



HTML 4.01 (Hypertext Markup Language)

Level 1: Basic Instruction Manual

Basic HTML Concepts & Tags, Text & Image Formatting and Lists

media.gmu.edu

<http://www.w3schools.com/html/>

Reference

HTML Tutorial

<http://www.w3schools.com/html/>



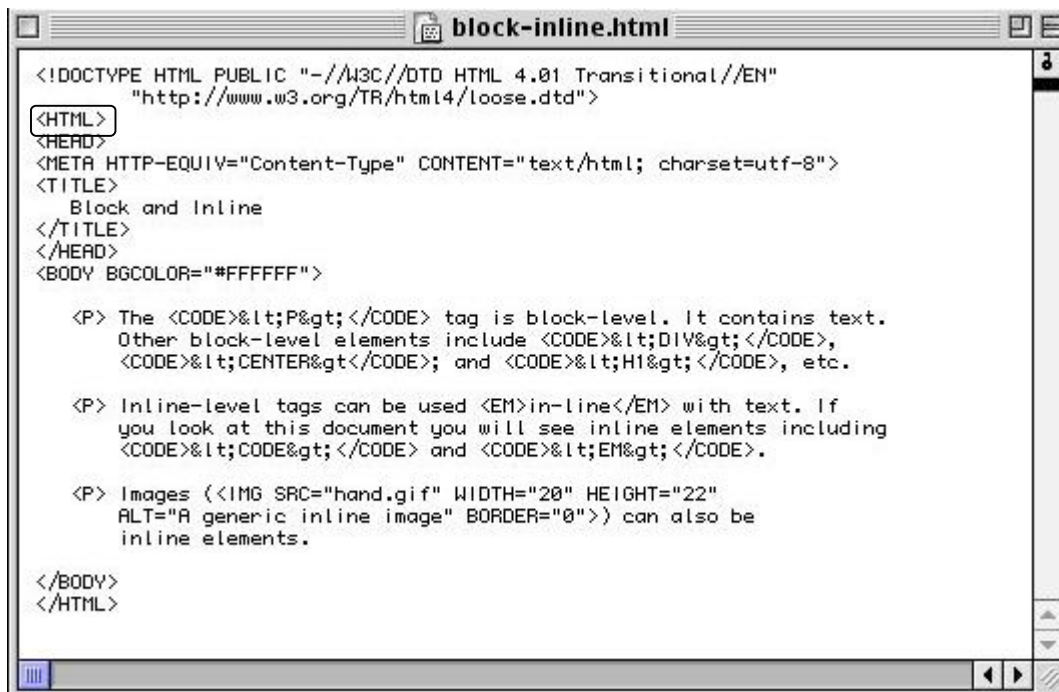
Hypertext Markup Language (HTML) 4.01

Introduction

Welcome to Learning HTML. Our goal in this workshop is to help you understand what you need to know to write the code for building a simple website or to read and modify the HTML code written by another programmer, a WYSIWYG (What You See Is What You Get) editor, or other HTML generators. We hope you enjoy the course and meet your goals as well. For a short course/refresher, go to the [STAR website](http://media.gmu.edu/pagebuilder/). (<http://media.gmu.edu/pagebuilder/>)

Brief History of HTML

What is HTML? It stands for Hypertext Markup Language (though Hypertext is technically one word, it has been given two letters to clearly distinguish it). HTML was invented by Tim Berners-Lee around 1989-1990 when he decided to connect up the world with computers in what he called the World Wide Web. Starting in the 1960s email had become the main communication between scientific communities such as MIT, the Defense Advanced Research Projects Agency (DARPA), and Stanford. The email system had created an expanded network called the Internet that was designed to continue operating even in a disaster like nuclear war. Berners-Lee's idea was to make it possible for researchers and designers to meet virtually in a forum on that Internet, by pulling together and standardizing the many different languages used by other hypertext resources. (At the time, hypertext was used on CDs, floppy disks, and on single, unconnected computers.) As part of the process he invented a language describing documents that could link to each other which he called HTML, based on a previous language called SGML (Standard Generalized Markup Language), designed in the early 80s as a language for providing structure to the media mentioned above. See *Weaving the Web*, by Tim Berners Lee.



```
block-inline.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE>
  Block and Inline
</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">

  <P> The <CODE>&lt;P&gt;</CODE> tag is block-level. It contains text.
  Other block-level elements include <CODE>&lt;DIV&gt;</CODE>,
  <CODE>&lt;CENTER&gt;</CODE>; and <CODE>&lt;H1&gt;</CODE>, etc.

  <P> Inline-level tags can be used <EM>in-line</EM> with text. If
  you look at this document you will see inline elements including
  <CODE>&lt;CODE&gt;</CODE> and <CODE>&lt;EM&gt;</CODE>.

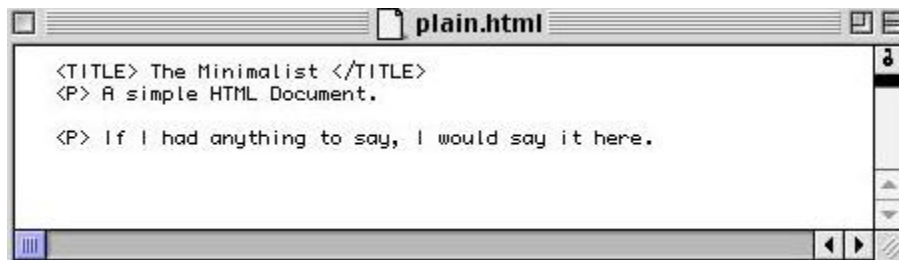
  <P> Images (<IMG SRC="hand.gif" WIDTH="20" HEIGHT="22"
  ALT="A generic inline image" BORDER="0">) can also be
  inline elements.

</BODY>
</HTML>
```

The Snapshot above is an example of an HTML document. You can see that these documents contain keywords which have angle brackets around them (<>). This is a tactic borrowed from SGML. We'll see as we go further into this workshop that it uses many of the features of SGML, including the Document Type Descriptions (DTD).

Getting Started


Now we'll go through a very simple example of HTML. This example is the way HTML was written in its early years.



```
<TITLE> The Minimalist </TITLE>
<P> A simple HTML Document.

<P> If I had anything to say, I would say it here.
```

And today, they make us write the same thing this way...



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE>
  A Simple Document
</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<P> A simple HTML Document.
<P> If I had anything to say, I would say it here.
</BODY>
</HTML>
```

The second document, with all the additional tags, is conformed to HTML 4.01 standard. The reason for this is that when HTML was originally invented, Tim Berners-Lee had no idea how popular it would become. In the time since then, HTML has become the archetypal standard for making the World Wide Web accessible to everyone; creative individuals have added new features, such as Presentation and Markup properties, Objects, and many more. HTML became a sort of a hodgepodge, and browser inventors would make their own versions, to be competitive. After a time, the language was standardized by the W3C (World Wide Web Consortium). Structure has been imposed on this language, and we'll be following that structure.

The second example looks virtually identical to the first, but is correct and standards compliant. Compliant coding is more complex, but will work across the range of browsers. It can also be validated with applications available on the web.

Remember, *you must not use a Word Processor* to write your code because Word Processors use Text Formatting and other Layout features of the application so that the product can be printed. Even if you type

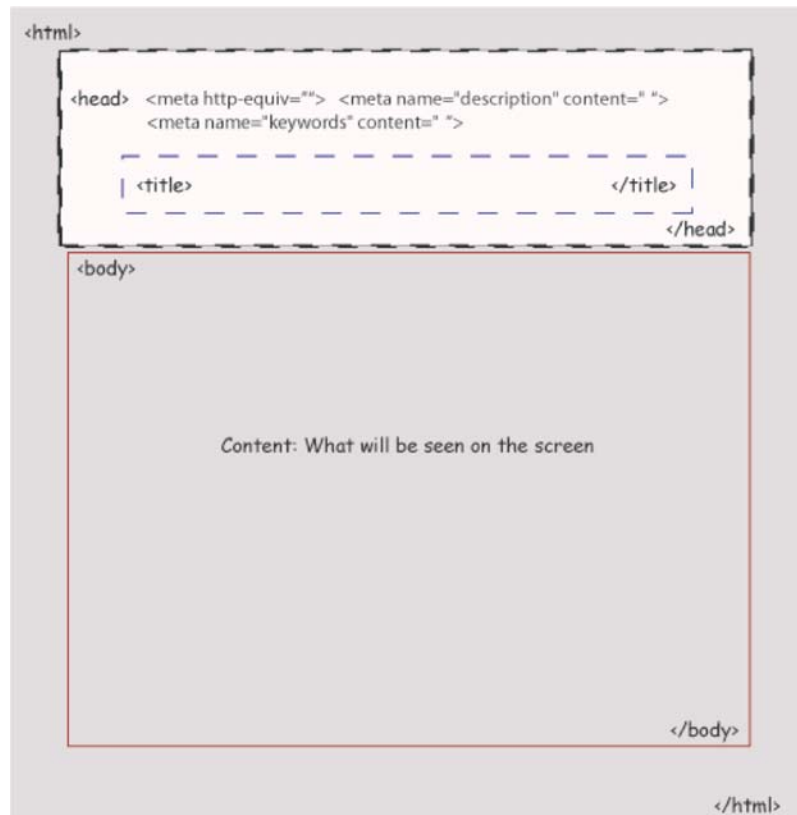
in ordinary code, the Word Processor will add additional information that describes WP layout features and if you put that in your HTML document, it will contain gibberish that will have to be removed, which is time-consuming and unnecessary. Use a plain text editor :

If you are in the windows environment, you could use *notepad*; if you are a UNIX/Linux user, you could use *vi* or *pic*; if you prefer to work on a Mac, you could use *bb edit*.

Basic Concepts

Next, we'll go through some of the basic concepts. The next section, Principles of Web Development, might seem a more likely beginning, but knowing the basic concepts in the first place makes the actual development principles easier to understand.

What is a Tag? A 'tag' is used for marking up text so that it is formatted as a Web document. Tag = `<Keyword Attribute(S)=Value(S)>`



What is close tag (anti-tag)? Any tag you open while writing HTML code needs to close, with a few exceptions. And for closing we use the same tag (excluding the attributes and values if any) with a backslash just before it, as shown below. Tag close= `</Keyword>`

Note: Avoid spaces between the angle brackets and keyword (for both tags and close tags). In later Web-language workshops, use of spaces in tags will be explored.

The individual tags are explained in the table below...

HTML Code Structure Explained...

Line No.	Tag(s)	Description
1	DOCTYPE Declaration:	<p>This tells the browser what type of language it can expect in the rest of the document. In the above case, the DOCTYPE is HTML and it is defined by the W3C in the Document Type Definition (DTD) HTML 4.01. And the URL specifies where the DTD can be found. There are three different types of DOCTYPES for the HTML 4.01 Language viz., Transitional, Strict, and Frames.</p> <ol style="list-style-type: none"> Strict: It is HTML without all the additional features that have accrued over the years to handle things like different fonts. However, it uses Style Sheets which we will discuss in the later levels of this workshop. We will <i>not</i> be using the <i>Strict</i> definition for our exercises. Transitional: We will be using the Transitional html DTD for most of our exercises. The transitional DTD meets the standards and with the W3C clause 'Best Current Practice'. This is a set of language elements that are best understood by the browsers that are out there. If your document is conformed to the Transitional DTD the chances that your document is read by a large array of browsers are high. Frameset: This is used for frames documents and in fact for the documents that only define the frames within which there will be transitional or strict DTDs. The Frameset DTD only contains the language that defines the Frames. We shall discuss these in Level II workshop when we discuss frames.
2 & 12	<html> & </html>	This defines the HTML container, the document itself: everything in between these html start and end tags. The HTML tag tells the browser what language you are coding your webpage in. Everything on your webpage must be between this tag and its close.
3 & 8	<head> & </head>	The Head Tag is used to store information pertinent to the webpage but <i>mostly</i> not viewable by the user. Examples of tags within the head tag are the Title Tags, Java Script, Meta information, CSS Scripts etc.
4	<meta>	Is used for Meta Information that would help the search engines classify the data. The Meta tag also describes the character set of the document which has become a requirement for many HTML validators. The character set recommended by the W3C is UTF-8 and not iso-8859-1. (see below)
5 & 7	<title> & </title>	The Text enclosed between the Title Tag and and the close will be displayed on the top blue bar of your browser window. When an HTML page is bookmarked, the Title is saved as its name.
9 & 11	<body> & </body>	The Body tag is used to contain everything that will be viewable to the user. Therefore, if some content is meant to be seen, or used, it should be between this tag and the close tag.

NOTE: *<TITLE>* and *Filename* are different. Title appears in the top blue bar, as mentioned above. The filename of a webpage is the name of the specific file that the hyperlink goes to.

```
<META NAME="description" CONTENT="Official web site for Student  
Technology Assistance and Resource Center (STAR), George Mason  
University, Fairfax, VA, a New Media Center">
```

```
<META NAME="keywords" CONTENT="workshops, technology training  
assistance, multimedia, multimedia applications, digital video, analog video,  
student mentors, student technology assistance, Macintosh, PC, hardware,  
software, desktop publishing, web authoring, presentations,  
graphics, macromedia, adobe "></head>
```

Examples of META subject matter for use by search engines.

Principles of Web Development

Before we get into more advanced concepts of HTML, it is important that you know and understand how HTML works and the rules you need to remember as your code. The following sub-sections deal with these aspects.

File and Folder Structure

At this stage, we would be at the most designing one or two web pages. But when you build your website, you might have a few dozen pages, with lots of images, style sheets, and maybe other media content. Therefore, it is important to organize your content. We recommend that you place all your images to build your site in a separate folder, and the pics which are part of your page content in a separate folder, all the external documents you have linked from your website like say PDFs or Word Files in a separate folder and so on. Hundreds of different files in a single folder would create much confusion.

Significance of index.html

index.html is the default home/main webpage of every Web site. If I go to <http://www.yahoo.com>, it is actually taking me to the index.html of that website i.e. <http://www.yahoo.com/index.html>. Every folder can have its own index.html page. You can link to each folder just by typing the folder name at the end of the URL and it will automatically take you to the index.html in that folder.

HTM or HTML Extension?

When you save an HTML file, you can use either the .htm or the .html extension. Earlier applications used the .htm extension, and operating systems did not allow four letter extensions, It is now perfectly safe and valid to use either.

File Naming Conventions

Use Lower Case whenever possible, primarily in extensions (.jpg and NOT .JPG).

Avoid Spaces when using multiple-word filenames: names should be multiple_word or multipleWord and NOT multiple word. This is a UNIX convention; other browsers will insert characters like % into the spaces, which can make the file difficult to access.

Use descriptive names

For consistency, stick to a single extension format (.html or .htm).

Dependency on Web Browsers

It is very important that you understand how HTML code is parsed by Web Browsers. Each browser comprehends the code in different ways. Therefore, your code on one browser might appear the way you want it to, but on another, it might appear a little or a lot different. *For best overall results, you should check your site on several browsers before you actually upload it, so that problems can be dealt with.*

We recommend that you stick to the transitional DTD without style sheets and other complex features till you are confident that you have control over the appearance of your website in different browsers.

Using Indentation

Indentation makes the discrete sections of the code more recognizable especially when the code is long, which is normally the case. It is recommended that you adapt to using indentation.

```
<html>
  <head>
    <TITLE> This is the title of the Page </TITLE>
  </head>
  <body>
    This is the Body Content
      <p> This is a paragraph </p>
    Some more Data...
  </body>
</html>
```

Exercises

Exercise 1: Creating a Simple HTML File

Create a simple HTML file with some basic text, save it and test it in different Web browsers.

Exercise 2: Setting a Background Color

Add a background color to a simple HTML page.

Methods...

Format	Example
<code><body bgcolor="colorname"></code>	<code><body bgcolor="red"></code>
<code><body bgcolor="color-hexadecimal-value"></code>	<code><body bgcolor="#ffab36"></code>

NOTE: There are several sources of color available to Web-making. There is color-by-name as noted above; there is RGB color, which is the standard for screen colors; and there is hexadecimal, which provides an expanded range of colors. Much will depend on the screen you are using. There are excellent resources for the use of color, including Holzschlag's Color for Websites, and the color panel on the pagebuilding Web site.

Exercise 3: Setting a Background Image

Methods...

Format	Example
<code><body background="filename.extension"></code>	<code><body background="startlog.gif"></code>
<code><body background="path/filename.extension"></code>	<code><body background="images/startlog.gif"></code>

NOTE: The Background Image tiles by default, that is, unless you tell it not to. It is recommended that background images be small in size, in image formats like .jpg or .gif (as .bmp's are large in size). The .gif format is preferred because .jpgs degrade. If the images are large in size the page will take time to load. Access to Adobe Photoshop is generally most helpful in creating suitable images for Web.

Exercise 4: Adding Text, Paragraphs & Line Breaks

Add text within an HTML page.

Methods...

Tags	Description	Format
<code><p></code>	Used for defining text within a paragraph.	<code><p> Some Text </p></code>
<code>
</code>	Used for inserting a blank line or a line break.	<code>Some Text
 Some More Text</code>

Attributes of the <p> tag:

Attribute	Value	Format
align	center	<code><p align=center>Some Text</p></code>
	right	<code><p align=right>Some Text</p></code>
	left	<code><p align=left>Some Text</p></code>

Exercise 5: Inserting Comments

"Commenting out" a page component can be a wonderful testing resource. You can blank the active text, picture, or other element, but leave the coding intact in case you wish to replace it.

Methods...

Tags	Description	Format
<!--	Opening a Comment	<!-- This Text will not appear on your
-->	Closing a Comment	Webpage -->

Exercise 6: Headings

You wouldn't prefer all of your text in your website to be of the same size. It would naturally contain headings and sub-headings. To insert these, you could use the Heading tags.

Methods...

Tags	Description	Format
h1	Heading 1 (Largest Size of Pre-defined text formats)	<h1> Heading1 </h1>
h2	Heading2	<h2> Heading2 </h2>
h3	Heading3	<h3> Heading3 </h3>
h4	Heading4	<h4> Heading4 </h4>
h5	Heading5	<h5> Heading5 </h5>
h6	Heading6 (Smallest Size of all Pre-defined text formats)	<h6> Heading6 </h6>

Attributes of the Heading Tags (applicable to <h1> to <h6>)

Attribute	Value	Format
align	center	<h1 align=center>Heading1</h1>
	right	<h1 align=right>Heading1</h1>
	left	<h1 align=left>Heading1</h1>

Note: The Align attribute is deprecated in HTML 4.01. It can still be used, however.

Exercise 7: Inserting Horizontal Rules

A Horizontal Rule is simply a line which you could insert using HTML. For instance, it can be inserted between the Heading and text. **Method:** Some Text <hr> Some Text (**Note:** <hr> has no closing tag.)

Attributes of the <hr> Tag

Attribute	Value	Format
align	center	<hr align=center>
	right	<hr align=right>
	left	<hr align=left>
size	(pixels or %)	<hr size=80>
width	(pixels or %)	<hr width=5%>
Example: <h1> Main Page </h1> <hr align=left size=300 width=2%> <p> This is the main page of the document </p>		

Methods. . .

Tags	Description	Format
 or 	Bolds the text enclosed within these tags.	<code>This text is Bold</code>
<i> or 	Italicizes the text enclosed within these tags.	<code><i>This text is Italicized</i></code>
<u>	Underlines the text enclosed within these tags.	<code><u>This text is underlined</u></code>
<sup>	Superscripts the text enclosed within these tags.	<code><sup>This text is superscripted</sup></code>
<sub>	Subscripts the text enclosed within these tags.	<code><sub>This text is subscripted</sub></code>

Exercise 8: Text Formatting

Formatting text would includes **Bolding**, *Italicizing*, Underlining, and **some combinations** of these three. You can also ^{superscript} or _{subscript} your text for footnotes or math.

Exercise 9: Creating Hyperlinks

HTML uses the **<a>** (**anchor**) tag to create a link to another document. An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor: `Text that would link to the URL when clicked`

Example: ` Google `

The **<a>** tag is used to create an anchor to link from, the **href** attribute is used to address the document to link to,

and the words between the open and close of the anchor tag will be displayed as a hyperlink.

Other Attributes of the <a> Tag

Attribute	Value	Description	Format
target	<code>_blank</code>	The target URL will open in a new window	<code></code>
	<code>name</code>	The target URL will be at a different place on the page, or a specific link on another page.	<code></code> identifies trgt <code></code> creates link <code></code>

Exercise 10: Using the Font Tag

The Font tag enables you to color your text, size it to any value you want, or choose from different font types. Since the font tag does not independently perform any action without its attributes, we will discuss the attributes in the table below.

Attributes of the Tag

Attribute	Value	Format
size	pixels	<code> Text </code>
color	color-name (or) color-hexadecimal value	<code> This text is in Red color </code> <code> This text is in Red color </code>
	face	font-name

Example: `Green Text Blue Text`

Exercise 11: Creating Lists

There are two types of lists **Ordered Lists** and **Unordered Lists**. Both list items. The items of an unordered list are marked with **Bullets** and the items of an ordered list are marked with **Numbers OR Letters**.

Methods...	
List Type	Format
Ordered	<pre> Item1 Item2 </pre>
Unordered	<pre> Item1 Item2 </pre>

Attributes of the Lists Tags			
List	Attribute	Value	Format
Ordered	type	A, a, I, i, 1	<ol type=A> OR <ol type = 'I'>
Unordered	type	Disc, Square, Circle	<ul type="disc"> OR <ul type = "Circle">
Ordered	start	Number ONLY (3, 10)	<ol start=20> OR <ol type=A start=3>

Exercise 12: Inserting Images

In HTML, images are defined with the **** tag. The **** tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the **source** attribute, **Src**. The *value* of the src attribute is the image file name or the URL of the image you want to display on your page.

The syntax of defining an image: ``

Attributes of the Image Tag			
Attrib	Value	Description	Format
src	URL	Describes the image source	<code></code>
Alt	Plain Text	Alternate Text	<code></code>
width	Pixels (OR) %	Defines the width of the image	<code></code>
height	Pixels (OR) %	Defines the height of the image	<code></code>
border	Pixels (OR) %	Defines the Border Size of an image. Beveled border of size=0 by default.	<code></code>

Exercise 13: Aligning Images within Text

Images can be inserted anywhere inside paragraphs as you can see in the picture below.

In the first case, the text was typed in first and then the image and in the second case the image was inserted in the **<p>**tag first and then the text was typed in. Notice that in both the cases the bottom of the image is aligned to the text. You might want your image to be aligned like the ones below (middle or top alignment).

HTML Tutorial: <http://www.w3schools.com/html/>



OR



HTML Tutorial: <http://www.w3schools.com/html/>

HTML Tutorial: <http://www.w3schools.com/html/>



OR



HTML Tutorial: <http://www.w3schools.com/html/>

To achieve this, you would need to use the align tag.

Attributes of the Image Tag

Attrib.	Value	Description	Format
align	top	Aligns top of image to text.	<code><p> This is the Logo of Star Training: </p></code>
	middle	Aligns middle of image to text.	<code><p> This is the Logo of Star Training: </p></code>
	bottom	Aligns bottom of image to text. This is the default alignment.	<code><p> This is the Logo of Star Training: </p></code>

Exercise 14: Creating Image Links

Example: ` `

Methods...		
Escape Sequence	Definition	Description
&nbsp;	Non-Breakable Space	Inserts an additional space
&gt;	Greater Than Symbol	Inserts >
&lt;	Less Than Symbol	Inserts <
&amp;	Ampersand	Inserts &

Exercise 15: Escape Sequences

Certain characters cannot be displayed normally like the others in a webpage by mere typing. The reason is that they are probably pre-defined for specific purposes like the angle brackets <>, or the ampersand &. To insert such symbols as part of your text, you will need to append special sequences to the text and these are called escape sequences. Also, you might have noticed that the HTML parser ignores additional typed spaces between words. Using Escape sequences, you can insert additional space.

Note: For a single space, you needn't use ** **. Use it only if you require more than one space character to be inserted. For more Escape Sequences check out the HTML Character Entities section at the W3Schools Website. The URL is http://www.w3schools.com/html/html_entities.asp.

Difference between HTML 4.01 and XHTML 1.0

The most visible difference between Extensible HyperText Markup Language (XHTML) and its close companion HTML is in their syntax. XHTML is stricter on the syntax rules, for example, on the presence of the closing tags and on character case (no caps), which are not so important in HTML.

XHTML follows the Extensible Markup Language (XML) rules and syntax guidelines. And because XML has very rigid requirements for writing code, XHTML is a more structured markup language than HTML. This actually enables one's documents to be viewed on multiple devices and browsers without errors.

1. The name space must be declared in the HTML tags.
2. The <head> and <body> tags cannot be omitted.
3. The <title> tags must be listed first after the opening <head> tag.
4. Code all of the tags, attributes and values with lowercase text. For example, instead of < A HREF="Index.html">.
5. Always use end tags for tags that enclose content. For example, every <p> opening tag used for a paragraph must have a </p> closing tag.
6. For self-closing tags, place a space and forward slash in front of the end caret.
(i.e. ,
, <meta /> and <hr />).
7. Code the attribute name with quotations. (i.e. <alt="GMU logo">).
8. In designing forms, ensure all attributes have values. (i.e. <input type="checkbox" checked="checked">).
9. In designing forms, use id attributes. (i.e. <input type="checkbox" id="Excellent">).
10. XHTML requires that special characters be converted to entity references.

In a nutshell, XHTML 1.0 is basically HTML 4.01 adding the syntax of XML. The positive is that XHTML facilitates style sheets, and so is the language for CSS; it creates structured data. XML defines content separate from formatting, so it can be used to put your Web site on someone's Blackberry, for example.

[Example of use of codes in the body tag](#). Remember, in HTML, tags that open also must close, except for
, <hr>, and (although in XHTML, these also must close).

<html>

Content: What will be seen on the screen

```

<body bgcolor=" " text=" " link=" " vlink=" " >
<font face=" " size=" " color=" " > </font>
<h1-6> </h1-6> 1 being largest

<p align=" " > </p> paragraph
<br>* line break

<hr align=" " size=" " width=" " >*



<a href="file name or url" >link text</a>

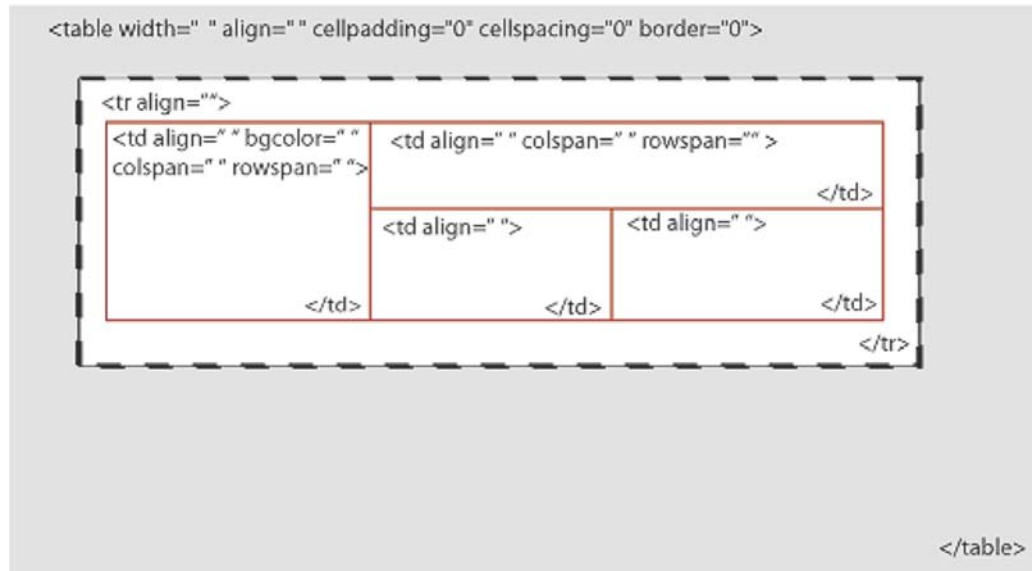
<ol type="1, A"> </ol> Ordered list
<ul type="disk,square,circle"> </ul> Unordered list
<li value=" " >*

<!-- comment-->

</body>
</html>

```

Example of use of table tags to shape content



Thank You for Coming

The Workshop Evaluation can be found at:

<http://ittraining.gmu.edu>

Select "Workshops," "Evaluate a Workshop," "HTML 4.01," then "Eval."

Password: byte

If you are interested in exploring other training opportunities on the George Mason University campuses, please visit the IT Training website at <http://ittraining.gmu.edu>.