

- To create a link to the TechRepublic site

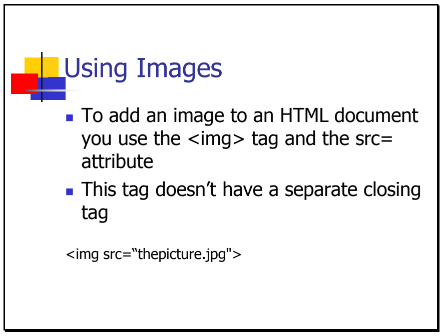
To go to TechRepublic:
<a href="http://www.techrepublic.com">
Click Here</a>

To go to TechRepublic: Click Here

For example, to create a link to the TechRepublic site, you add the URL to the <a href=> tag, follow it with some text that will become the actual line, and cap it off with the closing tag </a> as shown here.

The result of this HTML coding is appears right below the example and shows how the result would look in a browser window.

Slide 15



Using Images

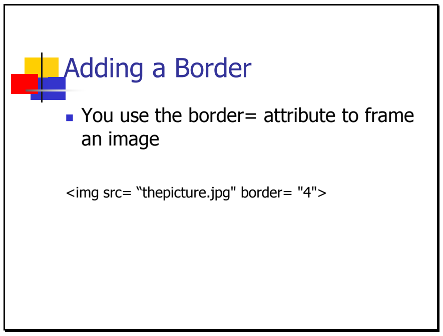
- To add an image to an HTML document you use the <img> tag and the src= attribute
- This tag doesn’t have a separate closing tag

<img src=“thepicture.jpg">

Another common element that you might want to add to an HTML document is an image. To do so, you use the <img> tag along with the src= attribute. This tag and attribute combination specify the name and location of the image file that you want to display in your HTML document. Note that this tag doesn’t have a separate closing tag and that the path and name of the image file must be enclosed in quotes.

There are several other attributes that you can use to customize the display of the image.

Slide 16



Adding a Border

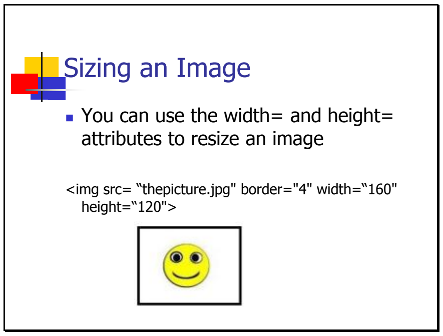
- You use the border= attribute to frame an image

<img src= “thepicture.jpg" border= "4">

You may want to place a border around the image to make it stand out. To do so you use the border= attribute along with a number that specifies the thickness of the border and add it to the <img src=> tag.

Note that the border= attribute is embedded in the<img src=> tag, directly follows the image file name, and that the thickness number must be enclosed in quotes.

Slide 17



Sizing an Image

- You can use the width= and height= attributes to resize an image

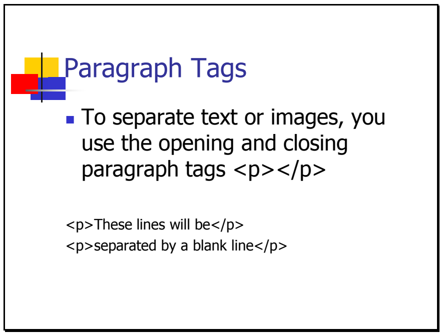
<img src= “thepicture.jpg" border="4" width=“160" height=“120">

If the image is too large, you can resize it by using the height= and width= attributes along with numbers that specify the desired height and width in pixels.

Again, you’ll see that the height= and width= attributes are embedded in the<img src=> tag and that the pixel numbers are enclosed in quotes.

The example image show the effects using the border=, height=, and width= attributes.

Slide 18



Paragraph Tags

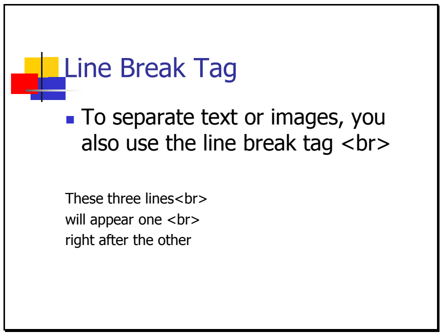
- To separate text or images, you use the opening and closing paragraph tags <p></p>

```
<p>These lines will be</p>
<p>separated by a blank line</p>
```

When you use text or images, you’ll want to separate these elements on the page so they don’t all run together. To do so, you enclose the text or image in opening and closing paragraph tags: <p></p> Doing so will add a blank line between the elements.

This example shows how you use the paragraph tags.

Slide 19



Line Break Tag

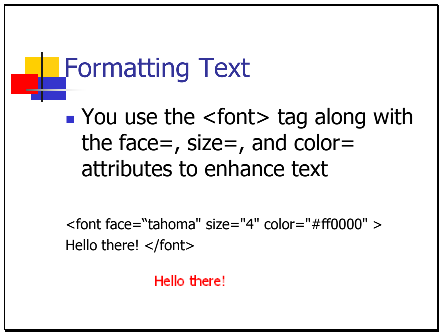
- To separate text or images, you also use the line break tag <br>

```
These three lines<br>
will appear one <br>
right after the other
```

If you want to separate elements but don’t want a blank line between the elements, you can use the line break tag <br>, which doesn’t have a closing tag.

This example shows how you use the line break tag.

Slide 20



Formatting Text

- You use the <font> tag along with the face=, size=, and color= attributes to enhance text

```
<font face=“tahoma" size="4" color="#ff0000" >
Hello there! </font>
```

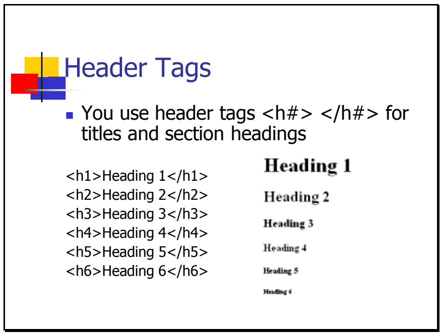
Hello there!

In order to enhance text, you use the <font> tag along with any one of several attributes. The most common attributes that you’ll use with the <font> tag are face=, size=, and color=. Here, the face= attribute is followed by a font name, the size= attribute is followed by a number in the range of 1 thru 7, and the color= attribute is followed by a color name or a hexadecimal color code.

Note that all three attributes are embedded in the opening tag, followed by the text to be formatted, and the closing tag </font> as this example shows.

As you use the <font> tag there are several things you need to keep in mind. First, when you specify a font by using the face= attribute, if the computer displaying the page doesn't have that particular font installed, the page will use whatever font is defined as the default font in the browser. Second, the default value for the size= attribute is 3, 1 is the smallest and 7 is the largest value you can use.

Slide 21



Header Tags

- You use header tags <h#> </h#> for titles and section headings

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Heading 1

Heading 2

Heading 3

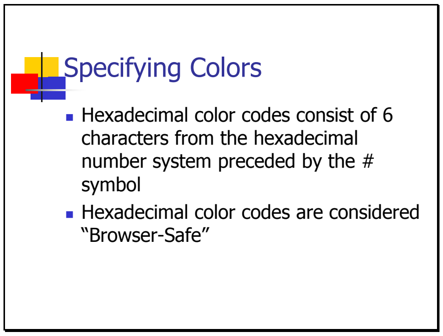
Heading 4

Heading 5

Heading 6

If you need larger sized text, say for titles and section headings, you'll need to use the header tags <h#> </h#>. As you can see in this example, the header tags use a number to specify the size of the heading text; with 1 being the largest and 6 being the smallest.

Slide 22



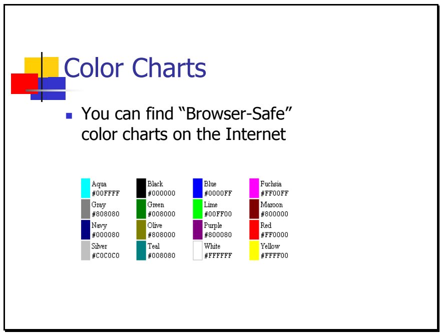
Specifying Colors

- Hexadecimal color codes consist of 6 characters from the hexadecimal number system preceded by the # symbol
- Hexadecimal color codes are considered "Browser-Safe"

You can specify colors for text or the background of the page using a hexadecimal color code. Hexadecimal color codes consist of various combinations of 6 characters from the hexadecimal number system (the numbers 0 to 9 and the letters a to f) preceded by the # symbol.

While you can indeed specify colors using color names, such as red or salmon, using the hexadecimal color codes will provide you greater range of possibilities. There are only 140 color names while there are 216 hexadecimal color codes. Another benefit of using hexadecimal color codes is that they're considered "Browser-Safe" meaning that they will display the correct color in all standard browsers.

Slide 23



Color Charts

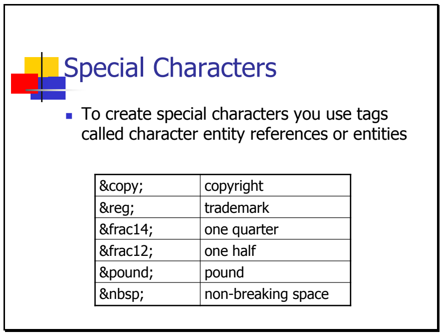
- You can find "Browser-Safe" color charts on the Internet

Aqua #00FFFF	Black #000000	Blue #0000FF	Fuchsia #FF00FF
Gray #808080	Green #008000	Lime #00FF00	Maroon #800000
Navy #000080	Olive #808000	Purple #800080	Red #FF0000
Silver #C0C0C0	Teal #008080	White #FFFFFF	Yellow #FFFF00

You can use your favorite search engine to track down "Browser-Safe" color charts like this one that shows the basic 16 colors.

You'll find a more comprehensive color chart in the handout.

Slide 24



Special Characters

- To create special characters you use tags called character entity references or entities

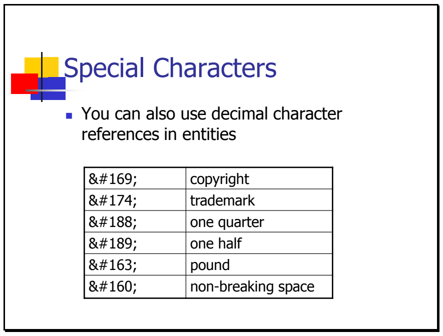
&copy;	copyright
&reg;	trademark
&frac14;	one quarter
&frac12;	one half
&pound;	pound
&nbsp;	non-breaking space

There are a number of characters that do not appear on your keyboard that you may want to use in an HTML document. To do so, you use special tags known as *character entity references*, also called *entities*. Entities begin with an ampersand (&) and finish with a semicolon (;). In between these two symbols you'll use a *shorthand name*.

For example, to create the copyright symbol, you'd use the entity: &copy;

Keep in mind that these shorthand names must be in lowercase, so do not use uppercase characters for entities.

Slide 25



Special Characters

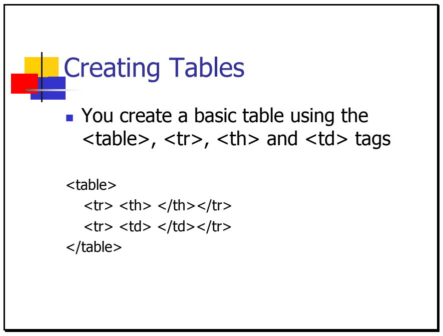
- You can also use decimal character references in entities

&#169;	copyright
&#174;	trademark
&#188;	one quarter
&#189;	one half
&#163;	pound
&#160;	non-breaking space

In addition to shorthand names for entities you can also use decimal character references.

To create the copyright symbol, you'd use the entity: &#169;

Slide 26



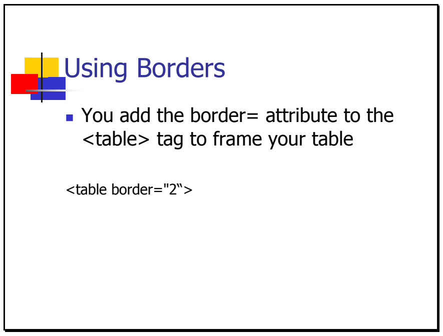
Creating Tables

- You create a basic table using the <table>, <tr>, <th> and <td> tags

```
<table>
   <tr> <th> </th></tr>
   <tr> <td> </td></tr>
</table>
```

Often you'll find that the information that you want to include in your HTML document would be more easily readable if it were presented in a table format. To create a basic table in HTML, you'll use the <table> tag along with the <tr>, <th> and <td> tags, which are used to establish rows, headers and columns respectively. The closing tags are </table>, </tr>, </th>, and </td>

Slide 27



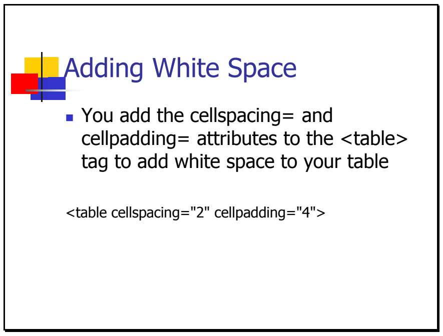
Using Borders

- You add the border= attribute to the <table> tag to frame your table

```
<table border="2“>
```

There are also a group of attributes that you can use to modify the appearance of a table. The most common of these attributes is border=, which is followed by a number that specifies the width of the border.

Slide 28



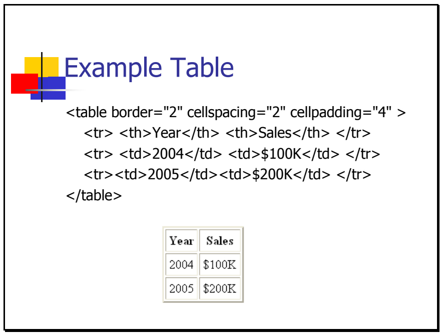
Adding White Space

- You add the cellspacing= and cellpadding= attributes to the <table> tag to add white space to your table

```
<table cellspacing="2" cellpadding="4">
```

Other common table attributes are cellspacing=, which sets the space between the cells, and cellpadding=, which sets the amount of space between the data in the cell and the edges of the cell. Both the cellspacing= and cellpadding= attributes are followed by numbers that specifies the width of the space.

Slide 29



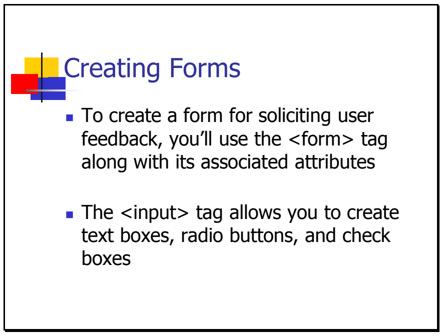
Example Table

```
<table border="2" cellspacing="2" cellpadding="4" >
    <tr> <th>Year</th> <th>Sales</th> </tr>
    <tr> <td>2004</td> <td>$100K</td> </tr>
    <tr><td>2005</td><td>$200K</td> </tr>
</table>
```

Year	Sales
2004	$100K
2005	$200K

This code example on the top will create the table shown below. As you can see, the border= attribute create a nice frame, the header tags <th> make the column headers bold and the cellspacing= and cellpadding= attributes actually center the text in each cell of the table.

Slide 30

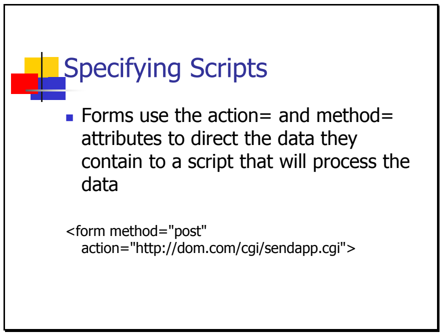


Creating Forms

- To create a form for soliciting user feedback, you'll use the <form> tag along with its associated attributes
- The <input> tag allows you to create text boxes, radio buttons, and check boxes

When you want to get input from the person reading your HTML document, you'll use the <form> tag and the associated attributes. You use the <input> tag to create a dialog box-like controls such as text boxes, radio buttons, check boxes, and action buttons. The closing tag is </form>

Slide 31



Specifying Scripts

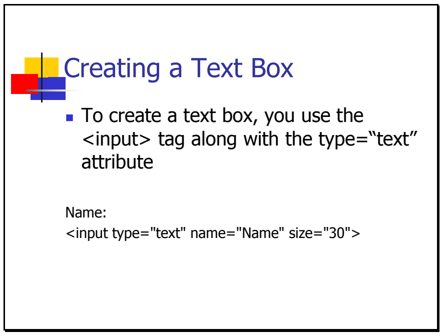
- Forms use the action= and method= attributes to direct the data they contain to a script that will process the data

```
<form method="post"
   action="http://dom.com/cgi/sendapp.cgi">
```

When you create a form to solicit user feedback, you need to have a way to retrieve the information. To do so you will use the action= and method= attributes. In this case, the action= attribute directs the form to Web address, or URL, containing a script that will handle the information contained in the form and the method= attribute instructs the form on how to send the information to the script. The most common method= attribute is post and the action= attribute can point to a CGI (Common Gateway Interface), a FrontPage extensions, or an ASP (Active Server Page) script. The script can compile the information into a report or send the information in individual email messages.

Of course, these scripting technologies are beyond the scope of this presentation; however, chances are that if you're asked to create a form, you'll be directed to use a preexisting script and will have to create your form accordingly.

Slide 32



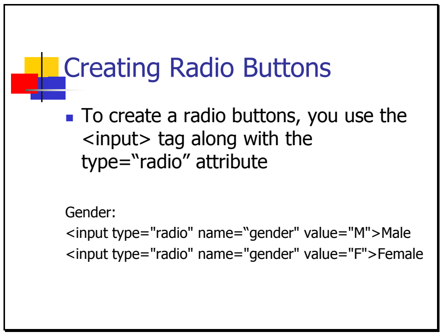
Creating a Text Box

- To create a text box, you use the <input> tag along with the type=“text” attribute

Name:
<input type="text" name="Name" size="30">

To create a text box, you use the <input> tag along with the type=”text” attribute. You also use the name= and size= attributes. The name= attribute sets up a reference name that identifies the contents of the text box in the report. The size= attribute is followed by a number that specifies the length of the text box. You’ll precede the <input> tag with the text that you want to appear on the page as a prompt.

Slide 33



Creating Radio Buttons

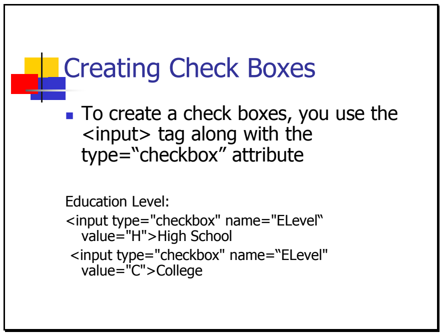
- To create a radio buttons, you use the <input> tag along with the type=“radio” attribute

Gender:
<input type="radio" name=“gender" value="M">Male
<input type="radio" name="gender" value="F">Female

To create a radio button, you use the <input> tag along with the type=“radio” attribute. You also use the name= and value= attributes. The name= attribute sets up a reference name that identifies the selected radio button in the report. The value= attribute sets up a reference name that identifies a particular button. You need the value attribute because when you use radio buttons, you’ll be using more than one button and will need to be able to identify which button was selected.

You precede the <input> tag with the text that you want to appear on the page as a prompt. You follow the <input> tag with the text that you want to identify the button.

Slide 34



Creating Check Boxes

- To create a check boxes, you use the <input> tag along with the type=‟checkbox” attribute

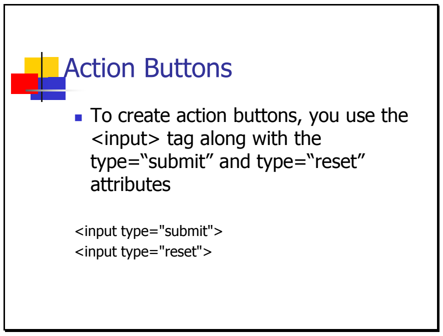
Education Level:

```
<input type="checkbox" name="ELevel‟
   value="H">High School
 <input type="checkbox" name=‟ELevel"
   value="C">College
```

To create a check box, you use the <input> tag along with the type=“checkbox” attribute. You also use the name= and value= attributes. The name= attribute sets up a reference name that identifies the selected check boxes in the report. The value= attribute sets up a reference name that identifies a particular check box. You need the value attribute because when you use check boxes, you’ll be using more than one check box and will need to be able to identify which check boxes were selected.

You precede the <input> tag with the text that you want to appear on the page as a prompt. You follow the <input> tag with the text that you want to identify the check box.

Slide 35



Action Buttons

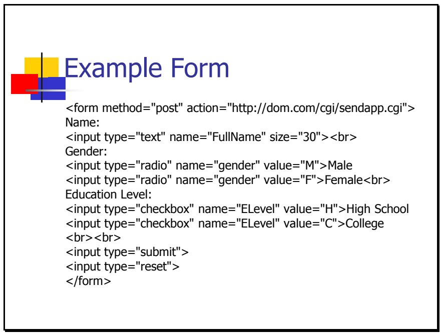
- To create action buttons, you use the <input> tag along with the type=‟submit” and type=‟reset” attributes

```
<input type="submit">
<input type="reset">
```

Once a user has filled in the form, they’ll click one of the action buttons titled Submit or Reset. To create these buttons, you use the <input> tag along with the type=“submit” and type=“reset” attributes.

The Submit button initiates the sending of the data in the form to the script that is specified by the action= attribute while the Reset button simply clears the form.

Slide 36

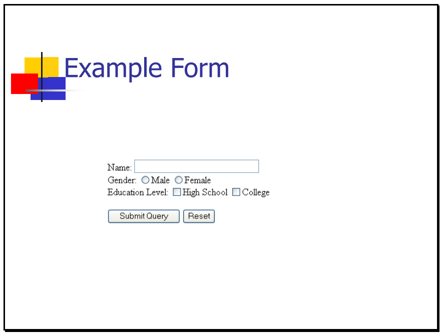


Example Form

```
<form method="post" action="http://dom.com/cgi/sendapp.cgi">
Name:
<input type="text" name="FullName" size="30"><br>
Gender:
<input type="radio" name="gender" value="M">Male
<input type="radio" name="gender" value="F">Female<br>
Education Level:
<input type="checkbox" name="ELevel" value="H">High School
<input type="checkbox" name="ELevel" value="C">College
<br><br>
<input type="submit">
<input type="reset">
</form>
```

This HTML code shows how all the previously discussed elements are used together to create a form.

Slide 37



Example Form

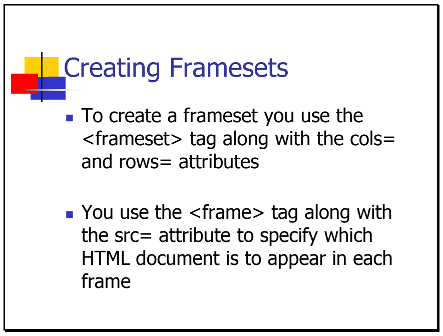
Name:

Gender: ○ Male ○ Female

Education Level: □ High School □ College

This example shows how the form will appear in a browser.

Slide 38



Creating Framesets

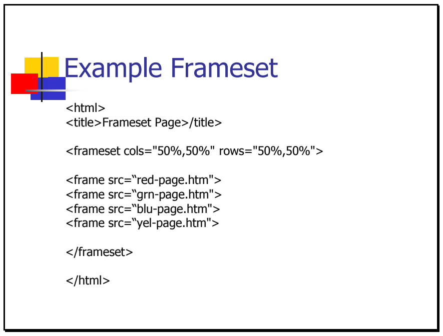
- To create a frameset you use the <frameset> tag along with the cols= and rows= attributes
- You use the <frame> tag along with the src= attribute to specify which HTML document is to appear in each frame

A frameset is essentially a template that you create in order to combine two or more HTML documents on one page. You typically create a frameset when you have an element that remains the same on several pages while the main content changes on each page. For example, you may have a menu that is the same on every page. If so, you could put the menu on a separate page and include that page in a frameset.

To create a frameset you use the opening and closing tags <frameset> </frameset>. To divide up a page with the <frameset> tag you use the cols= and rows= attributes followed by a percentage values.

You then use the <frame> tag along with the src= attribute followed by a name that specifies which HTML document is to appear in each frame

Slide 39



Example Frameset

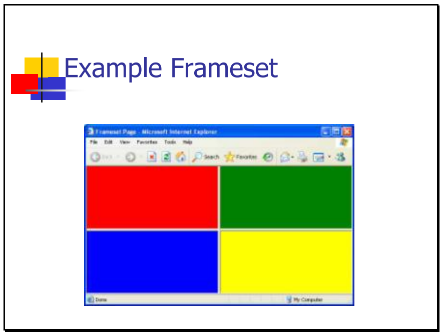
```
<html>
<title>Frameset Page>/title>

<frameset cols="50%,50%" rows="50%,50%">

<frame src="red-page.htm">
<frame src="grn-page.htm">
<frame src="blu-page.htm">
<frame src="yel-page.htm">

</frameset>

</html>
```

Here you see the code for an example frameset that use the cols= and rows= attributes to divide the page up into four sections by specifying two columns and two rows, each at 50 percent.

The four <frame> tags along with src= attributes specifies the name of each HTML document that will appear in each of the four quarters of the page.

As you can see by the <html> tags and the beginning and end of this example, a frameset is actually contained in a separate HTML document.

Slide 40

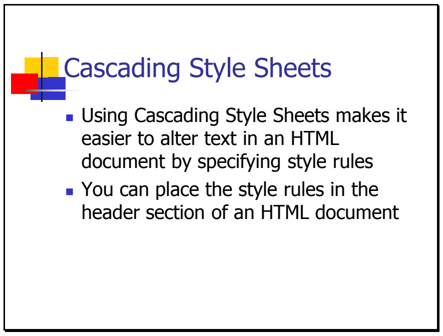


Example Frameset

Frameset Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Search Favorites
Done
My Computer

Here is what the resulting page looks like. As you can see, by default each frame is separated by a border.

Slide 41



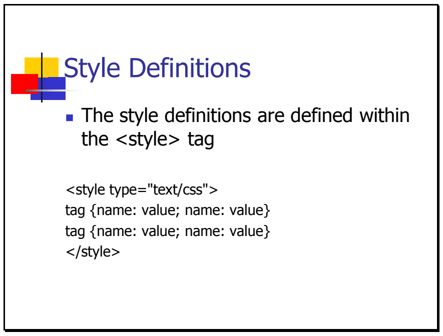
Cascading Style Sheets

- Using Cascading Style Sheets makes it easier to alter text in an HTML document by specifying style rules
- You can place the style rules in the header section of an HTML document

As we saw earlier, you can use the <font> tag to directly set the font typeface and size for text in an HTML document. However, using the <font> tag each time you need to change the font typeface and size can be a time consuming process. In order to make it easier to alter the text and other elements as you create HTML documents, you can use a Cascading Style Sheet. Basically, a Cascading Style Sheet acts as a master list of style definitions to which the HTML document refers when formatting text and layout throughout the document.

While a Cascading Style Sheet can actually be a separate HTML document, the most common way to implement them is to place the contents of the Cascading Style Sheet, i.e. the style definitions, within the header portion of the HTML document in between the <head> tags.

Slide 42



Style Definitions

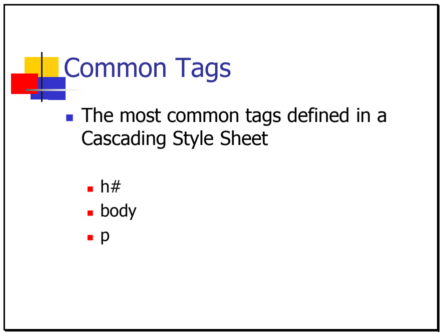
- The style definitions are defined within the <style> tag

```
<style type="text/css">
tag {name: value; name: value}
tag {name: value; name: value}
</style>
```

In order to set up a Cascading Style Sheet, you begin with the <style> tag along with the type=”text/css” attribute and then specify the style definitions. The style definitions identify the tag and then define the attributes to be applied to the text that follows that tag. The style definitions use a special format as shown here.

Take note that the actual definitions, which are enclosed in brackets, use colons to separate the definition name from the definition value and that each definition is separated by a semicolon.

Slide 43

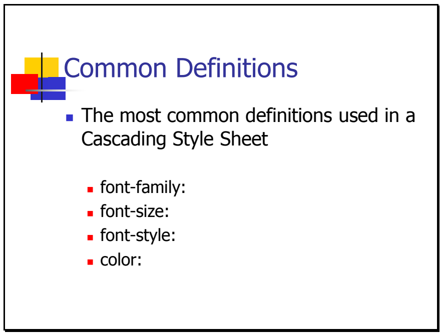


Common Tags

- The most common tags defined in a Cascading Style Sheet
  - h#
  - body
  - p

While you can define almost any tag in a Cascading Style Sheet, the most common tags that you’ll define are shown here. You can specify how you want individual headers to appear, how you want all text in the body to appear, as well as how you want the text in a paragraph to appear.

Slide 44



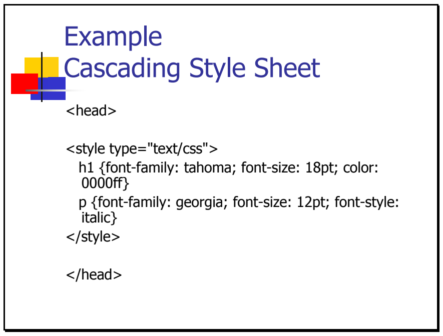
Common Definitions

- The most common definitions used in a Cascading Style Sheet
  - font-family:
  - font-size:
  - font-style:
  - color:

There are numerous definitions that you can use in a Cascading Style Sheet and they can be as specific as you want to make them. However, here are a few commonly used definitions to get you started.

The font-family definition is followed by the name of the font you want to use, the font-size definition is followed by a number that indicates the point size, the font-style definition is followed by a name such as bold or italic, and the color definition is followed by a six digit hexadecimal number.

Slide 45



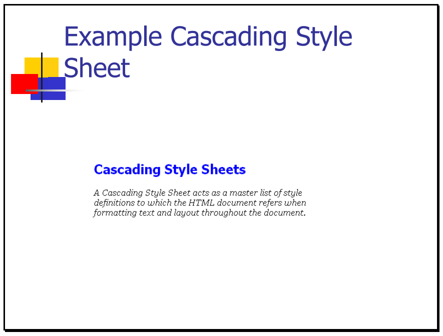
Example Cascading Style Sheet

```
<head>

<style type="text/css">
   h1 {font-family: tahoma; font-size: 18pt; color:
    0000ff}
   p {font-family: georgia; font-size: 12pt; font-style:
    italic}
</style>

</head>
```

Here is a basic example of a Cascading Style Sheet embedded in the header portion of an HJTML document. As you can see, this Cascading Style Sheet contains definitions for the <h1> tag and the <p> tag.

The <h1> tag is defined as using the Tahoma font at 18pt and colored red. The <p> tag is defined as using the Georgia font at 12pt and Italic.

Slide 46



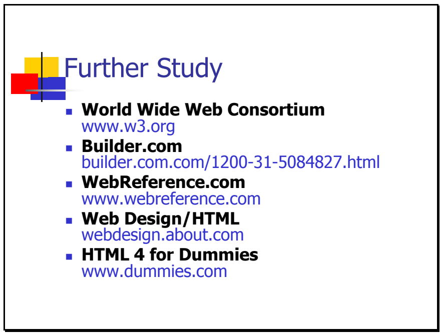
Example Cascading Style Sheet

Cascading Style Sheets

A Cascading Style Sheet acts as a master list of style definitions to which the HTML document refers when formatting text and layout throughout the document.

This example shows how the text in <h1> and <p> tags will appear in a Web browser.

Slide 47



Further Study

- **World Wide Web Consortium**
  www.w3.org
- **Builder.com**
  builder.com.com/1200-31-5084827.html
- **WebReference.com**
  www.webreference.com
- **Web Design/HTML**
  webdesign.about.com
- **HTML 4 for Dummies**
  www.dummies.com

This presentation has covered a lot of material but is by no means a complete tutorial on HTML. However, keep in mind that there are many resources that you can investigate to learn more about HTML. Here are the addresses of four top notch Web sites that feature HTML guides and tutorials as well and the title of a bestselling book on HTML.