

Introduction - What is HTML?:

HTML (Hypertext Markup Language) is the language used to make a web page. HTML tells your browser how the web page should appear. It's the "source code" of a web page. As computer languages go, HTML is relatively simple. Anyone can learn it.

Getting Started:

There are a few things you'll need before you actually get started with your HTML programming.

The first is a web browser to view the page (Microsoft Internet Explorer should be your first choice, with Netscape Navigator in second; other choices include Mozilla and Opera).

Second, you'll need a program to type and edit the HTML. Since HTML is a text-only document, programs such as Notepad (for Windows users) or SimpleText (for Mac users) are perfect. It's OK if you wish to use a word processor like Microsoft Word however, keep in mind that you **MUST** save as "text only," otherwise, the word processor will add its own proprietary formatting. In most programs, this is done by selecting "Save As" and changing the option in the drop-down box from whatever format your program is, to "Text Only."

When saving, always make sure that you are saving with an ".htm" or ".html" extension (".html" is the preferred), instead of ".txt" (even though they *are* text documents). The extension ".html" tells the browser to interpret the text as HTML, rather than just text.

You don't even need to be online to write and edit (or even view!) your HTML (and thus, web pages). Once you want to get them up on the web and view them there, however, you will need to have a connection to the internet.

Viewing the Source:

One the most helpful things potential HTML writers say to do is to simply look at the source code of other web pages. This allows you to get some kind of idea of how the final coding should appear, how tags interact with each other, etc. To do this in Internet Explorer and Netscape, click the "View" menu, and select "Source" (or "Document Source"). The page's source should open up in Notepad or a new browser window.

At first, the code will look incomprehensible. But as you begin to learn HTML, the code will begin to make more and more sense. Like any language, you increase your literacy by reading.

Tags Primer:

HTML is a very basic language to learn; in fact it's not even really a computer language! Nothing needs to be "compiled" by another program to then be used; once it's saved, it's ready. The majority of coding resembles natural spoken English. To format your page, "tags" are used at the beginning of a section and end of a section (opening and closing tags, respectively). This indicates to the browser where to begin and end a certain kind of formatting.

Since HTML is text-only, you must, in the coding, tell the browser to bold, italicize, or underline text; there is no "bold" button to click. You need to "command," in a sense, the browser to start the bold format, and then to end it at some point; you would put an opening bold tag at the beginning of a word, and a closing bold tag at the end of a word, for example.

Tags are identified by the "less-than" and "greater-than" signs (< and >). An opening tag always has "<" and ">" and a closing tag always has "</" and ">" (yes, the slash is needed). The tags will not show up on the page when a browser views them; they are there in the background for the browser to interpret. It is good common practice to use capital letters for your tags (for example, "" and "" for bold tags). This will help you quickly identify where tags are when viewing your source code.

It's possible to have text with no tags formatting them; however, the text will appear in "default font" of the browser. This is bad practice, as you want your page to look as similar as possible across multiple platforms and browsers.

Making a Page:

So you've got a blank text document open, and you're ready to make your first HTML page. The first thing you'll want to do is type in the opening HTML tag:

```
<HTML>
```

This one should be pretty obvious. You're making an HTML document, so you need to indicate in the coding that it's an actual HTML document. You don't want to close this tag yet; that'll be the very last thing you do.

All HTML documents are divided into two parts, the HEAD and the BODY. The HEAD contains special tags that support the page; the BODY has your actual content. So, start with <HEAD>--we'll close it soon.

First, you'll want to indicate what the title bar of the browser should say when someone comes to your page. This is done with the <TITLE> tag. For example, you might have:

```
<TITLE>My Personal Home Page</TITLE>
```

Now that you've got an HTML document initialized, and you've got a title for it, it's time to close the HEAD with </HEAD>.

Now we'll tell the browser to start the BODY of the document with <BODY>. We'll close this tag after we put in all our content.

Type out some words, maybe even a paragraph from a book. Don't worry about how to format it yet; first we want to make sure that what you have works. Once you've got something written, it's time to close out the HTML document. First, close the body </BODY> then enter in the closing HTML tag (</HTML>), and you're set. Make sure the file is saved as text-only, and with a file extension of (preferably) ".html". You may need to change the filename from "example.txt.html" to "example.html" if your text program insists upon putting in the ".txt".

Testing a Page:

Find the file you've created on your hard drive, and simply double-click it. The HTML document (which is really just a text file saved as ".html") should open up in a web browser, and contains the words that you've typed. Notice that the browser does not show any of the tags you entered. If you don't see your content, or see extra content, make sure your <HEAD> </HEAD> and <BODY> </BODY> tags are right.

Basic Text Formatting:

You may have noticed that in whatever text you typed, if you hit "enter" to make a new paragraph, it didn't show up once you viewed the page. That's normal! Since HTML is really just text, and there's no character to represent a line break, you have to code a line break yourself.

 - Think of it as "BReak line." The text will go down to the next available line.

<P> - Think of it as "Paragraph." The text will go down to the next line, and then one more, to create a clear break between two paragraphs.

NOTE: The above tags are special. They have no closing tags.

You may also want to format your text:

 - Bold

<I> </I> - Italicize

<U> </U> - Underline

These are pretty self-explanatory text formatting options. You should be familiar with them if you've used any kind of word processing program. Simply surround the text you wish to format (for example, "This word is bold"). You can even use more than one, if you wish. Remember to close all tags, and try and keep them in order as you use them (for example, "<I>bold and italic</I>")- first opened, last closed.

You'll probably also want to format the size and font of your text at some point. The tag allows you to do this. For example, if you wanted the font for your text to be a size 12 Arial, the syntax of your coding would look something like this:

```
<FONT SIZE="12" FACE="Arial">insert all of your text here</FONT>
```

The "FONT" indicates to the browser that you're formatting the text. The "SIZE" lets the browser know what size to place the text at (the sizes are the same as they would be if you were working in a word processor). Finally, the "FACE" lets the browser know which exact font to write in. "SIZE" and "FACE," in this case, are examples of tag "attributes." Most tags have attributes that let you specify even more how the tag should affect your content. To use an attribute, include it in the opening tag, add an =, and put the specific info in quotation marks.

You'll notice that all of your text will be left-justified. To center anything on your page, simply surround it with the <CENTER> and </CENTER> tags.

However, if you're going to be aligning text alone (rather than images, etc.), it is better programming to use the ALIGN attribute of the <P> tag. You'll remember the <P> tag from earlier, when creating a new paragraph. If you want to have a single paragraph be aligned to the right, it would look something like this:

`<P ALIGN="Right">insert all of your text here</P>`

You'll notice that now there is a closing tag ("`</P>`"). Since you are using the `<P>` in conjunction with new attributes, you must close the tag.

Images :

You can place images on the page along with your text. To do this, have your image in the same folder (directory) as your HTML file (for the time being). The only types of images you should have on a web page are JPEG and GIF files; anything else would take too long for people to load up and view, especially on dial-up internet connections.

To show that image on your page, the coding would look similar to this:

``

The "IMG" tells the browser that you're going to be showing a picture. The "SRC" attribute tells the browser where to look, and what file it is, exactly (think "source"). You can do more with the IMG tag, however. Look at the following example:

`<IMG SRC="filename.gif" HEIGHT="20" WIDTH="20" ALT="This is my image"
BORDER="1">`

The "HEIGHT" indicates exactly how tall (in pixels) the image is, while the "WIDTH" indicates the width. By putting in these two options, the browser can draw the space where the image would be without even loading it first, and move on to loading the rest of the HTML file. This speeds up viewing times for browsers. The "ALT" is an interesting option. The text that you place in an "ALT" tag will appear on the web page when you place your cursor over the image it belongs with. Finally, the "BORDER" will place a border of indicated width (in pixels) around the image.

There are even more attributes you can use with your images. You'll probably notice that if you place an image within text on your page, the text will not flow around the image; it will appear as if the image has its own line in the paragraph (technically, it does!). To align the images with the text, you will use the "ALIGN" attribute in the `` tag. The image will automatically be aligned to the left, by default, once you put it in. However, if you use `ALIGN="LEFT"` in the tag, the text will flow around the image on the right, as the image is justified to the left. The same effect can be done by using `ALIGN="RIGHT"` to have the image justified right, with text flowing around on the left.

Links :

Technically speaking, there are two types of links: one links to something directly on the same page, the other links to a completely different file (this file can be another HTML document in your site, an image, a different website, etc.).

An "anchored link" is a link on a page that links to elsewhere on the same page; this is commonly seen on FAQ pages where the question links downwards on the same page to the answer. To accomplish this, you'll first need to set an "anchor" somewhere on a page. Find some kind of text towards the bottom of your HTML document that you'd like to link to. Before that text, enter the following:

```
<A NAME="myanchor">
```

The "NAME" attribute of the <A> tag indicates that you're naming an anchor. Now, find some text near the beginning of the page that you wish to have link DOWN to where the anchor is. Your link should look something like this:

```
<A HREF="#myanchor">text that you wish to link down to anchor</A>
```

The "A HREF" attribute of the <A> tag opens the link, while the "#" tells the browser that the link is somewhere on the same page. The closes the link.

To link to a different web page in the same folder as your current page, the link would look something like this:

```
<A HREF="otherpage.html">text that you wish to link</A>
```

To link to a completely different website, such as Google, the link would look something like this:

```
<A HREF="http://www.google.com">text that you wish to link</A>. If the HREF attribute does not have "http," the browser will look for the page in your site.
```

Directory Structure (in terms of links):

You can have different folders for your website, each containing different HTML files. You can link between folders, too. However, you'll have to keep in mind the directory structure of your site as you're coding the links.

Suppose you have a main folder. Inside this folder, you have two other folders: "one" and "two". Folder "one" contains an HTML file you'd like to have link to another HTML file that is in folder "two." The link would look similar to this:

```
<A HREF="../two/htmlfile.html">text you wish to link</A>
```

The "../" is used to show going up in the directory structure, and can be combined as many times as necessary to get to the desired folder.

Directory Structure (in terms of filenames):

You'll probably notice that when you go to a website (say, www.rutgers.edu) that there is no filename (like "page.html") in the address bar. Why is this? Shouldn't all pages have some kind of filename? They do!

Hopefully you've discovered the ease of using directories on your site to organize different sections. These directories lend a neat little option to you. If you put a file named "index.html" into any directory, the browser will automatically load up that "index.html" page if the directory is in the address bar.

Huh?

Consider this example:

<http://www.yoursite.com/folder/index.html>
<http://www.yoursite.com/folder>

It doesn't matter which of the above two you type in. Each would go to the file "index.html" by default.

Email Links:

Somewhere on your page, you'll probably want to have a link so that your visitors can e-mail you with any feedback or questions they may have. This is done similar to a link you've already made, with a slight adjustment.

```
<A HREF="mailto:address@provider.com">text to link here</A>
```

The `<A HREF>` should look familiar to you from traditional links. The "mailto:" lets the browser know that this is a link for an e-mail program; not the browser. The "address@provider.com" should be replaced with your e-mail address.

Tables:

There's an easy way to organize things on a page, without having to go through the hassles of line breaks and aligning. Wouldn't it be easy to just create a table with rows and columns to format your page? Wouldn't it be nice if you could even make this table invisible, for that extra-professional look of organization? HTML makes it easy!

The following example is a very basic table. Take a look at the tags.

```
<TABLE>

<TR>
<TD>One</TD>
<TD>Two</TD>
<TD>Three</TD>
</TR>

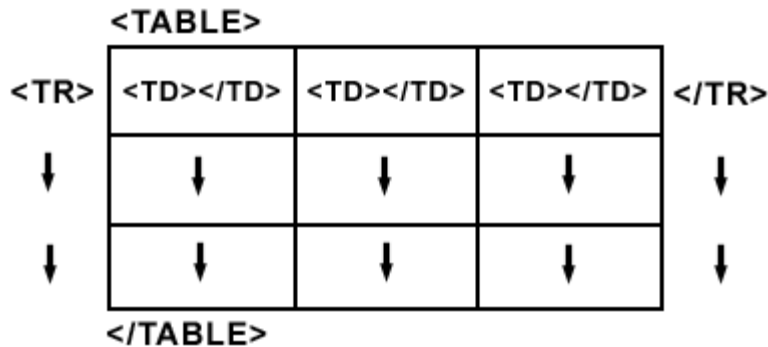
<TR>
<TD>Four</TD>
<TD>Five</TD>
<TD>Six</TD>
</TR>

<TR>
<TD>Seven</TD>
<TD>Eight</TD>
<TD>Nine</TD>
</TR>

</TABLE>
```

How exactly is this working? `<TABLE>` lets the browser know that, well, the table is about to start. `<TR>` starts a new Table Row. `<TD>` indicates a new column ("table

data"). So we've got a table of three rows and three columns. As in the example, make sure to close out all tags. Some browsers, especially Netscape, are extremely picky about displaying tables if the HTML is not coded perfectly. TABLE is a tricky tag. Perhaps this illustration will help:



So you've got a basic table, and now you want to really start formatting it. In the initial `<TABLE>` tag, we can add in a few attributes to alter how the table looks. Look at the following example:

```
<TABLE WIDTH="500" BORDER="0" CELSPACING="2" CELLPADDING="2">
```

The `WIDTH="500"` tells the browser that the absolute size of the table is five hundred pixels wide; no matter how many columns you add, the size will never exceed this number. The `BORDER="0"` is telling the browser to draw a border of zero pixels; essentially an invisible table! You can put any number you wish, here; be aware of just how large you're making the border, though. `CELLSPACING="2"` tells the browser to put two pixels of space between cells. This keeps things from being too close to each other inside the table. `CELLPADDING="2"` tells the browsers to put two pixels of space between individual cell borders and the contents inside the cell; this also keeps things from being too close to each other, inside the table (imagine putting your cell content into a padded cell).

You may want to align the contents of a cell to the left and right. Similar to images, you can do this, but inside the `<TD>` tag. For example:

```
<TD ALIGN="RIGHT">
```

... would align the contents of the cell justified right. You can use `LEFT`, `CENTER`, or `RIGHT` as you wish.

What will you place inside of a table? You can put anything that you'd normally put on a page inside the cell of a table. Tables are great for organizing links, image galleries, etc. Experiment with the size, border, etc. of tables to see what works best for you.

The `<BODY>` Tag (and everything inside it)

Wouldn't it be nice if there was a way to set the text color, background color, etc. of your page all in the same place? The `<BODY>` tag allows you to do this. The `<BODY>` tag comes after the `<HEAD>` tag in the beginning of your HTML document.

```
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#800080"  
ALINK="#FF0000">
```

What's going on here? "BGCOLOR" is telling the browser to set the background color of the page whatever is contained in the quotes. What is "#FFFFFF" though? This is called a hex code. Colors aren't represented by name, they are represented by a six-digit number. In this case, "#FFFFFF" is pure white. The text of the page is set to "#000000" (black). You can also format the color of links that have not been visited, links that have already been visited, and links that are being clicked (LINK, VLINK, and ALINK, respectively). In this case, they are set to blue, purple, and red (respectively), which are the standard among most websites.

A hex code is actually divided up into 3 sets of two numbers each (forming the six-digit number code). Each pair represents a color of the RGB code (RGB = Red, Green, Blue). The first pair in the hex code represents the red factor, the second pair equals green, and the third pair represents the blue factor. For example, a full shade of red in RGB code would be Red=255, Green=0, Blue=0. However, in hex code, the color would look like this: #FF0000. The hex code is the way to interpret RGB colors in HTML.

META Tags:

Many people wonder how to get search engines to find their site. Sure, submitting your address to their catalogue is helpful, but there are other things you can do to help people find your site.

While offering no visual advantage on your pages, META tags work behind the scenes to help your site be found. These tags are placed after the <HEAD> tags at the beginning of your HTML document. Check out the following example:

```
<META NAME="keywords" CONTENT="one,two,three,four">  
<META NAME="description" CONTENT="brief description of website">  
<META NAME="generator" CONTENT="Notepad">  
<META NAME="author" CONTENT="your name">  
<META NAME="copyright" CONTENT="Copyright © 2002">
```

How are these helpful? The "keywords" section allows you to identify as many key words and phrases as you wish that uniquely identify your site. Search engines use these words to compare to what users enter in for search results. The "description" section is what will appear on the search engine when someone searches for your site. Normally the search engine would display (approximately) twenty five words that come first on your page; this makes no sense to the user. By coming up with a description, you can immediately let the user know what kind of site you are offering. The "generator" section lets you enter in what you used to make your page; it's not all that helpful, but can be interesting. The same idea goes for the "author" and "copyright" sections.

Outside Resources:

[HTMLGoodies < http://www.htmlgoodies.com >](http://www.htmlgoodies.com)

Regarded as the ultimate resource for anything regarding HTML and web design,

HTML Goodies covers everything from the basic creation of an HTML document, all the way up to the very latest in XML, Java, etc.

[Introduction to HTML](http://www.cwru.edu/help/introHTML/toc.html) < <http://www.cwru.edu/help/introHTML/toc.html> >

A nicely organized, comprehensive tutorial (with sequels, no less) to nearly everything you'd want to learn before HTML, during your HTML coding, and after.

[HTML Primer](http://www.htmlprimer.com) < <http://www.htmlprimer.com> >

Organized into "lessons," the primer leads you from beginning to end, with additional discussions, glossaries, and advanced topics such as JavaScript.

[The HTML Pit Stop](http://www.pageresource.com/html) < <http://www.pageresource.com/html> >

Ranging from getting started with HTML, going through even topics such as DHTML, all the way up to posting your site, the Pit Stop covers all your needs.