

Beginning HTML Tutorial

Based on HTML Beginner Tutorial from www.htmldog.com – June 15, 2010

Getting Started

HTML files are nothing more than simple text files, so to start writing in HTML, you need nothing more than a simple text editor.

Notepad is a common text editor (on Windows this is usually found under the Programs > Accessories menu).

Type this in to your text editor:

```
This is my first web page
```

Save the file as "myfirstpage.html". It is important that the extension ".html" is specified - some text editors, such as Notepad, will automatically save it as ".txt" otherwise.

Go to the File menu of the browser, select Open, and browse for the file.

Pow. There it is. Your first web page. How exciting! All it took was a few typed words.

Software programs such as Dreamweaver will never give you the same control over a web page as coding by hand. These programs often throw in unnecessary or non-standard code to "help" you. Be careful!

Tags

Although the basics of HTML are plain text, we need a bit more to make it a valid HTML document.

Tags

The basic structure of an HTML document includes **tags**, which surround content and apply meaning to it. Change your document so that it looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html>  
<body>  
    This is my first web page  
</body>  
</html>
```

Now save the document again, go back to the web browser and select "refresh" (which will reload the page).

The appearance of the page will not have changed at all, but the purpose of HTML is to apply meaning, not presentation, and this example has now defined some fundamental elements of a web page.

The first line on the top that starts `<!DOCTYPE...` is to let the browser know that you know what you're doing. You may think that you don't actually know what you're doing yet, but it's important to stick this in. If you don't, browsers will switch into "quirks mode" and act in a very peculiar way. To get back to the point, `<html>` is the **opening tag** that kicks things off and tells the browser that everything between that and the `</html>` **closing tag** is an HTML document. The stuff between `<body>` and `</body>` is the main content of the document that will appear in the browser window.

Closing tags

The `</body>` and `</html>` close their respective tags. ALL HTML tags should be closed. Although older versions of HTML allowed some tags not to be closed, latest standards require all tags to be closed.

Not all tags have closing tags like this (`<html></html>`) some tags, which do not wrap around content will close themselves. The line-break tag for example, looks like this : `
`. We will come across these examples later. All you need to remember is that all tags must be closed and most (those with content between them) are in the format of opening tag → content → closing tag.

Page Titles

All HTML pages should have a page **title**.

To add a title to your page, change your code so that it looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html>

<head>
  <title>My first web page</title>
</head>

<body>
  This is my first web page
</body>

</html>
```

We have added two new elements here, that start with the **head** tag and the **title** tag (and see how both of these close).

The head element (that which starts with the `<head>` opening tag and ends with the `</head>` tag) appears before the body element (starting with `<body>` and ending with `</body>`) and contains information *about* the page. The information in the head element does not appear in the browser window.

We will see later on that other elements can appear inside the head element, but the most important of them is the **title** element.

If you look at this document in the browser (save and refresh as before), you will see that "My first web page" will appear on the title bar of the window (not the actual canvas area). The text that you put in between the title tags has become the title of the document. If you were to add this page to your "Favorites" or "Bookmarks", depending on your browser, you would see that the title is also used there.

Paragraphs

Now that you have the basic structure of an HTML document, you can mess about with the content a bit.

Go back to your text editor and add another line to your page:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html>

<head>
  <title>My first web page</title>
</head>

<body>
  This is my first web page
  How exciting
</body>

</html>
```

Look at the document in your browser.

You might have expected your document to appear as you typed it, on two lines, but instead you should see something like:

This is my first web page How exciting.

This is because web browsers don't usually take any notice of what line your code is on. It also doesn't take any notice of spaces (you would get the same result if you typed "This is my first web page How exciting").

If you want text to appear on different lines, you need to tell the browser.

Change your two lines of content so that they look like this:

```
<p>This is my first web page</p>
```

```
<p>How exciting</p>
```

The **p** tag is for **paragraph**.

Look at the results of this. The two lines will now appear on two lines.

Emphasis

You can emphasise text in a paragraph using **em** (emphasis) and **strong** (strong emphasis). Browsers display **em** in italics and **strong** in bold. You can also use **** for bold and **<i>** for italics.

```
<p>Yes, that <em>is</em> what I said. How <strong>very</strong> exciting.</p>
```

Line breaks

The line-break tag can also be used to separate lines like this:

```
This is my first web page<br />  
How exciting
```

However, this method shouldn't be used if two blocks of text are intended to be separate from one another. If that's what you want to do, you probably want the **p** tag.

Note that because there's no content involved with the line-break tag, there is no closing tag and it closes itself with a "/" after the "br".

Headings

The **p** tag is just the start of text formatting. If you have documents with **headings**, then there are HTML tags specifically designed just for them.

They are **h1**, **h2**, **h3**, **h4**, **h5** and **h6**, **h1** being the larger and **h6** being the smallest.

Change your code to the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
  
<html>  
  
<head>  
  <title>My first web page</title>  
</head>
```

```
<body>
  <h1>My first web page</h1>

  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>

  <h2>Why this is</h2>
  <p>To learn HTML</p>
</body>

</html>
```

Note that the `h1` tag is only used once - it is supposed to be the main heading of the page and shouldn't be used multiple times.

`h2` to `h6`, can be used as often as you desire, but they should always be used in order. For example, an `h4` should be a sub-heading of an `h3`, which should be a sub-heading of an `h2`.

Lists

There are three types of lists; **unordered lists**, **ordered lists** and **definition lists**. We will look at the first two here.

Unordered lists and ordered lists work the same way, except that unordered lists are used for non-sequential lists with items usually preceded by bullets. Ordered lists are for sequential lists, which are normally represented by incremental numbers.

The `ul` tag is used to define unordered lists and the `ol` tag is used to define ordered lists. Inside the lists, the `li` tag is used to define each list item.

Change your code to the following:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html>

<head>
  <title>My first web page</title>
</head>

<body>
  <h1>My first web page</h1>

  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>
```

```

<h2>Why this is</h2>
<ul>
  <li>To learn HTML</li>
  <li>To show off</li>
  <li>Because I want to make millions of dollars</li>
</ul>

</body>

</html>

```

If you look at this in your browser, you will see a bulleted list. Simply change the `ul` tags to `ol` and you will see that the list will become numbered.

Lists can also be included in lists.

Replace the above list code with the following:

```

<ul>
  <li>To learn HTML</li>
  <li>
    To show off
    <ol>
      <li>To my boss</li>
      <li>To my friends</li>
      <li>To my cat</li>
    </ol>
  </li>
  <li>Because I want to make millions of dollars.</li>
</ul>

```

There you have it - a list within a list. You could put another list within that, and another within that, and so on.

Links

So far you've been making a stand-alone web page, which is nice, but what makes the internet so special is that it all **links** together.

The 'H' and 'T' in 'HTML' stand for '**hypertext**', which basically means a system of linked text.

An **anchor** tag (`a`) is used to define a link, but you also need to add something to the anchor tag - the **destination** of the link.

Add this to your document:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html>

<head>
  <title>My first web page</title>
</head>

<body>

  <h1>My first web page</h1>

  <h2>What this is</h2>
  <p>A simple page put together using HTML</p>

  <h2>Why this is</h2>
  <p>To learn HTML</p>

  <h2>Where to find the tutorial</h2>
  <p><a href="http://www.htmldog.com">HTML Dog</a></p>

</body>

</html>
```

The destination of the link is defined in the **href** attribute of the tag. The link can be **absolute**, such as "http://www.htmldog.com", or it can be **relative** to the current page.

So if, for example, you had another file called "flyingmoss.html" then the line of code would simply be `The miracle of moss in flight`.

A link does not have to link to another HTML file, it can link to any file anywhere on the web.

A link can also send a user to another part of the same page they are on. You can add an **id** attribute to just about any tag, for example `<h2 id="moss">Moss</h2>`, and then link to it by using something like this: `Go to moss`. Selecting this link will scroll the page straight to the element with that id.

Images

Things might seem a little bland and boring with all of this text formatting. Of course, the web is not just about text, it is multi-media and the most common form of media is the **image**.

The **img** tag is used to put an image in an HTML document and it looks like this:

```

```

The **src** attribute tells the browser where to find the image. Like the **a** tag, this can be absolute, as the above example demonstrates, but is usually relative. For example, if you create your own image and save it as "alienpie.jpg" in a directory called "images" then the code would be ``

A few things about images...

The most commonly used file formats used for images are **GIFs** and **JPEGs**. They are both compressed formats, and have very different uses.

GIFs can have no more than 256 colors, but they maintain the colors of the original image. The lower the number of colors you have in the image, the smaller the file size will be.

GIFs SHOULD BE USED FOR IMAGES WITH SOLID COLORS, LIKE CLIPART.

JPEGs on the other hand use a mathematical algorithm to compress the image and will distort the original slightly. The lower the compression, the higher the file size, but the clearer the image.

JPEGS SHOULD BE USED FOR IMAGES, SUCH AS PHOTOGRAPHS.

Images are some of the largest files a web designer will be handling. It is a common mistake to not take into account the file size of images, which can be extremely large. Web pages should download as quickly as possible. You do not want a large image file to slow down the download time of a full page, discouraging users.

You need to strike a balance between image quality and image size. Most modern image manipulation programs allow you to compress images and the best way to figure out what is best suited for yourself is trial and error.

Tables

There are a number of tags used in tables, and to fully get to grips with how they work is probably the most difficult area of this HTML Beginners Tutorial.

Copy the following code into the body of your document and then we will go through what each tag is doing:

```

<table>
  <tr>
    <td>Row 1, cell 1</td>
    <td>Row 1, cell 2</td>
    <td>Row 1, cell 3</td>
  </tr>
  <tr>
    <td>Row 2, cell 1</td>
    <td>Row 2, cell 2</td>
    <td>Row 2, cell 3</td>
  </tr>
  <tr>
    <td>Row 3, cell 1</td>
    <td>Row 3, cell 2</td>
    <td>Row 3, cell 3</td>
  </tr>
  <tr>
    <td>Row 4, cell 1</td>
    <td>Row 4, cell 2</td>
    <td>Row 4, cell 3</td>
  </tr>
</table>

```

The **table** element defines the table.

The **tr** element defines a table **row**.

The **td** element defines a **data cell**. These must be enclosed in **tr** tags, as shown above.

If you imagine a 3x4 table, which is 12 cells, there should be four **tr** elements to define the rows and three **td** elements within each of the rows, making a total of 12 **td** elements.

Putting It All Together

If you have gone through this entire tutorial then you should be a competent HTMLer.

In fact, due to the fact that most people who use HTML use it rather badly, you should be better than most.

The following code incorporates all of the methods that have been explained in the previous pages:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html>

<head>

```

```
<title>My first web page</title>
```

```
<!-- By the way, this is a comment -->
```

```
</head>
```

```
<body>
```

```
<h1>My first web page</h1>
```

```
<h2>What this is</h2>
```

```
<p>A simple page put together using HTML. <strong>A simple page put together using HTML.</strong> A simple page put together using HTML. A simple page put together using HTML.</p>
```

```
<h2>Why this is</h2>
```

```
<ul>
```

```
  <li>To learn HTML</li>
```

```
  <li>
```

```
    To show off
```

```
    <ol>
```

```
      <li>To my boss</li>
```

```
      <li>To my friends</li>
```

```
      <li>To my cat</li>
```

```
    </ol>
```

```
  </li>
```

```
  <li>Because I want to make millions of dollars.</li>
```

```
</ul>
```

```
<h2>Where to find the tutorial</h2>
```

```
<p><a href="http://www.htmldog.com"></a></p>
```

```
<h3>Some random table</h3>
```

```
<table border="1">
```

```
  <tr>
```

```
    <td>Row 1, cell 1</td>
```

```
    <td>Row 1, cell 2</td>
```

```
    <td>Row 1, cell 3</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Row 2, cell 1</td>
```

```
    <td>Row 2, cell 2</td>
```

```
        <td>Row 2, cell 3</td>
    </tr>
    <tr>
        <td>Row 3, cell 1</td>
        <td>Row 3, cell 2</td>
        <td>Row 3, cell 3</td>
    </tr>
    <tr>
        <td>Row 4, cell 1</td>
        <td>Row 4, cell 2</td>
        <td>Row 4, cell 3</td>
    </tr>
</table>

</body>

</html>
```

There you have it. Save the file and play around with it - this is the best way to understand how everything works. Have fun!

When you're ready, you can check out these other resources to learn more:

- <http://www.htmldog.com/guides/cssbeginner/>
- <http://www.w3schools.com/> - Everything from HTML to CSS and JavaScript.
- <http://www.echoecho.com/html.htm> - Want to know how to do something specific? Check here!

Add a link to your site to the HTML coded pages you've made and don't forget to upload them!