

The HTML Mini-Course

Mike Hayes
<http://www.the-html-book.com>

Introduction

Thank you for asking me for your own copy of the *HTML Mini-Course*.

Well done if you have decided to start coding your own web-page. I promise you - IT IS EASY!

This book is choc-a-bloc with many valuable tips to help you on your way. They are taken from ***The HTML Course Package*** which contains not only the book but eight videos.

Just to clarify one thing before we start, I am often asked which HTML editor I use. Although I have Dreamweaver on my PC and have in fact used it a few times, my *main* tool is good, old MS Notepad. And I recommend that you use it as well. You get used to getting the coding right. When you are accomplished in coding then you can always consider getting a sophisticated system.

Good luck.

Mike Hayes
Amsterdam

Starting up

What is an HTML document? It is a collection of instructions, written according to pre-defined rules, which instruct the browser how the contents of the document must be displayed. We're going to look at some of these instructions and rules.

We'll see the words TAG and ATTRIBUTE a great deal. What are they? A TAG tells the browser what to expect, text manipulation, link to somewhere, image download, etc. A TAG has ATTRIBUTES which supply the data to the browser as to how the TAG must be applied. And finally, ATTRIBUTES have different values that are the parameters that the browser need in order to complete the actions "ordered" by the TAG.

A simple HTML document which contains the compulsory TAGs could be; (The <> brackets have been replaced with <> to make the HTML visible.

<!-- These characters allow you to write comments which are not visible in the document **-->**

`<HTML><!-- tells your browser that this is an HTML document -->`

`<HEAD><!-- identifies the header section -->`

`<TITLE><!-- is exactly what it says, your document's title -->`

`This is Mike Hayes' HTML First Document!!`

`</TITLE> <!-- All tags have a closing tag. Its the same as the opening tag but with a '/' in front of it.-->`

`</HEAD> <!-- See? End of HEAD section-->`

`<BODY><!-- And this is where the story begins, the BODY section of the document -->`

`Yes! This is Mike Hayes' First Ever HTML Document!!`

`</BODY> <!-- End of BODY -->`

`</HTML> <!-- End of HTML document -->`

Of course, we don't need to write so many comments in a document so let's look at it without them.

`<HTML>`

`<HEAD>`

`<TITLE>`

`This is Mike Hayes' HTML First Document!!`

`</TITLE>`

`<HEAD>`

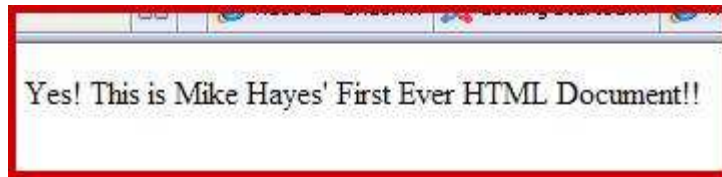
`<BODY>`

`Yes! This is Mike Hayes' First Ever HTML Document!!`

`</BODY>`

`</HTML>`

And what you see on the screen is



Let's have a look at a couple of <BODY> tag attributes that you could use, BGCOLOR and TEXT - for example <BODY BGCOLOR="blue" TEXT="yellow"> would give you a blue screen with all the text yellow. There are other ways to adjust these factors but they fall outside the scope of this article.

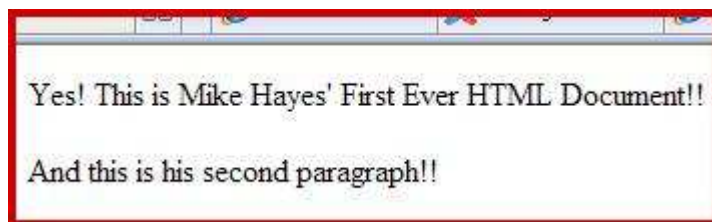
With text we have a great deal of freedom as to size, sort, colour, font, etc. What we must remember is that we want visitors to our site to be able to read it easily and to understand it; i.e. it must be easy on the eye and the brain.

Let's start with paragraphs; the way to organize your text into paragraphs is as follows:

```
<BODY>
Yes! This is Mike Hayes' First Ever HTML Document!!
<P>
And this is his second paragraph!!
</P>
</BODY>
```

The <P> tag starts a new paragraph for you after inserting an empty line, notice the closing </P> tag.

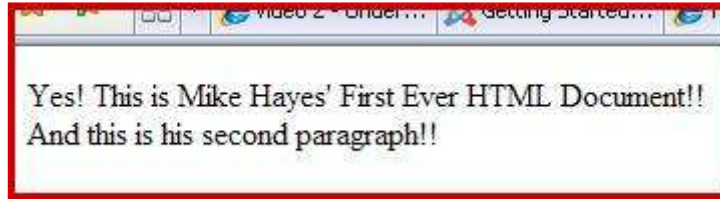
What you see is:



What if you want a line break but no empty line?

```
<BODY>
Yes! This is Mike Hayes' First Ever HTML Document!!
<br />
And this is his second paragraph!!
</BODY>
```

What you see is:



Notice the slash in the `
` tag, what's that doing there? There are some tags which didn't need a closing tag under HTML but XHTML expects closing tags for all tags.

Now the `
` tag was one which didn't need a closing tag therefore there wasn't one. This has been solved by placing the 'closing' slash in the `
` tag itself after a space.

Of course, you may want something a bit more definite than an empty line between our paragraphs. If you use the `<HR />` tag it'll draw a line across the page for you and you can adjust the width, length and colour with its attributes.

There are tags to make your text bold, italicized, big, small and underlined. Let's see how.

There are tags which make your text

bold

italicized

big

small

underlined

You may use any of these tags, for example, `<I><U>` ***in any*** `</U></I>` combination you wish to get your message across but a quick word about the sequence. The sequence is unimportant for the opening tags as long as you reverse it for the closing tags. It's the same rule for all 'nested' tags - FILO - First In Last Out.

One more thing about manipulation. If you don't instruct the browser as to where how the text should be aligned it will always be aligned left but you may want to be more flexible. Have a look at what the `<DIV>` tag does. This is the only use for the `<DIV>` tag.

```
<DIV ALIGN="left">
```

```
You don't have to actually define left</DIV>
```

```
<DIV ALIGN="center">
```

```
Put it in the middle</DIV>
```

```
<DIV ALIGN="right">
```

```
By the right quick march!</DIV>
```

Lists

It is easy just to write your own lists but why do that when HTML offers a couple of excellent solutions. So let's have a look at them.

Here's how a shopping list looks in a bulleted format.

- o Potatoes*
- o Butter*
- o Eggs*

Pretty smart, but how to code it. Let's look.

```
<UL TYPE="circle">  
<LI>Potatoes  
<LI>Butter  
<LI>Eggs  
<LI>Cabbage  
</UL>
```

That's the `` tag, (Unordered List). The TYPE attribute gives the choice of bullet. You have "circle", "disc" and "square" to choose from. Each item is preceded by the `` tag, (List Item). This generates the bullet and a line break.

But you may like to see your list in an orderly fashion, like this.

- A. Potatoes*
- B. Butter*
- C. Eggs*

This is a so-called Ordered List and makes use of the `` tag.

```
<OL TYPE="A">  
<LI>Potatoes  
<LI>Butter  
<LI>Eggs  
</OL>
```

Get the idea? Of course you may be super fussy and like to have lists within lists like this.

- A. Greengrocer*
 - a. Potatoes*
 - b. Cabbage*
 - c. Carrots*
- B. Dairy*
 - I. Butter*
 - II. Eggs*
 - III. Cheese*

As you can see, all is possible as the TYPE attribute gives the choice of the formats A,B,C, etc., a,b,c, etc., 1,2,3, etc., I,II,III, etc. and i,ii,iii, etc.

```
<OL TYPE="A">  
  <LI>Greengrocer  
<OL TYPE="a">  
  <LI>Potatoes  
  <LI>Cabbage  
  <LI>Carrots  
</OL>  
  <LI>Dairy  
<OL TYPE="I">  
  <LI>Butter  
  <LI>Eggs  
  <LI>Cheese  
</OL>
```

It's that easy to make your lists in HTML.

Links

Links are going to connect your website to the World Wide Web and to itself internally, that is to pages in your website or paragraphs on the same page. Let's look at them first.

Internal Links

These are for when you want to send visitors to another part of the page they are viewing at that moment. Here we'll look at the <A> (for address) tag and the NAME and HREF attributes.

Here's a simple example where the visitor is looking for Christian names. They must click on a letter to jump to a list of names, beginning with that letter. These are somewhere else on the same page.

The coding you need to make the jumps work looks like this. Of course, it doesn't have to be a list.

The NAME tag assigns a name to a line of coding on the page, in this case "top".

```
<A NAME="top"></A>  
This is a Christian name finder.  
<OL TYPE="A">  
<LI><A HREF="#A">A names</A>  
<LI><A HREF="#B">B names</A>  
<LI><A HREF="#E">E names</A>  
<LI><A HREF="#etc">etc names</A>  
</OL>
```

The paragraphs you want visitors to link to are coded as follows.

```
<A NAME="A"></A><H3>The A's</H3>  
Adrian  
Alan  
Anthony  
A etcetera  
Back to the <A HREF="#top">top</A>. <!--Click here and you return to "top".-->
```

```
<A NAME="B"></A><H3>The B's</H3>  
Barry  
Boris  
Brian  
B etcetera  
Back to the <A HREF="#top">top</A>.
```

Etc. for C's, D's.

Notice the # in the HREF anchor addresses; this is the indication that it's an internal link. The actual address itself has no # but don't forget, it must have the tag.

External Links

These are the links to other sites and to other pages on your own site.

Here again we'll look at the <A> tag and its attributes HREF and TARGET.

Right; what does this statement tell us?

To visit my website

```
<A HREF="www.my-web-site.com">click here</A>.
```

Simple, click on "[click here](#)" to visit my website. And please notice the closing tag. It's that simple!

As we know all clickable text is coloured blue and is underlined.

A couple of things before we go on. (Note that with attribute data you may use "double quotes", 'single quotes' or no quotes to enclose the parameter. Choose one and stay with it!) Of course you don't have to use text for the click text. There are a couple of other possibilities. You may want to use a banner or photo as link to make the illustration clearer.

Here's the coding needed when you want to use an image instead of text. And by the way, the image can be anywhere, even on another website. The image simply replaces the "*click here*" text.

```
<A HREF="http://www.the-html-book.com">  
<IMG SRC="images/linkhtml.jpg" TARGET="_blank" WIDTH="468"  
HEIGHT="60"></A>
```

If you're sending your visitor to another page or site but don't want to lose him then it's a good idea to include the TARGET attribute in the <A> statement, as above.

Your original page will still be visible with the target page on top. So when they close the target page, your page is still there. The default value is "**_self**" which is the one that causes the new page to overwrite your page in the same window.

Refreshing and redirecting

I want to talk about the HTTP-EQUIV attribute in the <META> statement and this in combination with the DATA and the CONTENT attributes.

When could you use this? When you want a visitor to be redirected to another webpage or site from the one they're originally sent to. You've all seen pages which say,

"This page will redirect to etc. etc. in 10 seconds."

It could come in handy one day.

What does refresh coding look like?

```
<HTML>
<HEAD>
<TITLE>Refresh demo coding.</TITLE>
<META HTTP-EQUIV="Refresh"
  CONTENT="15;URL=http://www.the-job-hunters.book.com">
</HEAD>
<BODY>
<!-- Empty because in principle no-one's going to see it -- >
</BODY>
</HTML>
```

It's quite straight forward, the "15" represents the number of seconds you want the browser to wait before refreshing, and you will realize that if this parameter is "0" then the refresh will happen immediately. The URL is, of course, the target. (Note where the "quotes" are.)

If you wanted to refresh some dynamic information, e.g. share prices, whatever, just refresh the same page as follows.

```
<HTML>
<HEAD>
<META HTTP-EQUIV="refresh" CONTENT="20">
</HEAD>
</HTML>
```

This page will refresh itself every 20 seconds.

Where to?

Where are the pages you want to link to?

When working on your PC the pages you want to link to are most probably all in the same directory. So the coding is:

```
<A HREF="destination-page.html">click here</A>.
```

On the web you might have your site organised in logical directories so 'destination-page.html' might be in a sub-folder 'sales-pages'.

```
<A HREF="sales-pages/destination-page.html">click here</A>.
```

The destination-page may **be** another site or **on** another site.

```
<A HREF="http://www.another-site.com">click here</A>.
```

```
<A HREF="http://www.another-site.com/sales-pages/destination-page.html">click here</A>.
```

If in doubt, always use the full option.

Images

Here we'll be looking at the tag and its attributes SRC, ALIGN, ALT, HEIGHT and WIDTH.

If you intend to have a webpage you'll find that images will be unavoidable. Whether they're photos of your product or graphical images like banners and buttons, you will need them.

I'm going to start with photos. You can digitize them in different ways. Scan them in, download them or use a digital camera. Make sure you've set up a separate directory for your images so you know where to find them.

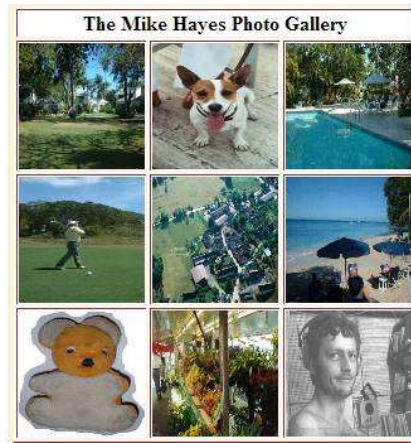
First let's have a look at some random photos.



Lots of photos but presented in a rather messy, unstructured way. Simply programmed like this:

```
<IMG SRC="photo04.jpg" HEIGHT="312" WIDTH="200">  
<IMG SRC="photo05.jpg" HEIGHT="200" WIDTH="300">  
<IMG SRC="photo06.jpg" HEIGHT="225" WIDTH="300">  
<IMG SRC="photo07.jpg" HEIGHT="300" WIDTH="202">  
<IMG SRC="photo08.jpg" HEIGHT="200" WIDTH="300">  
<IMG SRC="photo09.jpg" HEIGHT="230" WIDTH="300">
```

No attempt at an ordered presentation. We could have done it like this. I've added three more snaps.



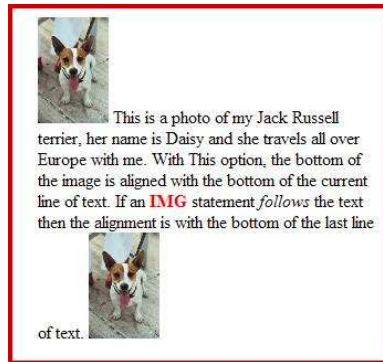
Here the use of a TABLE has given some order.

```
<DIV ALIGN="center">
<TABLE BORDER="5" BORDERCOLORLIGHT="beige"
BORDERCOLORDARK="brown" WIDTH="50%">
<TR>
  <TH COLSPAN="3">The Mike Hayes Photo Gallery
<TR>
<TD><IMG SRC="photo01.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo02.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo03.jpg" WIDTH="100" HEIGHT="100"></TD>
</TR>
<TR>
<TD><IMG SRC="photo04.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo05.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo06.jpg" WIDTH="100" HEIGHT="100"></TD>
</TR>
<TR>
<TD><IMG SRC="photo07.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo08.jpg" WIDTH="100" HEIGHT="100"></TD>
<TD><IMG SRC="photo09.jpg" WIDTH="100" HEIGHT="100"></TD>
</TR>
</TABLE>
</DIV>
```

As you saw in the first show of photo's they are not actually the same size so if we stuff them into this table and make them all the same size, a size which is not proportional for each particular photo, then some distortion does occur.

Let's continue now with the positioning of one image on a page. This series of examples are of the vertical alignment of images in respect to the line of text it's positioned in. The first example shows the ALIGN attribute with the "bottom" option, also the default option.

The HTML Mini-Course



Let's look at the coding

```
<IMG SRC="http://www.my-web-site.com/test/photo02.jpg" WIDTH="64"
HEIGHT="96">
This is a photo of my Jack Russell terrier, her name is Daisy and she travels all
over Europe with me.
With this option, the bottom of the image is aligned with the bottom of the current
line of text.
If an <FONT COLOR="red"><B>IMG</B></FONT> statement <I>follows</I> the
text then the alignment is with the
bottom of the last line of text.
<IMG SRC="http://www.my-web-site.com/test/photo02.jpg" WIDTH="64"
HEIGHT="96">
```

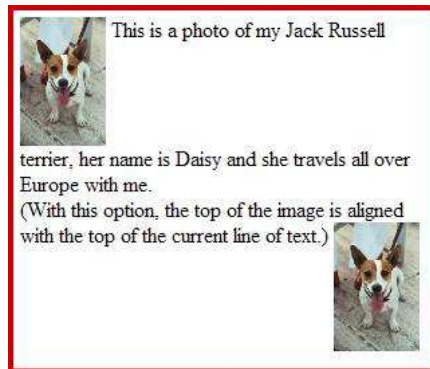
You can manipulate so that the alignment is different. With the "middle" option for example.



Let's look at the coding.

```
<IMG SRC="http://www.my-web-site.com/est/photo02.jpg" WIDTH="64"
HEIGHT="96" ALIGN="middle">
This is a photo of my Jack Russell terrier, her name is Daisy and she travels all
over Europe with me.
(With this option, the middle of the image is aligned with the middle of the current
line of text.)
<IMG SRC="http://www.my-web-site.com/test/photo02.jpg" WIDTH="64"
HEIGHT="96" ALIGN="middle">
```

And last the "top" option.



Let's look at the coding.

```
<IMG SRC="http://www.my-web-site.com/test/photo02.jpg" WIDTH="64"  
HEIGHT="96" ALIGN="top">  
This is a photo of my Jack Russell terrier, her name is Daisy and she travels all  
over Europe with me.  
(With this option, the top of the image is aligned with the top of the current line of  
text.)  
<IMG SRC="http://www.my-web-site.com/test/photo02.jpg" WIDTH="64"  
HEIGHT="96" ALIGN="top">
```

So we see that these methods place the image with regard to the text line that it's embedded in. And that could be anywhere on the page, left, right, middle.

Do you want a BORDER around your image? Well, for images that are also links, the browser will usually put a 2 pixel border round it. You can, of course, manipulate the data for this attribute. But first, let's experiment.

An aspect of images you have to pay attention to is the size as this influences the download time and as we all know, we hate waiting for the images to appear. We can improve this in two ways.

We can make our images very small to speed up the download, but then they'll either be too small to make any impression, or, if we adjust the size with the HEIGHT and WIDTH attributes, their quality will be abominable. Look at these examples.

The HTML Mini-Course



As you see the top image, (size 50 by 37), is unrecognizable when small but when enlarged its dreadful!

So let's see the real image. The first one is the original image; it's 1280 by 960 and is 77kb. I've reduced its display size with the HEIGHT and WIDTH attributes to 500 by 375. The one that follows has been physically reduced by my graphics program to 500 by 375 and takes up 11kb of disc space!



Notice the difference. The first of the two is of a slightly better quality but the second is good enough and it speeds up download time considerably.

Tables

Tables can be as simple or as complicated as you want. They give you a great deal of flexibility when getting your site together.



Here's the HTML for the first flag.

```
<CENTER>
<TABLE>
  <TR> <TH>A Horizontal Flag</TH></TR>
  <TR> <TD BGCOLOR="red" WIDTH="200" HEIGHT="50"></TD></TR>
  <TR> <TD BGCOLOR="white" WIDTH="200" HEIGHT="50"></TD></TR>
  <TR> <TD BGCOLOR="blue" WIDTH="200" HEIGHT="50"></TD></TR>
</TABLE>
</CENTER>
```

Let's look at the tags one for one starting with `<TABLE>` - `</TABLE>`. These two always enclose your table. It's that simple. There are, of course, attributes but not now.

The `<TR>` - `</TR>` tags indicate an individual row within the table and the `<TH>` - `</TH>` and `<TD>` - `</TD>` define the individual cells within that row. Let's look a bit closer.

`<TH>` - `</TH>` are the cells in which you can put the headings. These headings will be centred as bold text automatically.

`<TD>` - `</TD>` are the cells which contain the information you want to display. (These two tags can also have images.)

The `<TH>` - `</TH>` and `<TD>` - `</TD>` tags share the same attributes so let's look at the ones we've used here. First `BGCOLOR` - "background colour" which is really self explanatory. It's one that is used by several other tags as we'll see later. Colour can be defined as above but it may also be represented by the R/B/G hexadecimal equivalent e.g. you can replace "red" with "#FF0000", "blue" with

"#0000FF" and "green" with "#009900". Notice the hash - # - character, don't forget it!

We've used the WIDTH and HEIGHT attributes before so they should be self-explanatory. The value can be a percentage of the table or pixels as I've used here.

Why did I have to define the size? Let's see what the 'flag' looks like without these attributes.

Look at the skinny flag. It's skinny because actually the cells are empty so the minimum is displayed and we have to adjust the size to achieve the image we want.

Right, that was a table with three information *rows* plus a header



```
<CENTER>
<TABLE>
  <TR><TH>A Vertical Flag</TH></TR>
  <TR>
    <TD BGCOLOR="red"  WIDTH="50" HEIGHT="150"></TD>
    <TD BGCOLOR="white" WIDTH="50" HEIGHT="150"></TD>
    <TD BGCOLOR="blue"  WIDTH="50" HEIGHT="150"></TD>
  </TR>
</TABLE>
</CENTER>
```

The main difference here is that instead of three information *rows* we only have one but that one contains three *cells*. Again, to give our flag body we've defined the sizes of the cells. Let's look at the influence of the WIDTH and HEIGHT attributes on what we see. Look at this coding.

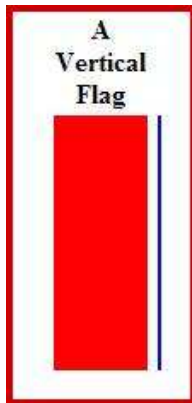
The HTML Mini-Course

```

<CENTER>
<TABLE>
  <TR><TH>A Vertical Flag</TH></TR>
  <TR>
    <TD BGCOLOR="red" WIDTH="50"> </TD>
    <TD BGCOLOR="white" HEIGHT="150"></TD>
    <TD BGCOLOR="blue"> </TD>
  </TR>
</TABLE>
</CENTER>

```

We've given "red" WIDTH, "white" HEIGHT and "blue" neither. And what happens?



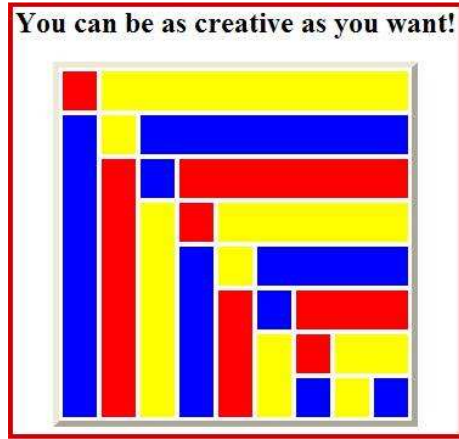
Red's the only one with the WIDTH attribute so only *red's* affected. *White's* the only one with the HEIGHT attribute and that affects them all. *Blue* just follows.

So, you are now aware of the flexibility of the table. Have a look at these two. First a vertical 3D table then a horizontal.

A horizontal 3D table				
The Elvis Presley Story				
The early years		The later years		
Rock'n'Roll		Ballads		
er Hound Dog	Jailhouse Rock	Crying In The Chapel	In The Ghetto	Ted
Reddy Teddy	Rip It Up	Blue Hawai	It's Now Or Never	Me

A vertical 3D table		
My Favorite Wines		
Red Wine	Burgundy	Beaune
		Meursault
	Bordeaux	St. Emilion
		Fronsac
White Wine	Burgundy	Chablis
		Montrachet
		Entre Deux Mers

And there's so much more you can do, look at the last example. A combination of a table and the colours of your choice.



As I said at the beginning of this section, tables can be as simple or as complicated as you want.

Forms

We all know about forms. We see them all over the place. Forms are the method used to collect information. If you want people to, for example, subscribe to your newsletter then a form is what you need. This article does not cover CGI scripts. If you want information just Google 'mail scripts' and you'll find lots of free ones with instructions.

We're going to look at the <FORM> and <INPUT> tags with some of their attributes.

Here's a simple form.

Your Name:	<input type="text"/>
Your E-Mail:	<input type="text"/>
<input type="submit" value="Subscribe Now!"/>	<input type="reset" value="Clear form"/>

Here's the HTML. You'll notice that the form has been placed in a table to proportion and organize it neatly. But we're going to pay attention to the form itself.

```
<TABLE>
<FORM ACTION="http://www.your-domain.com/cgi-bin/formmail.cgi"
METHOD="post">
<TABLE WIDTH="300" BORDER="3" CELSPACING="5" BGCOLOR="#3399CC"
BORDERCOLOR="#000000">
<TR>
<TD WIDTH="50%">Your Name:</TD>
<TD WIDTH="50%">
<INPUT TYPE="text" NAME="fullname" MAXLENGTH="25" SIZE="10"></TD>
</TR>
<TR>
<TD WIDTH="50%">Your E-Mail:</TD>
<TD WIDTH="50%">
<INPUT TYPE="text" NAME="email" MAXLENGTH="25" SIZE="30"></TD>
</TR>
<TR>
<TH WIDTH="50%">
<INPUT TYPE="submit" VALUE="Subscribe Now!" STYLE="background-
color:#FFFF00"></TH>
<TH WIDTH="50%">
<INPUT TYPE="reset" VALUE="Clear form" STYLE="background-
color:#FF6633"></TH>
</TR>
<INPUT TYPE="hidden" NAME="recipient" VALUE="you@your-domain.com">
<INPUT TYPE="hidden" NAME="subject" VALUE="Newsletter">
<INPUT TYPE="hidden" NAME="required" VALUE="fullname,email">
<INPUT TYPE="hidden" NAME="confirmation" VALUE="thanks.html">
<INPUT TYPE="hidden" NAME="ref" VALUE="code">
</FORM >
</TABLE>
```

We start with the <FORM> tag which encloses our form. I'll be handling the ACTION and METHOD attributes at the end.

Next the <INPUT> tag and its NAME and MAXLENGTH attributes. <INPUT> is easy; here we input data in one form or other. The TYPE indicates what sort of input we're dealing with, in this case plain text. The data of the NAME attribute is the title or description we give that data element; in the first line "fullname" and the second "email". Both fields have been given a MAXLENGTH of "25" but different values for SIZE. MAXLENGTH is the maximum number of characters which may be entered; SIZE is the physical size of the field that you see. For the rest of the forms in this section I have just used the former attribute.

The TYPE sorts "submit" and "reset" generate our buttons at the bottom of the form allowing people to submit or clear the form if they change their minds. The VALUE attribute, which we see for the first time, lets us define the text we want on the buttons.

The TYPE "hidden" is an interesting one. It lets us define certain parameters for our form that won't be visible to the reader but give parameters through to the web-side script.

A word of warning!

By clicking on the View menu and then the Source option, visitors to your webpage can see your HTML coding including the "hidden" elements. If one of these a confidential download site, for example, you may not want people to visit it until they've subscribed or maybe bought something. The answer is to encrypt your page with one of the many systems available on the web.

Let's look at the different VALUE's we've used here as "hidden" elements.

NAME="recipient" VALUE="you@your-domain.com"

This informs your server-side script where the form information must be sent to. Don't worry about scripts now, I'll be touching on them later with the ACTION and METHOD attributes.

NAME="subject" VALUE="Newsletter"

When you receive the form information the email subject will be, in this case, "Newsletter"

NAME="required" VALUE="fullname,email"

This informs your system that the named fields are compulsory, if either isn't filled in then the person is rerouted to "whoops.htm" (see below)

NAME="confirmation" VALUE="thanks.html"

This is where your visitor is taken after successfully submitting the form

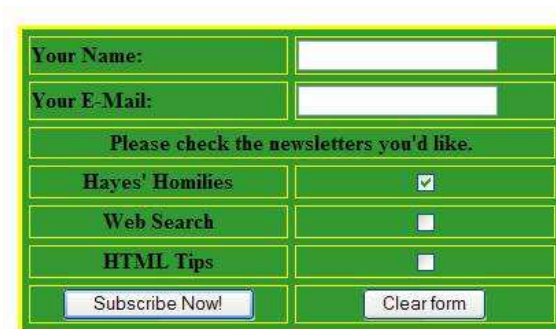
NAME="ref" VALUE="code"

Lets you have the form send an extra piece of information for, for example, marketing purposes.

Last but not least, I included a CSS instruction in this form to colour the two buttons. That's the STYLE attribute. Let's you freshen up your form.

Right, such a lot of information just for a simple form with only two fields but they're the majority of things to be considered with a form.

There are lots more possibilities, like this one for example.



Now we're going to look at the <FORM> tag with its, ACTION and METHOD attributes. I'm going to start with the METHOD attribute. We usually use the "post" option, this results in the contents of the form being emailed to, in this case, me. The other option, "get" results in the form's contents being added to a URL in order to, for example, do a search via a search engine. So for us it's the "post" option.

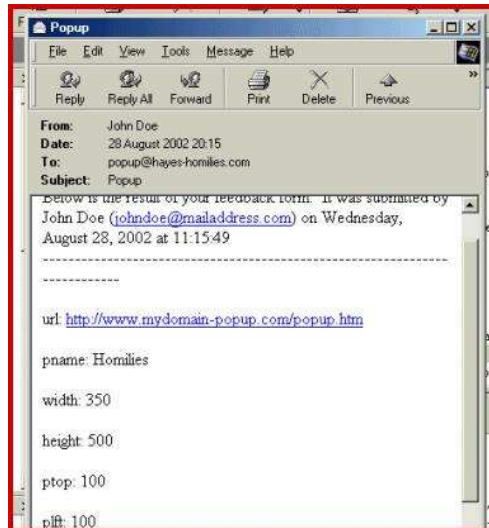
We can fill METHOD in, in two ways; the first way sends the form's contents to a script somewhere in order to be unravelled, the second just emails the raw data to our given email address.

Let's look at that first one.

```
FORM ACTION="http://www.your-domain.com/cgi-bin/formmail.cgi"  
METHOD="post"
```

Via this definition the contents of our form will first be sent to the CGI script *formmail.cgi* in the **cgi-bin** in our website *www.your-domain.com* where it will be 'unravelled' and sent to us in a readable form.

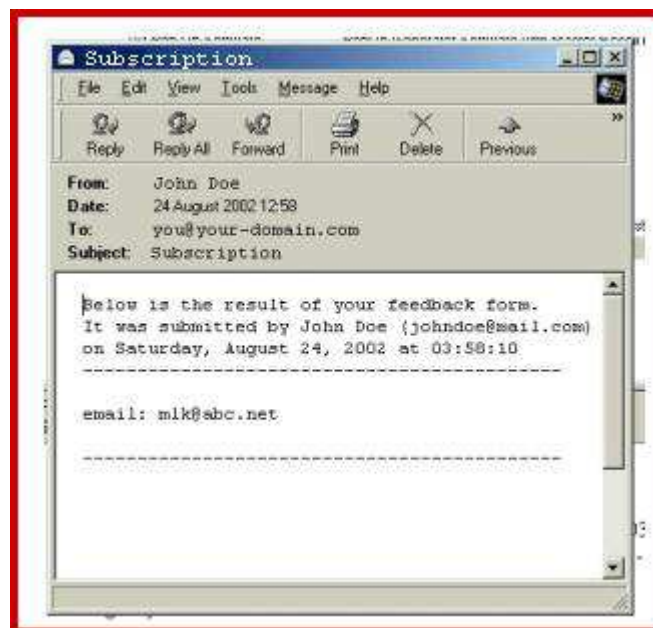
The HTML Mini-Course



However, the simple way, which doesn't need a script is this.

```
FORM ACTION="mailto:you@your-domain.com" METHOD="post"
```

And this is what the email we receive looks like.



The difference is quite clear. With this one it's up to you to do the unravelling, as you can see, the fields are separated by the "&" character and characters, like "/" and ";", are translated into hexadecimal codes.

Be Accurate!

Now you should be in a position to program your first web page. So its design, program, test, adjust, test, test and test again.

Now and then you're going to forget something small maybe a ' " ', maybe a ">" maybe even a whole tag. So be aware of the tiny things that can have large consequences.

Good luck.

Come and check what's on offer at <http://www.the-html-book.com>, you'll find it worth the visit.

Mike Hayes

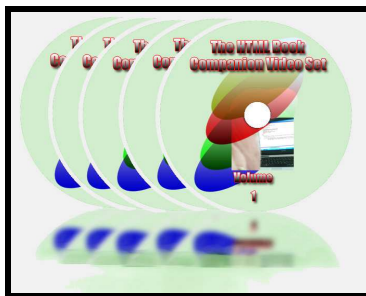
The HTML Mini-Course

If this mini-course has whetted your appetite for more HTML knowledge then you're most probably ready for ***The HTML Course Package***.

It comprises of:



The HTML Reference Book plus The HTML Exercise Answer Book.



The HTML Course Companion Video Set. Demonstrating coding HTML.



The HTML Course Bonus Videos: 1. Getting a PayPal buy now button.
2. Using cPanel on your host provider. 3. Using SmartFTP.

It's all here.

<http://www.the-html-book.com>

Other books by Mike Hayes

<http://www.the-job-hunters-book.com>

<http://www.the-sudoku-solutions-book.com>

<http://www.jigsaw-sudoku-solutions.com>

Published by Mike Hayes

An Irishman's Difficulties With The Dutch Language at

<http://www.irish-humor.com>