



How the Web Works

When you type a url into the address bar of your browser and hit return, the following occurs, assuming you typed in the correct url:

- you request the document you specified to be sent to your web browser, or the standard *index.html* if no document name was given.
- The document is then served to your web browser one line at a time.
- If the line being served is standard, plain html there is no processing performed, and you are served that line of code instantly.
- If the line of code contains a web language such as Perl, PHP or ASP, that block of code from this point to the ending marker is processed on the web server, and your web browser receives the results of the processing.
- When there are no more lines to serve you, the connection is closed.

This is how servers got the name - they serve you documents.

The processing function is what makes code such as Perl, PHP and ASP so powerful, and sought after - items are processed on the server before anybody receives anything, and we are simply given the results of that processing. In other words, you could safely put a list of usernames and passwords on your server, process a login script with PHP, and you are given a result of valid or not valid - nobody ever sees the list. To perform the same function in basic HTML is impossible - you would have to be served the above list and let your web browser process the information - that would mean anybody visiting your web page would also have the list served to them as well ... not very secure.

For the scope of this document we are going to concentrate on building a simple web page using nothing more than basic HTML. We should at least learn the structure and flow of a web page before we get into complex processing.

Basic Text Document

To write HTML you will need a text editor of a sort, or an

HTML Editor application. The end of this document has a few places to look for HTML editors. When you write HTML it is important to save it as an HTML type document, and not TXT, DOC, RTF, or some other extension. If you are using Notepad, it is important to select *All Files* in the *Save as Type* drop-down box before you save anything, and supply the **.HTML** extension.

HTML Tags

HTML works in a very simple, very logical, format. HTML is written with TEXT in a simple document editor such as Windows Notepad, or in a commercial/freeware HTML editor. What you use to define sections such as headings, subtitles, bold text, underlined text, etc is a series of tags.

Think of tags as making your structure. Let's say you want a heading. You will put a tag at the exact point you want the heading to start and another tag where you want the heading to stop. If you want just a word to be emphasized, you will place a start emphasis tag at the beginning of the word and an end emphasis tag at the end of the word.

Tag Format

All tag formats are the same. They begin with a less-than sign: < and end with a greater-than sign: > – always. The tag is contained inside the < and >. Learning HTML is simply learning which tag performs which result. Most tags will wrap around the text you are trying to define with a start and end tag. For example, you would begin a paragraph with the <p> tag, and end the paragraph with the </p> tag. With the exception of a few tags you will see that all HTML is a series of start and end tags. Here are a few basic tags we will use to get things started:

Tag	Name	Description
<p></p>	paragraph	Defines the block contained in the tags as a paragraph, and ends with a line break.
	bold	renders the text contained within the tag as bold.
	strong	identical to

<i></i>	italics	renders the text contained within the tag as italics.
	emphasized	identical to <i>
 	break	breaks the text and starts again on the next line. Line breaks do not occur in an HTML document as displayed in your source code. You must use this tag to perform a line break.
<hr>	horiz. rule	inserts a horizontal line, and the text flow begins on the next line. The horizontal line will be the entire width of the browser window you are viewing the document in.

You can also use multiple tags (nesting) on the content you are defining. The important thing to remember is getting into the habit of nesting your documents tags with flow and logic. Basically, when you nest tags you begin them how you wish, but close in a reverse order. Here is an example that will clarify what is meant with less words:

```
<i><b>Hello World</b></i>
```

Writing Our First Page

A web page is structured as easily as the tags are. HTML is basically blocks of code defined with tags. View this HTML document, or of course type it in and follow along. We will now step through the code line by line:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>My first HTML document</title>
  </head>
  <body>
    <p>Hello <strong>world!</strong>
    </p>
    <br>
    <p>This is my very first HTML page.
    </p>
  </body>
</html>
```

Line 1-2 define the type and language of the HTML on the page. For our purposes in this document, this isn't really necessary, but as your pages grow in complexity this is very important.

Line 3 is the start tag for our HTML

Line 4-6 is the document's head. The head is where you'd find the title, javascript, css, and other special code that is accessed by the main body. Notice how line 6 ends the head.

Line 7-13 is the body of our small document.

within the document you'll notice how we have formatted certain parts with beginning and ending tags, and they have a logical flow.

Line 14 ends the html document.

This document points out the structural element of a web page, and a few formatting tags. You will find a complete list of tags and attributes at the end of this text. You are encouraged to play around with tags until you are used to the way they fit into a web page.

Creating Links

Using links in your web pages is probably the *meat and potatoes* of the internet. After all, without links we just have static pages of text and a vast amount of typing to do! Links are created in the same *wrapper* fashion as any other HTML element. First we will dissect a link and show you the different parts. A typical link would look like this:

```
<a href="http://www.biccweb.com">Click here for BICC</a>
```

Links begin with an *Anchor* `<a>` and of course have a closing tag of ``. The *href* is an html reference, or just another way of telling the browser that this is going somewhere. Within the quotes is the actual *url* you wish to link to, in this case it is BICCs new web site. After the `>` and before the ending anchor is the element you click on to zoom away to this link. In this case it is simple text that will be underlined and blue (by default), but it could just as easily be an image.

A common attribute to use with the anchor tag is the `target=` attribute. Using this attribute you can define where your clicked link will open. Most web designers have a link open on a brand new page, this way we are still on the original site after we close the page. Originally used for sites with frames, this continues to be a common

element in a web page. The modified version of the above link, that opens on a new page, would look like this:

```
<a href="http://www.biccweb.com" target="_blank">Click here for BICC</a>
```

The other common href directive would be `Send E-mail` ... although in today's spyware, web-crawling, e-mail harvesting world I wouldn't recommend using this one.

Working With Images

Images in an html document are a little different than text, simply because they must load from somewhere - a server on the internet, or a saved location on your computer. For the next while I have placed some images on my server space in case you want to play with html, and they are shown at the end of this document. Images are placed in an html document with the following tag, and you will notice there is no closing tag for this one.

```

```

The location of the image can be anywhere accessible, but it's usually your defined server space. Be aware that many web creators don't appreciate their graphics being linked to, and usually take precautions to prevent people from *hot-linking*. For our purposes here, we will create an image tag to the first graphic in our list:

```

```

This code would simply show that particular image at the point you inserted it into your html document. We can put everything learned so far together as a web link to Flickr.com (a photo-sharing website):

```
<a href="http://www.flickr.com" target="_blank"></a>
```

Building a Functional Web Page

We can build a functional web page with formatting and links, and it could be the basis of your personal Home page if you wish every time you start up your browser. For our purposes here we will simply build a link page to

various sites.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>
  <head>
    <title>My Start Page</title>
  </head>
  <body>
    <b>Local Websites</b><br/>
    <hr>
    <a href="http://www.biccweb.com">
      BICC</a> - Computer Club<br/>
    <a href="http://www.mcsbrockville.com">
      Micro Computer Service</a> - Computer
      Sales/Service<br/>
    <a href="http://www.recorder.ca">
      Recorder and Times</a> - Brockville's
      newspaper.<br/>
    <br/>
    <b>Software Downloads</b><br/>
    <hr>
    <a href="http://www.download.com/">
      C-Net Downloads</a> - Various software
      downloads<br/>
    <a href="http://downloads.zdnet.com/">
      ZDNet Downloads</a> - Various software
      downloads<br/>
    <a href="http://www.freewarefiles.com/">
      Freeware Files</a> - Thousands of free
      downloads<br/>
    <br/>
    <b>Security Tools</b><br/>
    <hr>
    <a href="http://www.free.gtisoft.com/">
      AVG</a> - Free antivirus utility<br/>
    <a href="http://www.free.gtisoft.com/">
      Ewido</a> - Free spyware
      scanner/cleaner<br/>
    <a href="http://ccleaner.com/">
      CCleaner</a> - Clean-up garbage on your
      system<br/>
    <a href="http://lavasoftusa.com/">
      AdAware</a> - Free spyware
      scanner/cleaner<br/>
  </body>
</html>
```

If you type everything in correctly you should have a web

page that resembles something like this:

Local Websites

[BICC](#) - Computer Club
[Micro Computer Service](#) - Computer Sales/Service
[Recorder and Times](#) - Brockville's newspaper.

Software Downloads

[C-Net Downloads](#) - Various software downloads
[ZDNet Downloads](#) - Various software downloads
[Freeware Files](#) - Thousands of free downloads

Security Tools

[AVG](#) - Free antivirus utility
[Ewido](#) - Free spyware scanner/cleaner
[CCleaner](#) - Clean-up garbage on your system
[AdAware](#) - Free spyware scanner/cleaner

Table Building Crash Course

Until Content Management Systems (CMS) became mainstream, most website were built within an intricate table structure. After a brief look at tables, we will build a table for our Link Page that looks similar to this:

Tables are no different than any other html element. The main thing to remember is that it is constructed of

elements within elements. Of course we begin and end a table with the `<table>` and `</table>` tags, but within these tags we also have to build rows and columns. Looking at our diagram we can see that a table with three columns, and three rows will be built. Our first and last rows have all the columns merged into a single column. The tags we will be using are:

Tag	Name	Description
<code><table></code> <code></table></code>	table outline	Defines the block contained in the tags as a table
<code><tr></tr></code>	table row	defines a row in a table
<code><td></td></code>	table column	defines the columns in a table

Here is a quick explanation of the code (see next page). After we start the table, and define the first row, we also define the first column. In the case of the first row we want to span this column over three columns. I've inserted a `bgcolor` attribute simply to outline the different cells. We then end our column, and then our row, since we are finished with this row. The next row and set of columns have a few more attributes thrown in. First, we locked our column width to 33% of the browser page. Since our first line of the table defined our full table width to be 100%, this column will be 1/3 of the page width. This is not a necessary thing to do, I've done it purely for aesthetics. Next I've told the column to `valign="top"`. By default the contents of this column will align to the vertical center, and again this top-alignment is for aesthetics. And finally, I've used a background colour other than white. The next column was accomplished in the same fashion as the first. The only real difference in all of these columns is the lack of a width attribute in the last one. Since the first two columns allocate 2/3 of 100% width, it's simple math that tells me this column must be the final 1/3 - therefore, no need to define it.

Play around with this table by removing attributes to see what the difference is. By experimenting with it you will grasp a better understanding.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <title>My Personal Link Page</title>
</head>
<body>
<table width="100%">
  <tr>
    <td colspan="3" bgcolor="#99CCFF"><b><center>My Personal Link
Page</center></b><br/>
    </td>
  </tr>
  <tr>
    <td width="33%" valign="top" bgcolor="#CCFF66"><b>Local
Websites</b><br/>
    <hr>
    <a href="http://www.biccweb.com"> BICC</a> - Computer Club<br/>
    <a href="http://www.mcsbrockville.com"> Micro Computer
Service</a> - Computer Sales/Service<br/>
    <a href="http://www.recorder.ca"> Recorder and Times</a> -
Brockville's newspaper.
    </td>
    <td width="33%" valign="top" bgcolor="#FFCC33"><b>Software
Downloads</b><br/>
    <hr>
    <a href="http://www.download.com/"> C-Net Downloads</a> -
Various software downloads<br/>
    <a href="http://downloads.zdnet.com/"> ZDNet Downloads</a> -
Various software downloads<br/>
    <a href="http://www.freewarefiles.com/"> Freeware Files</a> -
Thousands of free downloads
    </td>
    <td valign="top" bgcolor="#FFFFFF"><b>Security Tools</b><br/>
    <hr>
    <a href="http://www.free.gtisoft.com/"> AVG</a> - Free antivirus
utility<br/>
    <a href="http://www.free.gtisoft.com/"> Ewido</a> - Free spyware
scanner/cleaner<br/>
    <a href="http://ccleaner.com/"> CCleaner</a> - Clean-up garbage
on your system<br/>
    <a href="http://lavasoftusa.com/"> AdAware</a> - Free spyware
scanner/cleaner
    </td>
  </tr>
  <tr>
    <td colspan="3" bgcolor="#99CCFF"><i><center>(c)2006 my personal
website inc.</center></i>
    </td>
  </tr>
</table>
</body>
</html>

```



<http://www.mcsbrockville.com/bicc/foto107.gif>
 66 x 66 animated gif
 8-bit, 256 colour
 72 dpi



<http://www.mcsbrockville.com/bicc/jacklantern.jpg>
 174 x 189 jpeg
 24-bit colour
 72 dpi






<http://www.mcsbrockville.com/bicc/wlmeter.gif>
 145 x 105 animated gif
 8-bit, 256 colour
 72 dpi

Images available to experiment with

HTML source code for the Table Demonstration

HTML Cheat Sheet

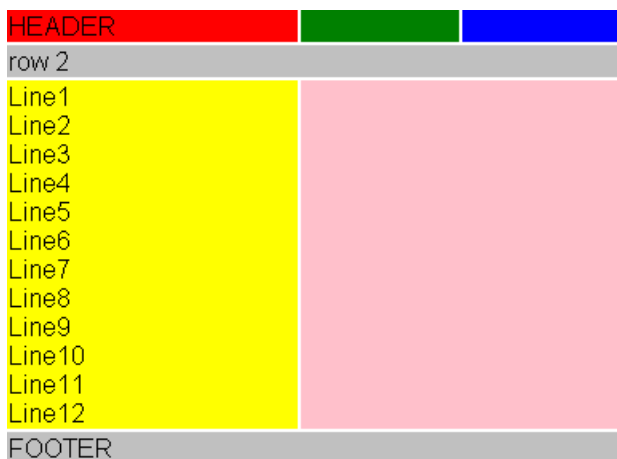
Syntax	Meaning	Example
Basic Tags		
<html></html>	Creates an HTML document	
<head></head>	Sets off the title and other information that isn't displayed on the Web page itself	
<body></body>	Sets off the visible portion of the document	
Header Tags		
<title></title>	Puts the name of the document in the title bar	
Body Attributes		
<body bgcolor=x>	Sets the background color, using name or hex value	for a red background: <body bgcolor="red"> or <body bgcolor="#FF0000">
<body text=x>	Sets the text color, using name or hex value	for blue text <body text="blue"> <body text="#0000FF">
<body link=x>	Sets the color of links, using name or hex value	display links as green text <body link="green"> <body link="#00FF00">
<body vlink=x>	Sets the color of visited links, using name or hex value	display visited links as pink <body link="pink"> <body link="#FF00FF">
<body alink=x>	Sets the color of links on click	display the active link as orange <body alink="orange"> <body alink="FF8040">
Text Tags		
<pre></pre>	all the spaces and carriage returns are rendered exactly as you type them.	<pre><PRE> Position Phone John Webmaster 924-1234 Don Secretary 342-1234 Gerald Treasurer 342-4321 </PRE></pre> <p>Would render text exactly as this on a web page.</p>
<h1></h1> .. <h6></h6>	creates headlined text. h1 is the largest text, and h6 is the smallest.	<pre><h1>BIG</h1> BIG <h3>Medium</h3> Medium</pre>
 or 	creates bold text	
<i></i> or 	creates italicized text	
<tt></tt>	Creates teletype, or typewriter-style text	
	Sets size of font, from 1 to 7	
	Sets font color, using name or hex value	using these two, create medium sized, red text: Red Med. Text
Links		
	creates a hyperlink	<pre>MCS home page ... results in: MCS home page</pre>
	creates a mailto link	<pre>Click here to email John ... results in: Click <u>here</u> to email John</pre>

<code></code>	creates a target location within a document	<code>TOP of PAGE ... defines a place in the document, mostly used with the following tag</code>
<code></code>	Links to the target <i>name</i> from elsewhere in the document	used with previous tag: click <code>HERE</code> for Top ... results in: click <u>HERE</u> for Top
Formatting		
<code><p></p></code>	creates a new paragraph	
<code><p align=x></p></code>	aligns the text between the <code><p></code> tags to the left, right or center	<code><p align="right">Don't Look</p></code> ... results in: <p style="text-align: right;">Don't Look</p>
<code>
</code>	inserts a line break	
<code><blockquote></blockquote></code>	indents text from both sides of screen	
Lists		
<code><dl></dl></code>	creates a definition list	<pre><dl> <dt>Memory</dt> <dd>Where data is temporarily held</dd> <dd>may be SDRAM or DDR</dd> <dt>CPU</dt> <dd>The brains of your computer</dd> <dt>Hard Drive</dt> <dd>long-term storage</dd> </dl></pre> <p>would produce:</p> <pre>Memory Where your data is temporarily held may be SDRAM or DDR CPU The brains of your computer Hard Drive long-term storage</pre>
<code></code>	creates an ordered list	used with the following tags, created a numbered list
<code></code>	wraps each item in the list, and creates a numbered list when used with <code></code>	
<code></code>	identical to <code></code> , except each item is shown with a bullet instead of a number	
Graphic Elements		
<code></code>	adds an image	<code> ... would produce</code> 
<code></code>	aligns an image relative to the current cursor position. Options are, left, right, center, top, bottom and middle	<code> ... would produce</code> 
<code></code>	sets the size of the border around the image, use "0" for no border	
<code><hr></code>	inserts a horizontal line	attributes for this are, width="x" , with x being a whole number or a percentage of the screen width. size="x" , where x would be a number noshade , would result in no shadow on the line
many of the elements can be used with other tags, For example, to use an image to click on for a link, instead of text, do the following:	Click here <code></code> for MCS ... would result in: Click here  for MCS	

Tables		
<table></table>	creates a table	
<tr></tr>	contains each row in a table	
<td></td>	sets off each cell in a table	
<th></th>	table header, usually a normal cell with bold, centered text	
Table Attributes		
<table border="x">	sets the width of the border around table cells	<table border="2"> ... would result in a border 2 pixels thick around table cells.
<table cellspacing="x">	sets the amount of space between table cells	
<table cellpadding="x">	sets the amount of space between the border of a cell and its contents	
<table width="x">	sets the table width in pixels, or as a percentage of the document width.	<table width="75%"> ... would result in a table 75% the width of the web page. This adds a dynamic flair to a website, as it appears to resize with different resolutions.
<tr align="x"> or <td align="x">	determines the horizontal alignment for the cells, either left , center or right .	
<tr valign="x"> or <td valign="x">	determines the vertical alignment for the cells, either top , middle or bottom .	
<td colspan="x">	determines the number of columns a cell should span (default value=1)	
<td rowspan="x">	determines the number of rows a cell should span (default=1).	
<td nowrap>	prevents the lines within a cell from being broken to fit.	

Advanced Table Building

Since tables are used for structuring websites when a CMS package isn't used, and there is no CSS involved, we will look at building a structured table similar to the diagram below.



You will notice that this is a table with 4 rows and 3 columns. The 2nd row spans 3 columns, the last row also spans 3 columns. The large area on the right is actually two columns spanned together to appear as one large block. Type the following HTML code into a text editor and then load the saved *.HTML file into a web browser. Play with the code and become familiar with table functions.

```
<html>
<head></head>
<body bgcolor="magenta">
<table width="80%">
  <tr rowspan="2">
    <td width="50%" bgcolor="red">Header</td>
    <td bgcolor="green"></td>
    <td bgcolor="blue"></td>
  </tr>
  <tr>
    <td colspan="3" bgcolor="silver">row 2</td>
  </tr>
  <tr>
    <td width="50%" bgcolor="yellow">
      Line1<br/>
      Line2<br/>
      Line3<br/>
      Line4<br/>
      Line5<br/>
      Line6<br/>
      Line7<br/>
      Line8<br/>
      Line9<br/>
      Line10<br/>
      Line11<br/>
      Line12<br/>
    </td>
    <td colspan="2" bgcolor="pink"></td>
  </tr>
  <tr>
    <td colspan="3" bgcolor="silver">FOOTER</td>
  </tr>
</table>
</body>
</html>
```