HTML Tutorial

This tutorial provides an overview of the HTML language and shows you how to use simple methods to take control of your web pages. The tutorial is designed for people with a basic understanding of how websites work (if this isn't you, take some of our beginner internet tutorials).

If you're not sure why you should make the effort, see why learn HTML?

What is HTML?

HTML is a computer language. It is the foundation of the World Wide Web and forms the basis of most web pages.

It's important to note that HTML is not a programming language. Technically it is known as a "markup" language, hence the name **H**yper**T**ext **M**arkup **L**anguage. The reason this is significant is that programming languages are quite complex and require a great deal of effort to master, wheras HTML is not. HTML is a very simple language and is quite easy to learn. In addition, you can learn small parts of the language and still get a lot of benefit. You don't need to know the whole language at all, although obviously the more you know the better.

Although new languages and technologies will probably supercede HTML eventually, it is still very important to learn HTML. It will continue to be a critical component of the web for a very long time.

HTML Basics

A web page is an "HTML Document". This is a file format which usually uses the extension ".html" or ".htm". For example, if you use Microsoft Word, you will usually save your files with the extension ".doc". However you can also save your files with many other extensions such as ".txt", ".wps" etc. Amongst the options is ".html".

HTML documents are actually just plain text, but contain snippets of code which carry vital information about how the page should be displayed. You can create such a document using any text editor - even a very simple one like Windows Notepad. In fact many web designers prefer to use simple text editors.

This is what a very simple HTML document looks like:

<html>
< head>
< title>A Simple Web Page</title>
< /head>
< body>
This is about as simple as a web page can get.
< /body>
< /html>

To view an HTML document, you must use a browser (or similar software). The browser opens the HTML document in the background and "decodes" it before showing it to you. What you see is your browser's interpretation of how the web page should look (note: this is actually an important point - it's why you should test your site using a variety of browsers).

Tags and Page Structure

HTML uses a system of "tags" to record instructions on how the page should be displayed. Tags are placed inside angle brackets like so: <instruction>
Tags often exist in pairs: The "beginning tag" specifies the instruction and is placed where you want the instruction to begin. The "end tag" is placed where you want the instruction to end, and is identified by the inclusion of a forward slash like so: </instruction>. The instruction is applied to everything between the two tags.

The most important tags are <html> and </html> - the entire document is contained within these two tags. The instruction here is simply "This is an HTML document".

Within the document, there are two parts: the "head" and the "body". The head is contained within the tags <head> and </head>, the body is contained within the tags <body> and </body>. The head includes information about the document, and is not displayed by the browser. The body contains all the contents for the page, and is what the browser displays.

Note: The original specification for HTML allowed you to use either upper or lower case letters in tags. Although this still applies, the latest specifications say you should use lower case only.

Confused yet? Never mind, it becomes clearer soon...

Let's have another look at our example HTML document. To make things easier to see we've colored the HTML tags blue and separated the head and body with an empty line (empty space between tags is ignored). The tags also don't have to be on new lines, in fact the entire document could all be on one line - it would still be displayed the same.

<html>

< head>
< title>A Simple Web Page</title>
< /head>

< body>
This is about as simple as a web page can get.
</body>

< /html>

The first and last tags identify this as an HTML document, so your browser or other software knows what to do with it. The head contains the "title" tags which identify the name of this document (we'll talk more about that later). The body contains one line of text, which is what you see when you open this document in a browser. To see what this example document looks like, click here, then click your browser's back button to return and continue.

More Tags

Although the example above is a valid web page, it's not very exciting. To make the page more interesting it needs to be formatted. To arrange the text and other elements on your page, there is an array of HTML tags available. The rest of this page includes a quick look at a few important tags, then we'll get into more detail on the following pages. Here are a few to get started:

- : Beginning and end of a paragraph (puts a space between paragraphs).
-
 : Single line break.
- : Bold text.
- : Italics.
- : Centre everything between these tags (note the American spelling).
- <hr> : Horizontal line.

Let's look at a slightly more advanced example (click here to see how this document looks):

<html>
< head>
< title>A Simple Web Page</title>
</head>
< body>
< center>
< p>A Simple Web Page
< p>This is still a simple web page, but it has a bit more to it.
< /center>
< /body>
< /html>

Hyperlinks

There is one very important tag you'll need to be familiar with: the *hyperlink tag*. It looks like this:

Click here

This example creates a hyperlink to a file called "mypage.html". The text inside the tags is the part which becomes the clickable hyperlink.

Images

Here's a very important thing to understand: HTML doesn't actually contain any graphical content. Instead, graphic files are created and stored separately (we'll explain it more later). Image tags look like this:

<img src="myimage.jpg">

Note: Image files must be in one of two formats: ".gif" or ".jpg".

Summary

To finish this section, let's see an example which uses all the tags we've covered (click here to see how this looks):

<html>

< head>
< title>A Simple Web Page</title>
< /head>

< body>
< center>
< p>It's nice to be able to add bold text and italics.</p>
< a href="webdesign04.html">Click here to return to the tutorial.
< img src="logo.gif">
< /center>
< hr>
< /body>

< /html>

Now here's the good news: If you can understand the example above, you have mastered the concept of creating websites. Everything else just builds on this system. Even if you never manually write a single line of HTML, the knowledge of how it works will be invaluable.

Before we move on, here's a trick: Browsers have a function which enables you to view the HTML source of any page you visit. From your browser's menu select "View" then "Source" (these options may have slightly different names depending on your browser). Voila! The HTML code which makes up the page is revealed.

Next Page: Hyperlinks

Hyperlinks

Hyperlinks are created with an "href" tag (**h**yperlink **ref**erence). In it's simplest form the tag looks like this:

Go To Page 1

In this example, the text "Go To Page 1" becomes a hyperlink to a page called "page1.html". The link looks like this:
Go To Page 1

There are two distinct types of hyperlink: "absolute" and "relative". Absolute links are quite simple, but relative links take a bit of getting used to. Relative links are actually divided into two further categories: "document-relative" and "site-root-relative".

Absolute Links

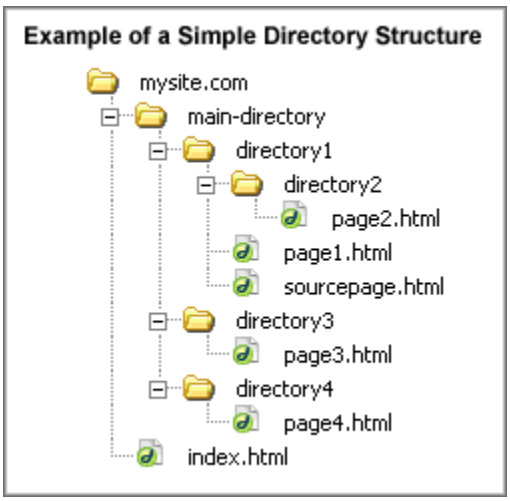
An absolute link (or "absolute URL link") contains a complete internet address, just the same as if you typed the address into your browser's address bar. It doesn't matter where the source page is in relation to the destination page, the link will always work as long as the destination page exists. Most absolute links begin with "http://" and follow this simple format:

Go To My Site

Document-Relative Links

Document-relative links use directions from the source page to the destination page, sort of like saying "Start here then follow this path until you reach the destination".

There are three ways to do this - the method you use will depend on the location of the source and destination pages in relation to each other. We will use the example on the right and create hyperlinks from the page called *sourcepage.html* to other pages in the site.



(1) If the destination page is in the same directory as the source page
All you need to specify is the source file name:

```
<a href="page1.html">Go To Page 1</a>
```

(2) If the destination page is in a folder inside the source page's folder
Specify the folder name and then the file name:

```
<a href="directory2/page2.html">Go To Page 2</a>
```

(3) If the destination page is in a folder outside the source page's folder
Use the special instruction ../ which signifies "one directory higher".

The following link means "Go up one directory level, then go to a folder called *directory3*, then to a file called *page3.html*":

```
<a href="../directory3/page3.html">Go To Page 3</a>
```

You can repeat the ../ to make the link more than one level higher, for example, "Go up two levels, to a file called *index.html*":

```
<a href="../../index.html">Go To The Index Page</a>
```

Important Note: For relative links to work, you must keep the file structure intact. For example, if you moved the *sourcepage.html* file down into the *directory2* folder, the relative links would no longer work. In this case you would need to add another ../ to point the links to the correct level. For example:

```
<a href="../../../index.html">Go To The Index Page</a>
```

Site-Root-Relative Links

Site-root-relative links use a single forward-slash / to signify the instruction: "Start at the document root of the site and go down the directory path from there." The format is:

Go To Page 4

Despite the cumbersome name, this method is quite simple. The link begins at the same level as your domain and works down from there. The link above is the same as:

Go To Page 4

This method has the significant advantage of staying intact if you move the source page. The exact same link will work from any page anywhere in the site.

Notes:

- This method will not work on files on your own hard drive unless you run them through a personal web server. However once you upload the pages to your server they will work.
- If your site does not have it's own domain name, this method is likely to cause problems. Consult your hosting provider for more information.

Targets

You can specify a target window or frame for a hyperlink. This is where the linked page will open. If no target is specified, the link will open in the same window/frame. The format is this (where the target is "self"):

Go To Page 1

The target window options are:
_self : Opens in the same window and same frame.
_top : Opens in the same window, taking the full window if there is more than one frame.
_parent : Opens in the parent frame.
_blank : Opens in a new window.

If you have a frameset defined, you can set your target as any particular frame. For example, if you have a navigation frame called"nav" and a main frame called "main", set the target of your navigation links like so:

Go To Page 1
Go To Page 2
...etc.

Note: Instead of specifying a target for each individual link, you can define a "base target" for an entire frame or page. This means that every link on the page will use the base target by default. Place the base target tag in the page's head. Example:

<base target="_blank">

Link to a Specific Part of a Page (Internal Hyperlink)

An internal hyperlink is a link which points to a particular part of a page. This can be useful in long pages with lots of sub-sections. For example, the links at the top of this page are internal links pointing to each sub-heading.

Internal hyperlinks require an anchor tag with the "name" attribute like so:

Part 2

Create an anchor like this at the place in the page you want to link to. It doesn't matter if there is anything between the open and close tags.

Then create a hyperlink which refers to this anchor with a hash (#) like so:

Go To Part 2

This assumes that the name anchor is on the same page as the hyperlink. To link to an anchor on a separate destination page, simply create a normal link with the anchor name appended to the file name like so:

Go To Page 1, Part 2

Notes:

- Browsers treat internal links exactly the same way as normal links, i.e. as if they were a separate page. This can result in some confusion with back and forward buttons.
- If an internal hyperlink on the same page doesn't work, try including the page's file name in the hyperlink (as if you were linking to a separate page).
- Sometimes internal hyperlinks don't work in pages on your hard drive unless you are running a personal web server. They will work when you upload them to your site.

E-Mail Links (mailto)

E-Mail links, otherwise known as a "mailto" links, use the following format:

Click Here to Email Me

This type of link is a special case, completely different to those described above. Instead of linking to a place on the internet, clicking this link causes the user's computer to open it's default email program (e.g. Outlook Express, Eudora, etc) and prepare a blank email to the specified address. The user then enters their message and sends the email in the normal way.

If you want to be tricky, you can add a "subject" attribute to the link. This will automatically insert the specified subject line into the email:

Click Here to Email Me

Notes:

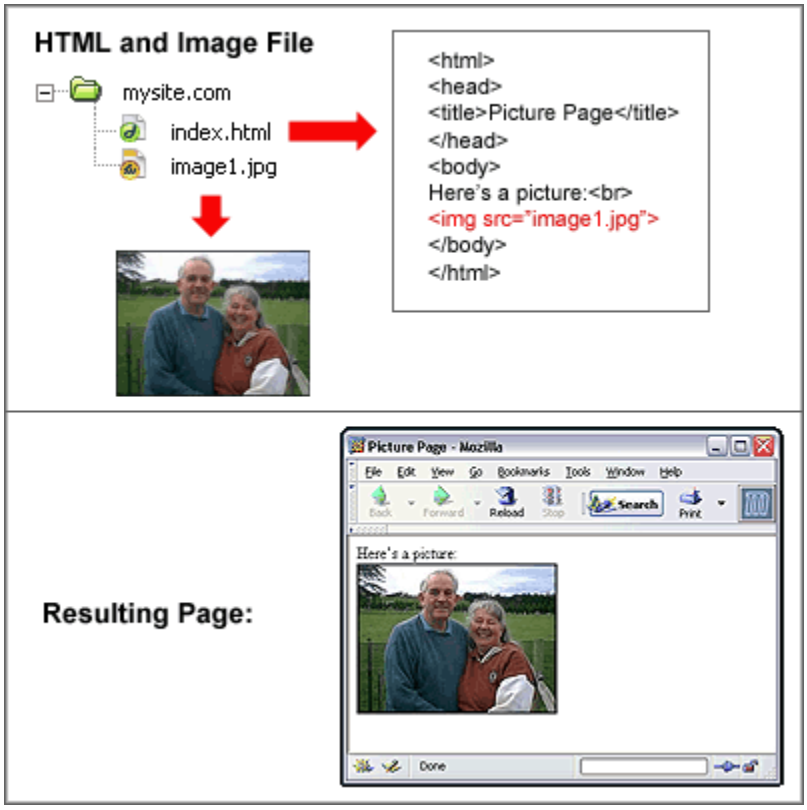
- The subject attribute doesn't work in all browser/email programs. Those programs which don't support it will probably just ignore it, but there are no guarantees that it won't cause an error. Use at your own risk.
- Be aware that using mailto links on your website is an open invitation to spam. Malicious programs called "email harvesters" troll the internet looking for these links - when they find yours they will add it to a database for spammers. This is why we recommend using a contact form rather than a mailto link wherever possible.

The HTML Image Tag

The image tag is used to place an image on the web page. In its most simple form it looks like this:

```
<img src="image1.jpg">
```

The Basics - How it Works



It is very important to understand that images are not technically "part" of the web page file, they are separate files which are inserted into the page when it is viewed by a browser. So a simple web page with one image is actually two files - the HTML file and the image file. The example on the right illustrates this.

In this example the two files are both located in the same folder. The HTML file includes an image tag which refers to *image1.jpg*.

When the HTML file is displayed in a browser, it requests the image file and places it on the page where the tag appears.

As you can see, the most important attribute of the image tag is **src**, which means *source* and tells the browser where the image file is.

Size Attributes

The size attributes define the width and height of the image. They look like this:

```
<img src="image.jpg" width="200" height="150">
```

These attributes are optional but strongly recommended as they help the browser arrange the page more quickly.

Resizing the Image

If the size attributes are set to different values than the original image size, the browser will resize the image to the specified size (this doesn't affect the image file itself, just the way it's displayed in the page). This is a bad idea - the browser will not do a very good job of resizing and there are other complications. In general you should always match the size attributes to the actual size of the image file. If you don't know the image file size, try either of these methods:

- Import the image into any graphics program and select "Image Size" from the menu. This will tell you the height and width.
- Open the image file in a browser, right-click the image, then select "Properties". This will tell you the width and height.

If the image file is the wrong size for your needs, you need to resize it properly using a graphics program. See How to Resize an Image.

Examples

The examples below show how the same 200x150 pixel image is displayed with different size attributes.

<img src="image.jpg" width="200" height="150">

The size attributes match the original image size, so the image is displayed correctly

<img src="image.jpg" width="150" height="150">

The height attribute is correct but the width attribute is less than the image width, making the image appear to be squashed sideways.

<img src="image.jpg" width="150" height="110">

The size attributes are both less than the image dimensions. In this case the displayed image is still in proportion, but may appear to be lower quality due to the browser's imperfect efforts to resize it. Also, the image file is larger than it needs to be.

<img src="image.jpg" width="260" height="194">

The size attributes are both greater than the image dimensions. Again, the displayed image is in proportion but the quality is poor.

Alt & Title Tags

These two tags are very similar and can be confusing. Basically, it makes sense to treat them as the same thing and use them both.

They define a short piece of text which appears instead of the image if the image cannot be displayed (for whatever reason), and/or as a "tool-tip" when you hold your mouse over the image.

If both attributes are specified, the title tag will override the alt tag. Hold your mouse over this image to see which tag your browser displays.

```
<img src="image.jpg" alt="Photo of Mr and Mrs Owen" title="Mr and Mrs Owen">
```

If no alt or title tag is specified, the results vary depending on the browser and user settings. Some will show nothing, some will show the file name.

Spacing Attributes

You can create space between the image and surrounding text by defining vertical and horizontal space like so:

```
<img src="image.jpg" vspace="5" hspace="10">
```

Alignment

You can use the align attribute to position the image:

```
<img src="image.jpg"align="left">
```

The following alignment options are available:
left, right, top, middle, bottom, absmiddle, absbottom, baseline, texttop

Note: These options are not particularly intuitive, that is, you may be surprised at how they actually function. We recommend that you experiment with these standard options and learn exactly how they work. Unfortunately image alignment is somewhat limited in standard HTML. For accurate placement you may need to use more advanced methods than we are covering here.

Border Size

The border attribute places a border around the image. In the following example a 1-pixel border is applied:

```
<img src="image.jpg" border="1">
```

If no border attribute is specified, no border is applied, *except* when the image is used as a hyperlink. In this case a 1-pixel border is applied. If you want to make an image into a hyperlink without a border, specify a zero border like so:

```
<img src="image.jpg" border="0">
```

HTML Tables

Tables are a way to arrange content on a web page. In the early days of the internet, tables were the only practical way to achieve any sort of advanced page layout. With the advent of CSS, tables have become less favoured. However they are still the preferred way to present certain types of information.

Whether or not you use tables for layout, they are an important part of web design and you need to know how they work. The HTML code for tables can be quite confusing at first but don't be put off - with a little practice it becomes quite easy.

The most simple table HTML code looks like this...

```
<table>
<tr><td>Content goes here</td></tr>
</table>
```

... and creates this:

=

Content goes here

As you can see, there are several components which make up a table. Before we explain them, we'll add a common table attribute - the border. This will make it easier to see what's happening. We'll tweak the look of the border later.

```
<table border="1">
<tr><td>Content goes here</td></tr>
</table>
```

=

Content goes here

Now let's break it down. There are three critical tags which make up any table:

<table> </table> Defines the beginning and end of the table.
<tr> </tr> Table Row - Defines the beginning and end of a horizontal row.
<td> </td> Table Cell - Defines an individual cell. Cells are always placed inside a row.

A table must have at least one row and one cell. In theory there is no upper limit to the number of rows and cells - you just need to be realistic about the size of the table on your page.

To add more cells, simply add more <td> tags. The example below has one row, 2 cells.

```
<Table border="1">
<tr><td>Cell 1</td><td>Cell 2</td></tr>
</table>
```

=

Cell 1	Cell 2

To add more rows, add more <tr> tags with their respective cells. The example below has two rows, each containing one cell.

```
<Table border="1">
<tr><td>Row 1</td></tr>
<tr><td>Row 2</td></tr>
</table>
```

=

Row 1
Row 2

Using the same system you can add rows and cells to your heart's content:

```
<table border="1">
<tr><td>1-1</td><td>1-2</td><td>1-3</td></tr>
<tr><td>2-1</td><td>2-2</td><td>2-3</td></tr>
<tr><td>3-1</td><td>3-2</td><td>3-3</td></tr>
</table>
```

=

1-1	1-2	1-3
2-1	2-2	2-3
3-1	3-2	3-3

Rowspan & Colspan

The examples above work fine as long as each row contains the same number of cells. The complications start when rows or columns have different numbers of cells. (*Note:* Although there is no HTML tag for columns, vertical lines of cells are referred to as columns.)

In order to have different numbers of cells you must force a cell to spread out across two or more rows or columns, using the following cell attributes:

colspan - The number of columns the cell spans
rowspan - The number of rows the cell spans

These attributes are shown in the following examples:

```
<table border="1">
<tr><td>1-1</td><td>1-2</td></tr>
<tr><td colspan="2">2-1</td></tr>
</table>
```

=

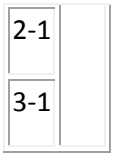
1-1	1-2
2-1	

```
<table border="1">
<tr><td>1-1</td><td rowspan="3">1-2</td></tr>
```

=

1-1	1-2

```
<tr><td>2-1</td></tr>
<tr><td>3-1</td></tr>
</table>
```



This brings us to an important limitation. Cells can only span whole numbers of rows or columns, i.e. cells can't span half a row or part of a column. In fact tables have many such limitations which is partly why they are not the best method for general page layout. However they are very good for content which fits into normal rows and columns.

Note: If your table has an invalid number of cells, for example, 2 cells in the first row and 3 cells in the second row (with no colspan to compensate), the results will be unpredictable and probably very ugly.

If you've made it this far you've done well. The good news is that we've covered all the difficult things about tables. From here on it's all about tweaking and making the table look neater.

Width and Height

The size of a table can be defined in two ways - as a fixed pixel value or as a percentage. If no percentage sign is used then the value is taken to mean pixels. The following examples are fairly self-explanatory.

```
<table width="90%">
<table width="640" height="300">
```

The same system can be used for individual cells, e.g. <td width="90%">Content</td>

Notes:

- Although the height attribute is widely supported, it isn't actually part of the official HTML standard. It may not always work as you expect.
- If no size values are specified, the browser will decide on an appropriate size. Results will vary between browsers.
- If a table width is wider than the browser window, the page will not fit on the screen and horizontal scroll bars will appear.
- The percentage value applies to whatever "container" the table is in, meaning that if the table is nested inside another table or page element, the width will be a percentage of the containing cell or element. If there is no containing element, the table width will be a percentage of the page width.

Spacing and Padding

There are two ways to set the spacing around cells.

- cellspacing - Defines the space *between* cells. If a border is used, the spacing will widen the border.
- cellpadding - Defines the space *inside* each cell, i.e. the space between the edges of the cell and the content within it.

Here are a few examples:

```
<table border="1" cellpadding="1" cellspacing="0">
<tr><td>1-1</td><td>1-2</td></tr>
<tr><td colspan="2">2-1</td></tr>
</table>
```

=

1-1	1-2
2-1	

```
<table border="1" cellpadding="10" cellspacing="0">
<tr><td>1-1</td><td>1-2</td></tr>
<tr><td colspan="2">2-1</td></tr>
</table>
```

=

1-1	1-2
2-1	

```
<table border="1" cellpadding="10" cellspacing="10">
<tr><td>1-1</td><td>1-2</td></tr>
<tr><td colspan="2">2-1</td></tr>
</table>
```

=

1-1	1-2
2-1	

Note: These attributes apply to the entire table - you can't specify spacing or padding for individual cells.

That's currently the end of this tutorial. We hope to expand and add much more information at a later date. In the meantime return to our internet section for more goodies.

Further references at → http://www.mediacollege.com/internet/