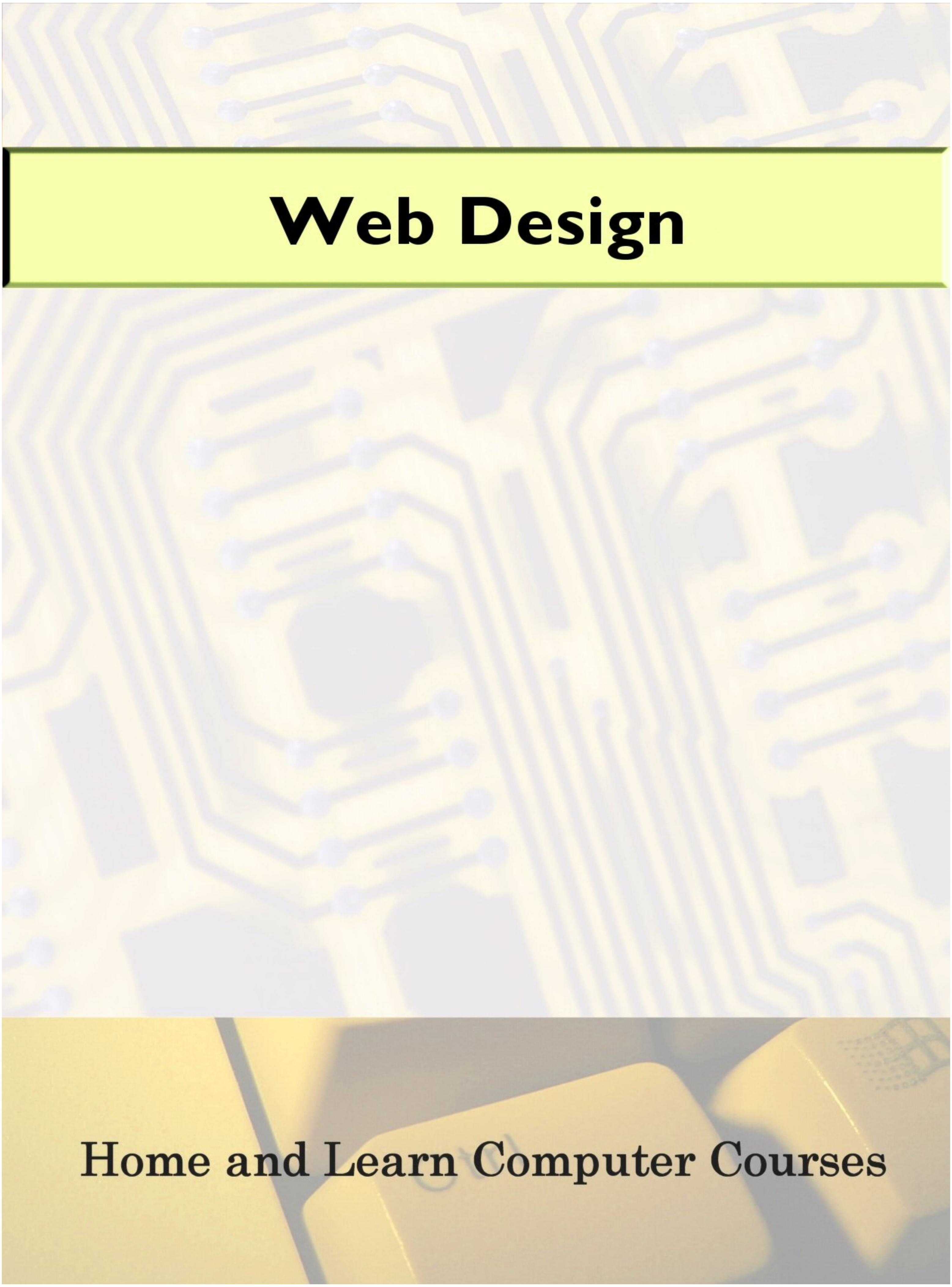
Web Design



Home and Learn Computer Courses

Web Design Contents

Introduction

Hello, and a very warm welcome to the Home and Learn computer book for Web Design. The software you need is set out below. We assume that you have absolutely no knowledge of designing web pages. Throughout the course of this book you will learn the fundamentals of Web Design. And, of course, you will start creating your own pages. By the end of the book, you will have acquired a good understanding of what web design is all about, and have the ability to take it further, if you so wish. At the very least, you will have given your brain a good work out!

What you will learn

The aim of this course is to get you started designing web pages. It is assumed that you have little or no experience of the subject. During our time together, you’ll learn some new concepts and ideas, most of which will not be difficult. After all, there are billions of pages on the internet, designed by the whole spectrum of humanity: children, adults, pensioners, people from all walks of life have sites out there. And if they can do it, so can you. In fact, follow the lessons carefully and you WILL do it. Before long, you’ll have your own pages designed and ready to be uploaded, there for all the world to see.

At the heart of every web page is something called HTML. You will learn what this is, and how to code it. In days gone by, people used to laboriously type all this HTML code by hand. Just one mistake could ruin an entire web page. Hours of painstaking searching could (and still can) be spent tracking down this single error. Nowadays, HTML editors are used to construct web pages. These editors make life a lot easier, and take the pain out of HTML coding. In fact, you can use something called a WYSIWYG editor (What You See Is What You Get) and not have to learn a single word of HTML code. A WYSIWYG editor will do everything for you with the click of a mouse. (Microsoft Word has a WYSIWYG editor built into it.) The problem with these editors is, because you don't learn any HTML code, you can't tinker when things go wrong. You therefore gain no real understanding of just what a web page is.

You, on the other hand, will learn exactly what a web page is, and how to fix it when things go wrong. You'll also be able to design a page to suit your needs, rather than have somebody else's design ideas forced upon you. The web pages that you construct will have your own unique stamp on them, and will not look like the ready-made templates that you see all the time when surfing the internet.

About the Home and Learn web design software

This course comes with its own web design software. It is a HTML editor that you will use to construct your pages quickly and easily. It has been designed so that you see the code at all times. This book will then explain what you are looking at and what the code does. Mostly, you won't have to type in any of the code itself, because it will be inserted with the click of a mouse. (However, feel free to use any editor you're comfortable with.)

The software can be downloaded from the web site. Connect to the internet, and go to the following page:

www.homeandlearn.co.uk/downloads/downloads.html

Once there, download all the files in the Web Book section. There is not only the Web Editor to download, but the Web Course files as well.

If you experience any difficulties, please contact us at the following email address:

online@homeandlearn.co.uk

Installing the software

Once you have downloaded the course material and software, you will need to unzip the files with WinZip. If you don't know how to do this, there is a tutorial at this address on our site:

http://www.homeandlearn.co.uk/BC/WinZipTutorial.html

Once you have unzipped the files, double click a folder to access it. To install the software, double click the **HTMLeditor** folder. Inside this folder is the set-up file **setup.exe**. Double click this file and the software should install.

Web Course Files

Besides the software, there are a number of files that you will need in order to complete certain sections.

Whenever you need a file for a section of your course, it will be explained in the relevant section. The **Web Course** files that you download in the link above should contain the following 8 folders:

Backgrounds
Forms
Frames
Imagemap
Images
Navigation
Review Four
Tables

Inside the folders are all the files referred to in this book.Once you have all the necessary files, you can begin.

Good luck!

Anatomy of a web page

In this section, you’re going to learn just what a web page is and how it can be read by a browser like Internet Explorer or Firefox. You’ll also get a first look at the web design software, and construct your very first web page. Let’s make a start. (If you’re reading this in Word, it’s a good idea to switch off the spelling: Tools > Options > Spelling and Grammar. This will get rid of the red wiggly lines.)

What is a web page?

Files on your computer come with extensions. If you wrote a letter using Microsoft Word and saved it with the name “MyLetter”, the software would add three letters to the file name. Because it was typed using Word, the three letters that get added to your filename are **.doc.** So your file name will be “MyLetter.doc” and not just “MyLetter”. If you created a spreadsheet in Microsoft Excel and called it “Accounts” the Excel software will add its own three letter extension to your file name. It will add **.xls**. So you file name will be “Accounts.xls” and not just “Accounts”. So you get a different three letter extension depending on the software you used.

These three letter extensions are very important to computers. They are used to identify the type of file it is. With a file extension, Word can recognise its own documents. It sees the letters **doc** and says “Ah yes, that’s one of mine. I can open it.” If it sees a different extension, **xls** for example, it says “What the heck is that?” You’ll then get an error message telling you that the file type is not recognised.

Web pages have their own file extensions. Oddly there are two different extensions, a three letter file extension and four letter extension. Web pages come with the extension **.htm** or **.html**. A browser can recognise either extension. With our software, the four letter extension is used when web pages are saved. (The letters HTML, incidentally, stand for Hypertext Mark-up Language.)

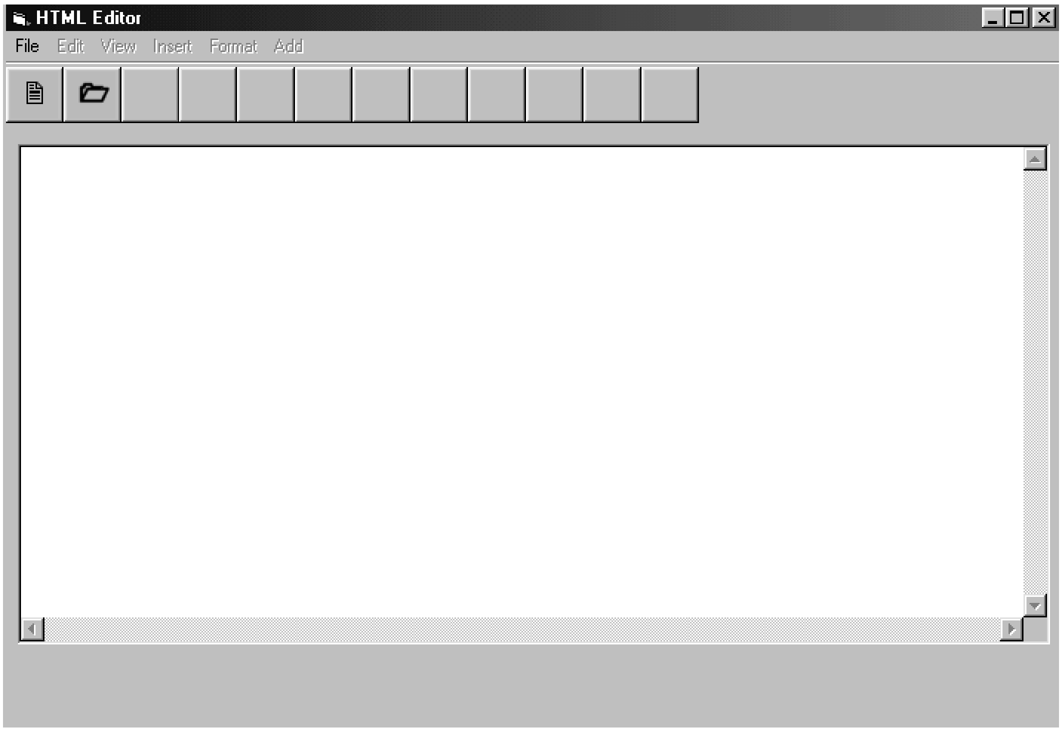
When you Open up a web page with your browser, Internet Explorer for example, the browser software checks the file extension, the same check that Word and Excel makes. If it sees the **.htm** or **.html** extension it knows it’s a web page and then tries to open it. (Modern browsers, though, can open up other types of files. Internet Explorer can open up your Word documents and PDF files.)

Behind the scenes, however, the thing that Internet Explorer is trying to open is really a text file that has had its extension changed from **.txt** to **.htm**. The text file though will have special symbols and words, called Tags. When the browser sees these Tags it goes to work, displaying whatever you typed and hiding the Tags from your viewers.

If all that seems a little confusing, don't worry. It will become clearer as you progress. For now, our goal is to design a first web page. So load up your web design software, and let's begin.

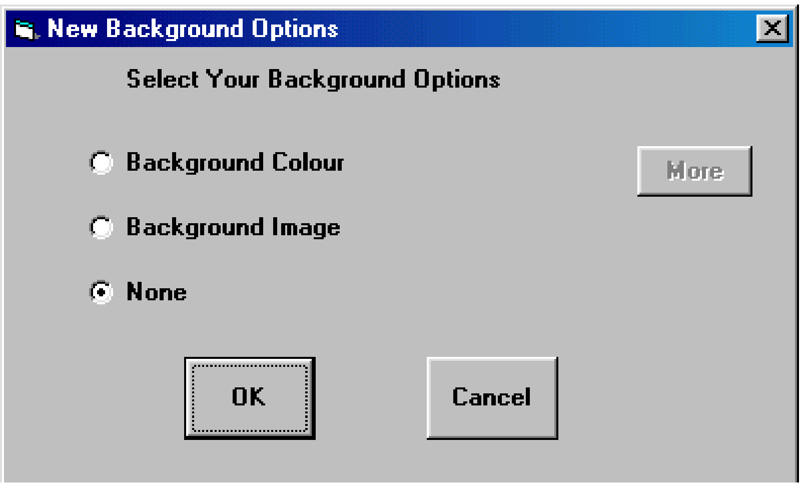
The Five Minute web page

When the software first loads it will look like this:



- Click **File** from the menu bar
- From the drop down menu, click **New**

When you click **New**, a dialogue box pops up asking you to choose your Background options. The dialogue box looks like this one:



New Background Options

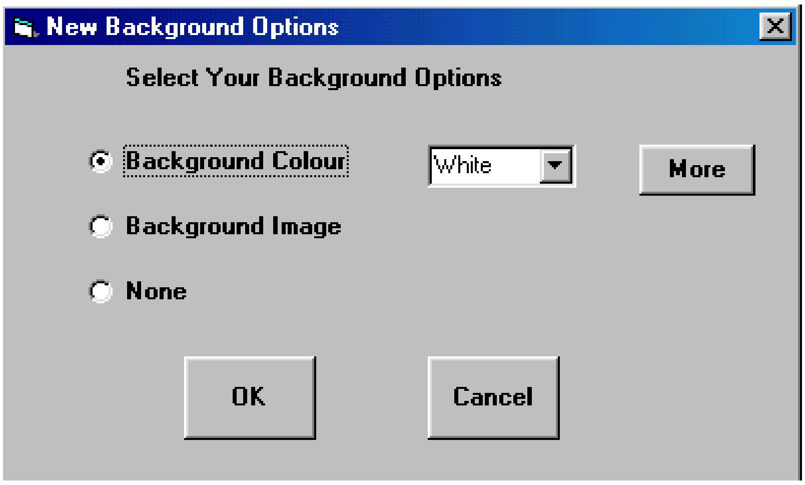
Select Your Background Options

○ Background Colour

○ Background Image

● None

The default is set to **None**. Click “Background Colour”. The dialogue box changes to show you the background colour options. It looks like the one below:



New Background Options

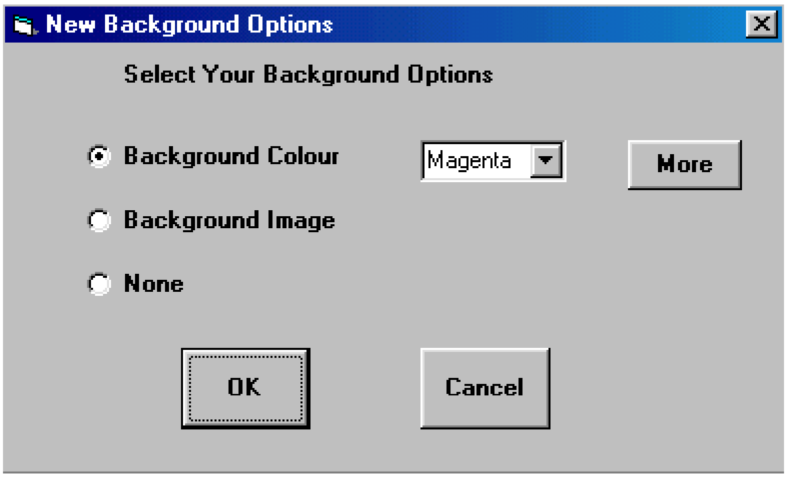
Select Your Background Options

● Background Colour: White

○ Background Image

○ None

Click the black down-pointing arrow just to the right of White to reveal a drop-down list of colours for your web page. Clicking the “More” button will reveal another dialogue box with a wider range of colours to choose from. For now, though, select one of the colours from the drop-down list. In the dialogue box below, we’ve gone for Magenta:



New Background Options

Select Your Background Options

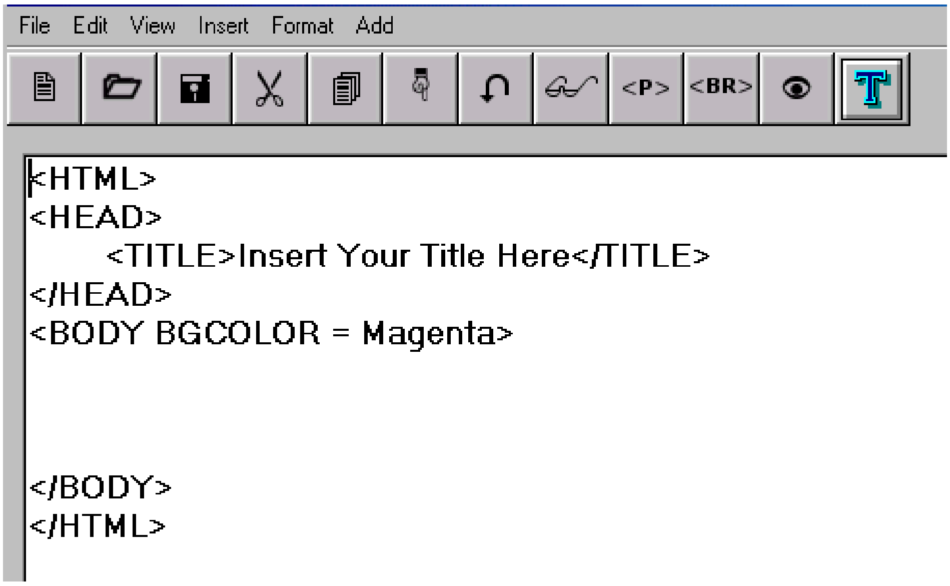
● Background Colour   Magenta

○ Background Image

○ None

Before clicking the OK button, notice that you can also set a Background Image for a web page, instead of using a coloured background. We'll explore this later.

When you have chosen your colour from the drop-down list, click the OK button. The dialogue box will close, and this rather daunting code will now be in your editor's window:

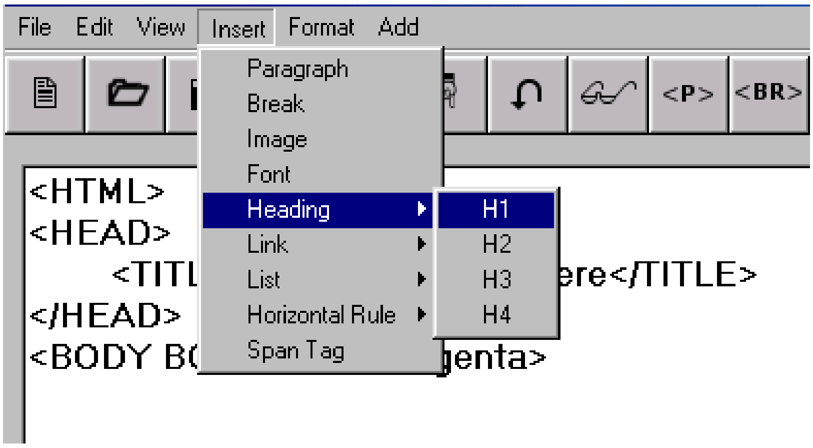


```
<HTML>
<HEAD>
        <TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY BGCOLOR = Magenta>



</BODY>
</HTML>
```

We'll explain what all that code means in a moment, but as we're designing a five minute web page, let's crack on.

The cursor will be flashing right at the top of your code. Move it down so that it's flashing on an empty line after "<BODY BGCOLOR = Magenta>" but before </BODY>.

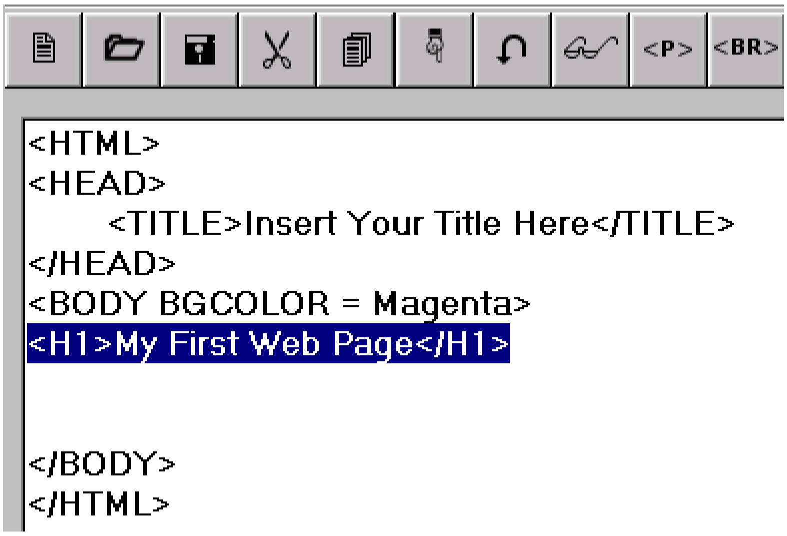
From the menu bar, click on **Insert > Heading > H1**



When you click H1 (the H stands for Heading) a pair of H1 tags will be inserted onto your page. We'll get on to what tags are soon, but for now just type something in between those pair of H1 tags:

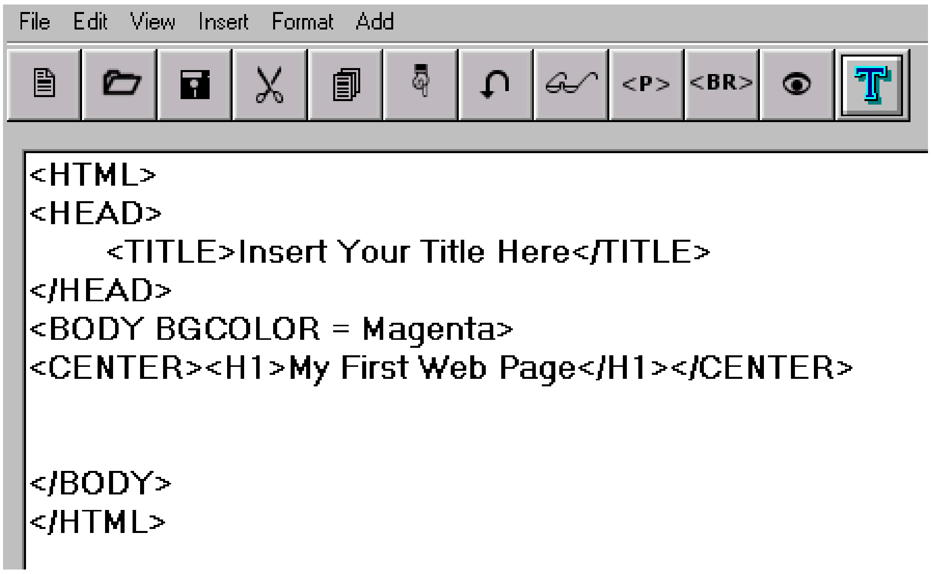
<H1>My First Web Page</H1>

Nearly there. Next, highlight that entire **H1** line, like in the image below:



```
<HTML>
<HEAD>
        <TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY BGCOLOR = Magenta>
<H1>My First Web Page</H1>


</BODY>
</HTML>
```

From the menu bar, click on **Format > Centre**. When you do, another pair of tags will be added to the start and end of those H1 tags. Your code will now look like this:



```
<HTML>
<HEAD>
    <TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY BGCOLOR = Magenta>
<CENTER><H1>My First Web Page</H1></CENTER>


</BODY>
</HTML>
```

Notice the spelling of the centre tag **- <CENTER>.** The American spelling “er” is used in HTML, rather than the English “re”.

OK, we’re done with our five minute web page. Time to see what it looks like.

- Click **File > Save As** and save your work
- Then click **View > View Web Page** (or click the pair of glasses on the toolbar).
- Your web page should launch in your browser

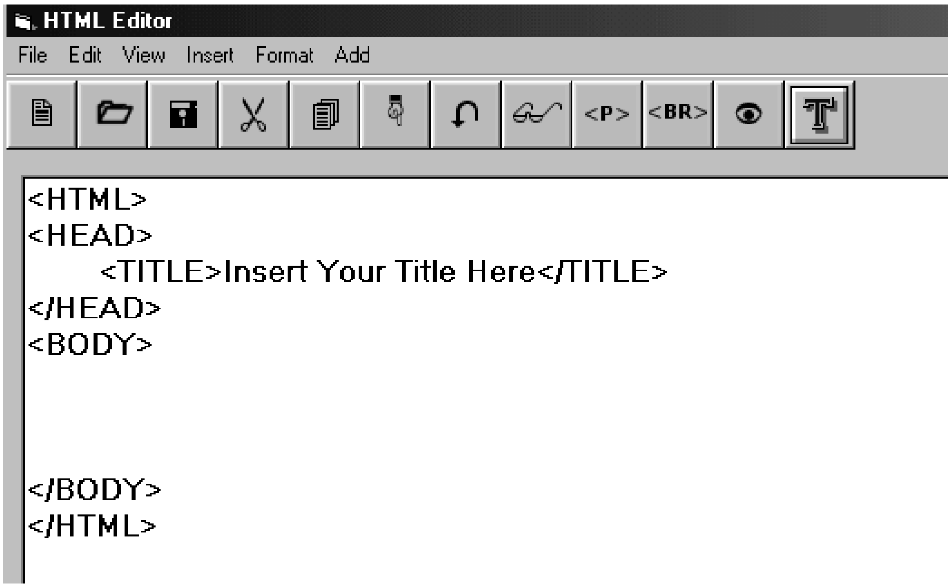
Congratulations! You have designed a web page. Easy, wasn’t it? Now for the slightly more difficult part of understand what it was you actually did.

The HTML Skeleton

In this next section we’re going to break down that code you inserted, and explain what it all means. To begin, let’s take a look at the basic HTML skeleton. The HTML skeleton is at the heart of most web pages on the internet. It is the starting point of web design, the basic code that is fleshed out to create something new and (hopefully) wonderful.

To see this skeleton, click on **File > New**. When the Background Options dialogue box pops up, leave it on the default “None” and just click the OK button.

Your editor will look like the one below:



```
<HTML>
<HEAD>
     <TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY>



</BODY>
</HTML>
```

- Highlight the part “Insert Your Title Here”
- Click on **Edit > Cut**

We’ll see what the Title tag does soon. But as the words “Insert Your Title Here” are not part of the basic HTML skeleton, we can chop them out. What you’re left with is this:

```
<HTML>
        <HEAD>
                <TITLE></TITLE>
        </HEAD>

        <BODY>

        </BODY>
</HTML>
```

And that is the HTML skeleton. That is what is at the heart of most page on the internet. Notice that there are two of everything: two HTML’s, two Heads, two Titles, and two Body’s. That is because Hypertext Mark-up Language is written in Tags, and Tags come in pairs (there are one or two tags that don’t come in pairs, but most of them do).

The first pair of tags are the HTML tags. You put one HTML tag at the top, and one at the bottom:

```
<HTML>



</HTML>
```

Two things to notice there. One, that the word HTML is surrounded by angle brackets (the Less Than and Greater Than keys on your keyboard); and two that the second Tag has a forward slash before the HTML.

All your tags must be surrounded by the angle brackets < >. This tells the browser that there is some HTML code that needs executing, and that it is not to be shown on the web page. Miss an angle bracket out and it can really mess up your web page.

The first Tag tells the browser to start doing something; the second tag tells the browser to stop doing it. The stop symbol is that forward slash. So miss that out and again your web page can really be messed up.

The next pair of tags are the HEAD tags: <HEAD> </HEAD>

```
<HTML>
        <HEAD>

        </HEAD>


</HTML>
```

Again, notice that both the HEAD tags are surrounded by angle brackets, and there is a forward slash before the final one </HEAD>.

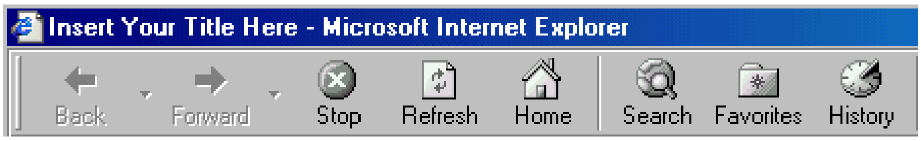
The HEAD section of HTML page is where you put special instructions for the browser. Your browser will not print directly to a web page when things are inserted into the HEAD section. For example, if you wanted to display the words “Hello World” on your page, and you typed them in between the two HEAD tags, then the Browser would just ignore them. That’s because “Hello World” is direct text, and not a special instruction that the browser can understand.

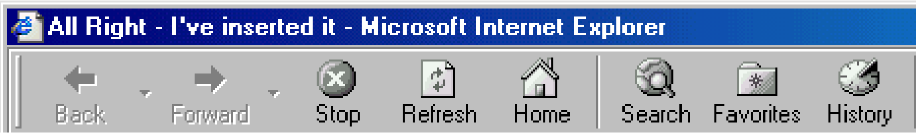
A special instruction your browser can understand is the TITLE tag: <TITLE> </TITLE>.

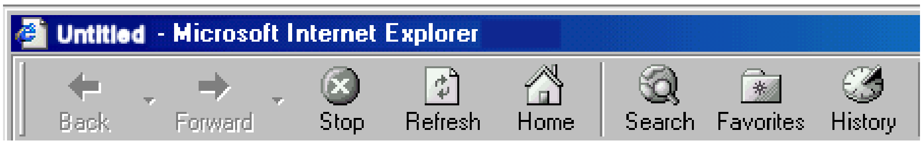
The Title tag always goes in the HEAD section of the HTML skeleton.

```
<HTML>
        <HEAD>
                <TITLE> </TITLE>
        </HEAD>


</HTML>
```

The TITLE tag doesn’t do very much. And it definitely should not be confused with the name of your web page. If you saved your web page as **MyFamily.html** then that would have no bearing on or relationship with the TITLE tag. Whatever you type between the two TITLE tags is what appears at the very top of your browser window. And that’s it, that’s all it does. As an example of what the TITLE tag does, examine these screenshots from the top of Internet Explorer:


Insert Your Title Here - Microsoft Internet Explorer


All Right - I've inserted it - Microsoft Internet Explorer


Untitled - Microsoft Internet Explorer

Remember when we chopped out the words “Insert Your Title Here”? Well, the first of the screenshots is what will happen if you don’t chop out the words “Insert Your Title Here”. The second screenshot has the words “Insert Your Title Here” chopped out and replaced by “All Right – I’ve inserted it”. The third screenshot is what happens when you have nothing at all between the two TITLE tags. You’ll quite often see web pages with “Untitled” right at the top of the browser window. If you see one of these on the internet, you know the web designer has not put anything between the two TITLE tags.

The final, and most important part, of the HTML skeleton is the BODY section.

```
<HTML>
        <HEAD>
                <TITLE> </TITLE>
        </HEAD>

        <BODY>


        </BODY>
</HTML>
```

In between those two BODY tags is where you’ll write most of your HTML code. Think of the BODY as the white page in Word Processing. When you type something in your word processing software, it is immediately displayed. Similarly, when you type something between the two BODY tags and then view the results in a browser, whatever you typed will be displayed. As long as it’s not a tag, your browser will try to display it for you. So remember: if you want people to see it, put it between the BODY tags.

Fortunately, you won’t have to type out the HTML skeleton yourself, because the software will do it all for you. But you will be making changes to the HTML skeleton, and it’s important you know what it is you’re amending.

So to recap, here’s the important points again:

- HTML is written in Tags
- Tags usually come in pairs

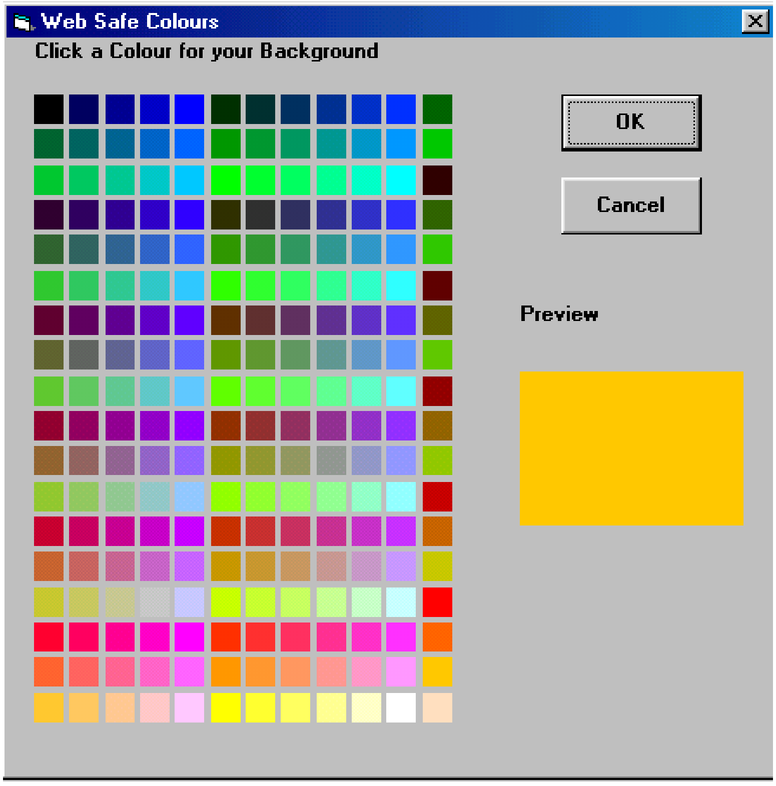
- A Tag is a word surrounded by angle brackets, e.g.: <BODY> </BODY>, <TITLE> </TITLE> , <HEAD> <HEAD>, <H1> </H1>
- A pair of tags has as a starting Tag and an end Tag. The end Tag is preceded by a forward slash
- The HTML skeleton is the foundation on which most internet pages are based

Now that you know what the basic HTML skeleton looks like, let's go through the five minute web page you just created, and see what we did to flesh out the skeleton.

- Click on File > New
- The Background Options dialogue box appears
- Click Background Colour
- Click the "More" button

When you click the "More" button, another dialogue box appears showing a list of Web Safe colours you can use (A Web Safe colour is one that will display properly no matter what browser or computer you have – at least, that's the theory).

Click on a colour that takes your fancy. We've gone for a yellow colour in the image below:



Web Safe Colours

Click a Colour for your Background

Click OK when you're happy with your chosen colour. You will be returned to the Background Options dialogue box. Click OK.

Your new background colour will be inserted into your code. Let's examine it in more detail.

In the basic HTML skeleton, the first BODY tag was just this:

<BODY>

If you don't specify a colour or background image to use, the browser will use the default colour. This is usually white. Notice what our BODY tag now looks like:

<BODY BGCOLOR = #FFCC00>

When we used a magenta colour, the tag looked like this:

<BODY BGCOLOR = Magenta>

A modern browser will recognise a lot of colours by name. But the colour names a browser recognises differ. The Firefox browser will not recognise a lot of the colour names that Internet Explorer uses, and vice versa. In which case, the default colour will be used, and not the name you specified. However, if you use a colour code instead of a colour name your colour should display properly (If it's a Web Safe colour).

The colour code we used was this:

#FFCC00

Colour codes start with the hash symbol (#) and then have six numbers or letters. The six numbers and letters are hexadecimal values. We won't go in to Hexadecimal maths, but here's the basic meaning of the code: The first two values are how much Red you want, the second two values how much Green you want, and the final two values are how much Blue you want. The values you are allowed to use for each number or letter are 0 to 9 and A to F.

So our code reads: switch the Red full on (FF), use a lot of Green (CC), but turn the Blue full off (00).

Here’s the whole line again:

<BODY BGCOLOR = #FFCC00>

Notice how the BODY tag has been extended. There’s now a space after **<BODY** and that is followed by the word **BGCOLOR** and then the code. Finally, we have the angle bracket at the end.

So an important point about tags is this: A lot of HTML tags can be extended. In other words, you can add to the basic Tag. The BODY tag is a good case in point. Not only can you extend the BODY tag by adding a background colour to it, you can also specify what colour you want the text to be on the page. Like this:

<BODY BGCOLOR=#FFCC00 TEXT=Black>

Notice where the spaces are. There’s one after **BODY** and there’s one after **BGCOLOR=#FFCC00**. So when you are extending a tag, remember to separate the different parts by a space.

There are a number of other ways to extend the BODY tag, which we’ll meet later. Let’s move on to what else was done in the five minute web page.

Headings

The Heading tag in HTML comes in seven different sizes: 1 to 7. In the editor, only sizes 1 to 4 are used. Size 1 is the biggest Heading size, and size 7 is the smallest. A heading tag is the letter “H” followed by a number 1 to 7 (You don’t need to use capital letters in HTML tags, by the way, because it is not case sensitive. So use lower case, if you prefer: The browser won’t care.)

In the five minute web page, we used size 1 heading tag by clicking on **Insert > Heading > H1**. This gave us the following:

<H1> </H1>

But that is just the heading tag. We then need to type something between the pair of H1 tags. We used this as a heading:

<H1>My First Web Page**</H1>**

What we’re doing is telling the browser to display the words “My First Web Page”. But because we’ve surrounded the words with a pair of tags, we’re also telling the browser how to format those words. In this case, format the words with a Heading size of 1.

The final thing we did in the five minute web page was to centre the title on the page. If we didn’t centre our heading, the browser would just align it on the left. First we highlighted the H1 tags, then we clicked **Format > Centre**. Which gave us this:

<CENTER><H1>My First Web Page</H1>**</CENTER>**

Notice that the Heading tag, and our text are now surrounded by the two CENTER tags. This is called Nesting. Here, we have nested the H1 tag inside the CENTER tag. The text itself is nested inside both of these tags.

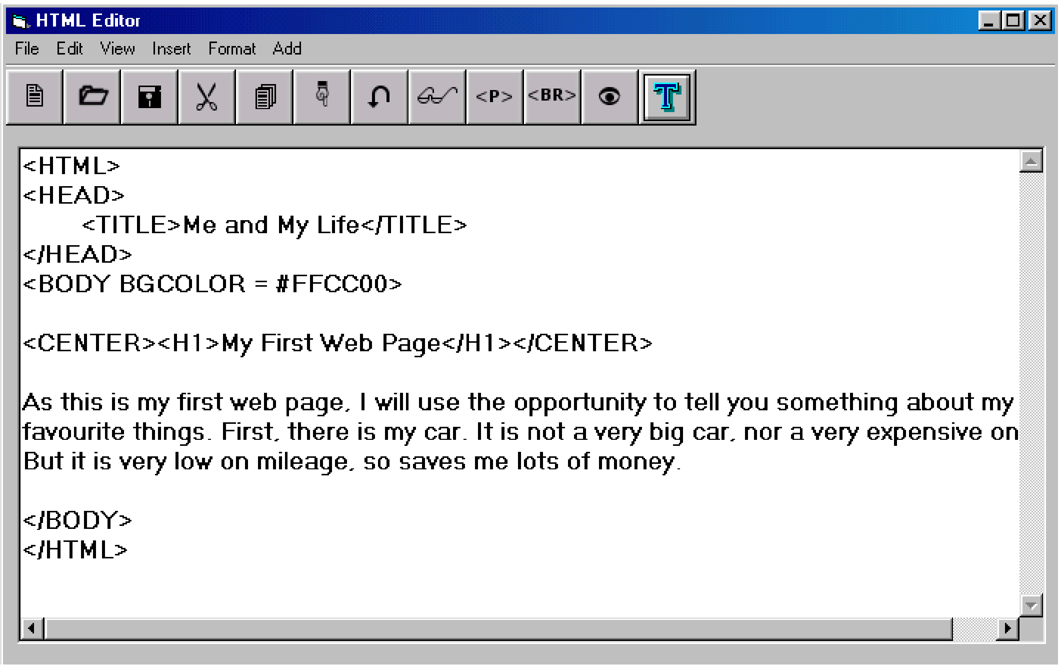
Save your work and then click **View > View Web Page** to see the results, and the new colour code in action.

As an experiment, change both the H1 tags to H7 (just delete the 1’s and type in a 7).

<CENTER><H7>My First Web Page</H7>**</CENTER>**

Save your work again, and then click **View > View Web Page** once more. See how much smaller the Heading is?

Set the Heading back to H1. On a line after the H1 and CENTER tags, type out some text, as in the image below:



```
<HTML>
<HEAD>
      <TITLE>Me and My Life</TITLE>
</HEAD>
<BODY BGCOLOR = #FFCC00>

<CENTER><H1>My First Web Page</H1></CENTER>

As this is my first web page, I will use the opportunity to tell you something about my
favourite things. First, there is my car. It is not a very big car, nor a very expensive on
But it is very low on mileage, so saves me lots of money.

</BODY>
</HTML>
```

When you have typed some sample text, save your work and then view the results. You should see that your text has gone below the heading, with some generous space between the two.

And now it’s time for your first Review.

Review Number One

Design a simple web page with one paragraph of text. The web page should have the following:

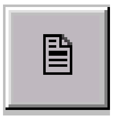
- A coloured background
- A Title that is not the default title
- A centred heading of size 1
- Some text below the heading, at least three lines

At this stage, you only need to keep it simple. When you are finished, you should be able to view your work in a browser.

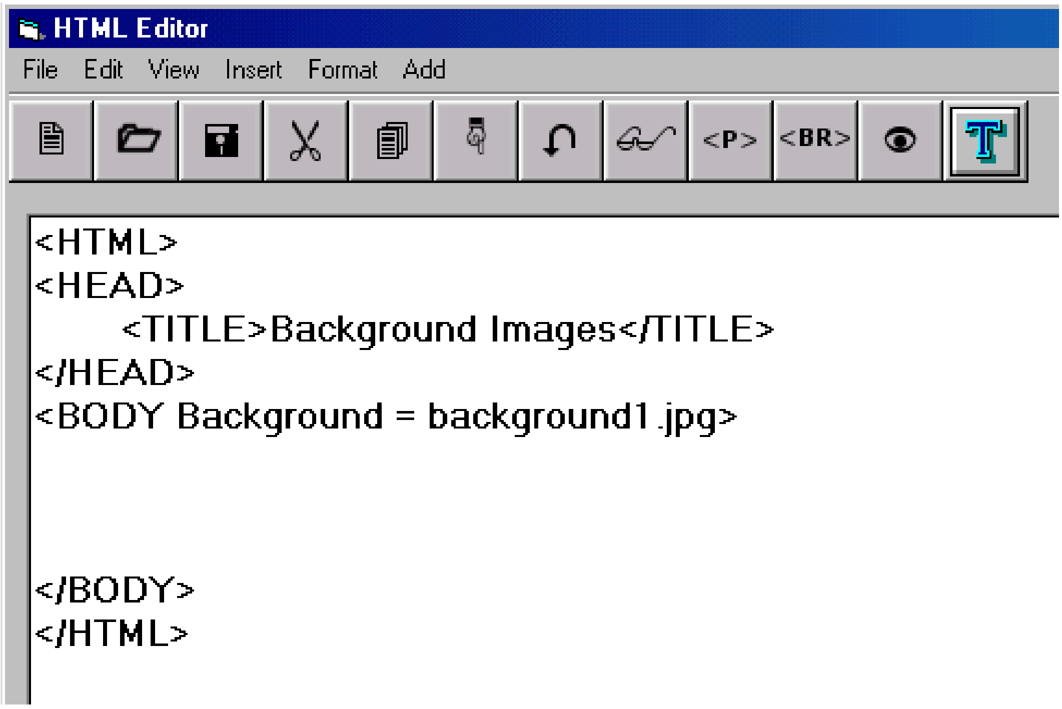
Images

In the last section, you used a colour for the background. But the background does not have to be a colour; it can be an image, if you prefer. We'll see how to add an image as the background now, and then discuss image types used on the internet. We'll also discuss another crucial topic to do with image files and HTML files – how they are referenced. We'll then move on to inserting images generally.

- So, load up your HTML Editor, and click on **File > New**
- Alternatively, click this icon:



- From the Background Options dialogue box, click on "Background Image". The Open dialogue box appears.
- Navigate to the folder you downloaded at the start of this book called **Backgrounds** (inside the HTML folder), and click on one of the background images.
- Click OK when you get back to the Background Options dialogue box
- Your HTML Editor will now look something like this one



```
<HTML>
<HEAD>
      <TITLE>Background Images</TITLE>
</HEAD>
<BODY Background = background1.jpg>



</BODY>
</HTML>
```

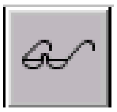
Notice the new Tag – Background. When we wanted a coloured background for our web page, the Tag was this:

<BODY BGCOLOR = “Maroon”>

When you want a background image, instead of a background colour, the Tag is:

<BODY **BACKGROUND** = “filename”>

Before discussing file types, and how to reference images, click **File > Save** or **File > Save As**, and save your web page. Then click on **View > View Web Page**. Or click this icon in the toolbar:



The background image you chose for your web page might not have displayed. Or even if it did, there will be a big problem if you were to put it on the internet. The problem is due to the way the image is referenced. We’ll discuss that now.

Referencing image files and HTML files

If the background image didn't display, then the culprit is your HTML Editor. When you clicked on Background Options, then Background Image, you searched for an image that was copied to your own hard drive. The full path to the image file your hard drive might be this:

D:\HTML\Backgrounds\background1.jpg

However, if you look at the code in your HTML Editor the path has been shortened to this:

background1.jpg

The reason why the file path has been shortened is because of something called a Relative Reference. The un-shortened file path of **D:\Backgrounds\background1.jpg** is an Absolute Reference. We'll discuss those first.

Absolute References

Consider what would happen if you had this in your HTML code:

<BODY BACKGROUND = D:\HTML\Backgrounds\background1.jpg>

If you put the page on the internet, you'd have a problem. Because you're saying, "There is a folder on the D Drive called HTML. Inside this folder, there is another folder called Backgrounds. Inside the folder called Backgrounds, there is an image called background1.jpg. Please use this image as the background image for my web page."

You have used an Absolute Reference. And that's the problem. Although you will have all these folders on your D drive, somebody else viewing your web page won't. But that person's browser will try to look for all these folders. Of course it won't be able to find them, because all the folders are on your computer, and not theirs. So the background you specified to use in your web page won't be displayed to anybody else but you.

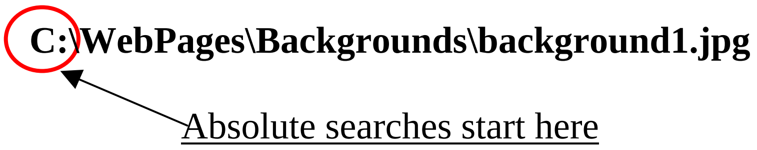
Absolute References are fine if you want to link to somebody else's web page on the internet. And you've seen this many times:

http://www.homeandlearn.co.uk

That's an absolute reference to a web page on the internet. Perfectly OK, and you'll see how to add links to web pages later. But Absolute References and not terribly practical when it comes to you own images and your own html pages. Relative References are much better.

Relative References

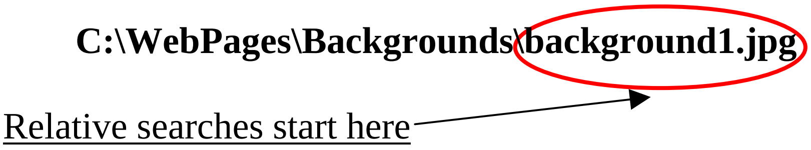
With Relative Referencing, the starting point is not your own computer, but the image file or HTML file itself. With Absolute Referencing, the browser starts the search for the image on the left hand side:



C:\WebPages\Backgrounds\background1.jpg

Absolute searches start here

With Relative Referencing, the browser starts the search for the image on the right hand side:



C:\WebPages\Backgrounds\background1.jpg

Relative searches start here

So with a Relative reference, the browser starts looking for a file called "background1.jpg". If you just put this:

<BODY BACKGROUND = "background.jpg">

the browser will look for the file in the same folder where you saved your web page. If it's there, no problem; the browser will display the image. You don't need to add anything else, because the browser will stop searching when the image has been found. In fact, the ONLY place the browser will look is in the same folder where you saved your web page.

So with Relative Referencing, if you put all your images and web pages in the same folder, the browser will know where to find everything. And when you're asking the browser to display an image or another web page, you only need the name of the image or web page. You don't need to do this:

<BODY BACKGROUND = **C:\WebPages\Backgrounds\background1.jpg**>

You can just do this:

<BODY BACKGROUND = “background1.jpg”>

With this HTML Editor, Relative References are used. So if you create a folder on your hard drive called “Web Pages”, you can save all your image and web pages into it. That way, all you images will display properly, and all the links to your web pages will work.

On a professional level, though, dumping everything into one folder is frowned upon. If you created a folder called “Web Pages” you would be expected to create other folders inside this one. A folder called “Images” to store all your image files; a folder called “Scripts” to store any external code; and a few other folders as well.

If you do that, Relative Referencing gets a little bit trickier. If you have a web page called index.html inside the “Web Pages” folder and wanted to use an image that was inside the folder called “Images”, and this folder itself was in the “Web Pages” folder, you couldn’t do this:

<BODY BACKGROUND = “background1.jpg”>

If you tried that, the image wouldn’t display. That’s because you haven’t told the browser about that folder called “Images”. You would have to do this:

<BODY BACKGROUND = “Images/background1.jpg”>

The forward slash means “look for a folder called . . . that is in the same folder as the current file.” In this case, look for a folder called “Images”.

If you wanted to point to an image that was outside the “Web Pages” folder, then life just got even trickier still. Unfortunately, the solution is outside the scope of this beginner’s book. But as a beginner, if you keep everything in the same folder – images and web pages – then this sort of relative referencing should work:

<BODY BACKGROUND = “background1.jpg”>

To sum up, then:

- Your HTML Editor uses Relative Referencing
- To ensure that images and web pages display correctly, create a folder on your hard drive and put all your course work into this new folder

So after you create a new folder on your hard drive, start the Background image project again. Save your web page into the folder you create. And then do one important thing: copy the background images (not the folder) from the old folder to your newly created folder. When you've done all that, click on **View > View Web Page**. Your background image should now display.

Experiment with the other background images. There are 17 in all, some awful and some OK. To quickly change the background image, do the following.

If you started with this:

<BODY BACKGROUND = "background1.jpg">

- Change the "1" to a "2"
- Click **File > Save** (or click the Save icon on the toolbar)
- Click **View > View Web Page** (or click the glasses icon on the toolbar)

Image Types

You remember at the start of this book we talked about file extensions, and you learnt what the .html file extension was. You're now going to learn about two new file extensions: JPEG and GIF.

Just like web pages and word process document have file extensions, so too do images have file extensions. There are a wide range of file extensions used for images. The two most popular file extensions used on web pages are JPEG and GIF.

GIF Images

GIF stands for Graphics Interchange Format, and was developed by Compuserve. It uses something called a LZW compression algorithm to reduce the size of the file. The size of your images on web pages is a crucial factor for people using 56k modems (most people). A 56k modem will have a maximum download speed of 56 kilobits per second. This is about 7 kilobytes per second. If you ignore image size, and have a picture on your web page that is 700 kilobytes, it will take well over two minutes to download (56k modems rarely work at their maximum speed). Most web surfers will not wait that long for your image to download (would you?). They'll be gone long before the two minute mark.

So compressing the size of an image makes sense on the internet. If you've scanned a photo and saved it to your hard drive, take note of the file extension used. The software package you used to view the scanned image will probably give you the opportunity to save the file in another format. Saving the file as a GIF image will greatly reduce the size of the file.

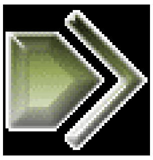
There is a down side, however, in converting to a GIF. The number of colours in your image is capped at 256 colours. For realistic photos, this is not nearly enough. And although your file size is greatly reduced, so too could be the quality of your image.

JPEG Images

JPEG is the other popular image format used on web pages. It stands for Joint Photographic Experts Group, and allows you to have images with more than 256 colours. In fact, millions more. Again, your original image is compressed when you convert to a JPEG image, so some loss of quality is inevitable. If you want to show off your photos in their best light, converting to JPEG rather than GIF is the best option for the internet. The size of the file might be slightly higher, but then so will the quality.

The downside to JPEG images is that jpeg is a Loss Compression format. This means that image quality goes down (rapidly) the more times you compress and uncompress the image. GIF images, on the other hand, is a Lossless Compression format, meaning there will be no loss of quality when you compress and uncompress the image.

Another thing you can’t do with JPEG images is have a transparent background. So if your image was this:



and you wanted to get rid of that black background, you could set the black colour as a transparent colour, if you save as a GIF. With a JPEG, you’re stuck with the black background. (you can also animate GIF image, which you can’t do with JPEG images.)

In general, if your image is less than 256 colours, then save the image as a GIF; if the image is more than 256 colours, and quality loss is important, then save the image as a JPEG.

One last thing about JPEG images in web pages. When specifying an image in your HTML code, the “E” of JPEG goes missing. So it’s this:

BACKGROUND = image1.**jpg**

and not this:

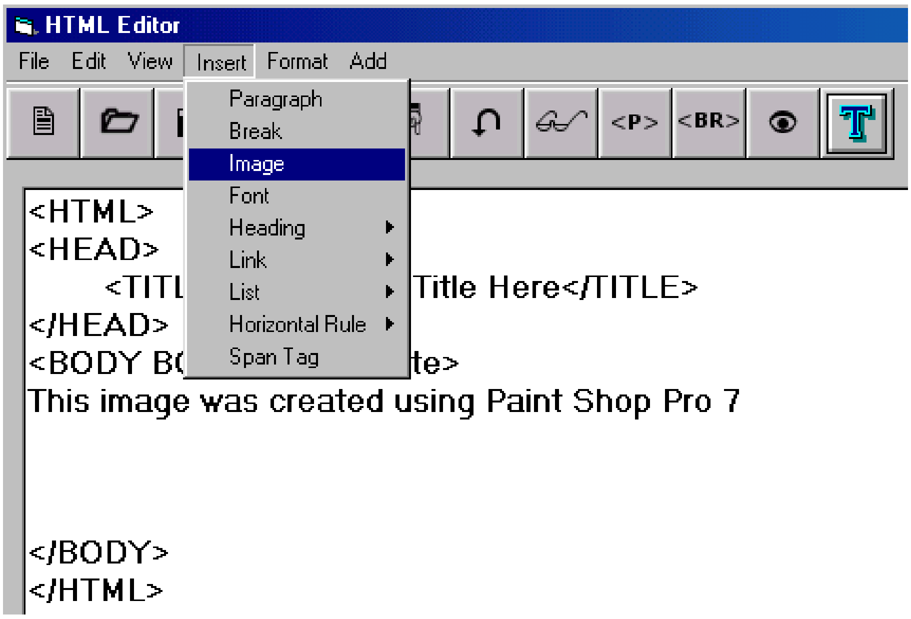
BACKGROUND = image1.**jpeg**

So let’s design a web page with an image on it.

- Create a new web page with your HTML Editor software
- Set the background colour to white
- In between the two BODY tags, type “This image was created using Paint Shop Pro 7”
- Save your web page

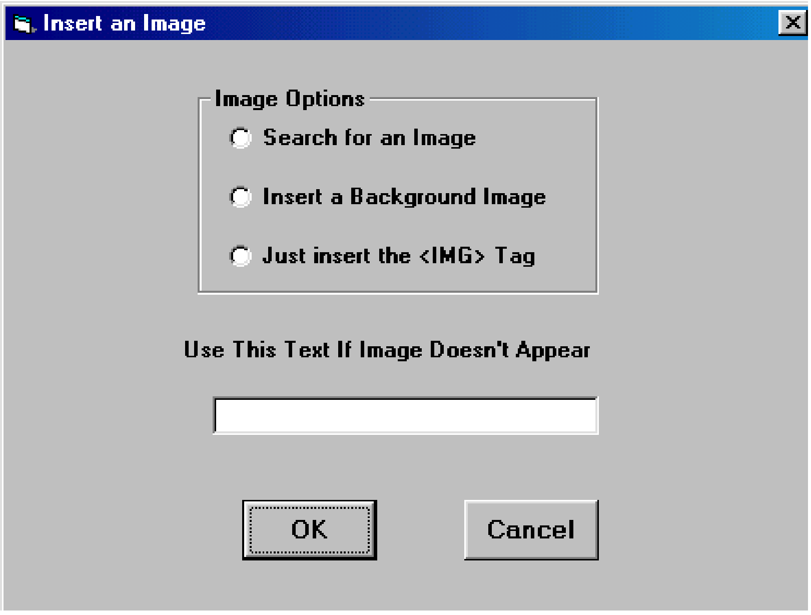
Copy the planet images you downloaded at the start of this book (HTML folder, then the images folder) to the same place where you saved your web page. To insert an image, do the following:

- From the menu bar, click on **Insert > Image**

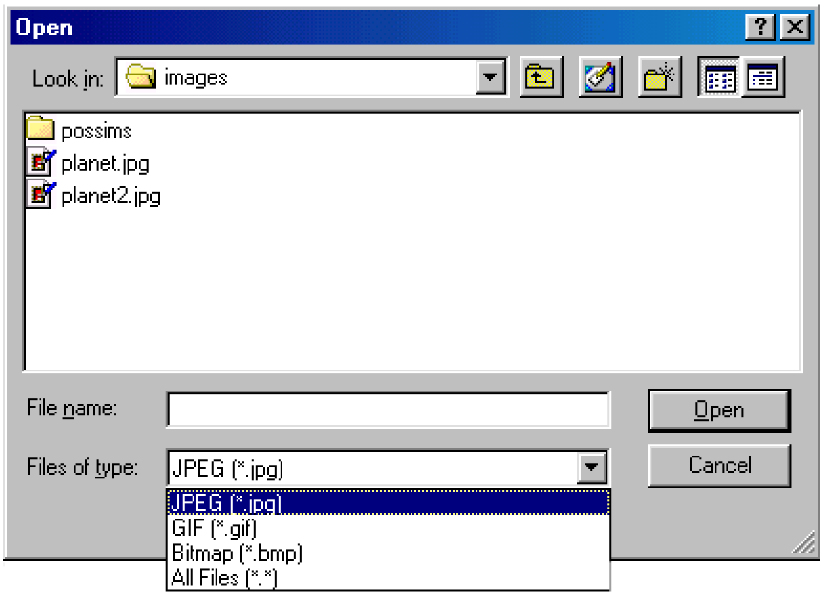


```
<HTML>
<HEAD>
      <TITL        Title Here</TITLE>
</HEAD>
<BODY B(             te>
This image was created using Paint Shop Pro 7


</BODY>
</HTML>
```

The Insert Image dialogue box appears:


Insert an Image
Image Options
○ Search for an Image
○ Insert a Background Image
○ Just insert the <IMG> Tag
Use This Text If Image Doesn't Appear
OK
Cancel

- Click the “Search for an Image” option
- The Windows “Open” dialogue box pops up.
- Locate the folder where you saved your web page and planet images



- Click the black down-pointing arrow next to "Files of Type"
- To view only GIF files, click GIF option. Here, we're going to be using one of the JPEG images
- Click on one of the planet images, then click Open

You will be taken back to the Insert Image dialogue box. Notice the text box at the bottom: "Use this text if image doesn't appear". Sometimes, things go wrong with web pages, and the browser cannot find the image. In that case, you can type some alternative text which will display if the image is not found.

- So click inside the text box and type "Planet Image"
- Then click OK
- Your HTML code should now look like this:

```
<HTML>
<HEAD>
	<TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY BGCOLOR = White>
This image was created using Paint Shop Pro 7

<IMG SRC = "planet2.jpg" ALT = "Planet Image">

</BODY>
</HTML>
```

- Click on **View** > **View Web page** to see the results of your work.

What happened? You probably found that the image displayed on the right hand side, and the text on the left hand side, near the bottom of the image.

But if you look at the code, you’ll see we typed the text first, then put the Image code on the line below it. So why didn’t the image appear underneath the text?

The image appears to the right of the text because of one reason: HTML code does not recognise the carriage return. So when you press the enter key on your keyboard to get a new line, the HTML Editor itself knows what you mean, and what you want. But when you load the code into a browser, that new line break is not recognised. The browser thinks you want the text and the image on the same line.

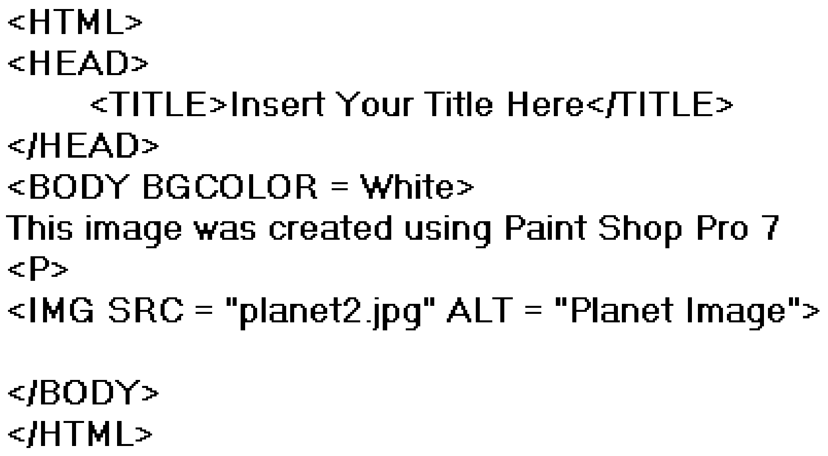
With HTML code, you have to “tell” the browser that you want to start a new paragraph. If you forget, then the browser will put everything on the same line.

The way you “tell” the browser to start a new line is with either the <P> tag or the <BR> tag. The P stands for Paragraph, and the BR for Break. The difference between the two is that the <P> tag gives you double space, and the <BR> tag gives you single space.

To insert a Paragraph tag, click where you want the new paragraph to be. Then click **Insert > Paragraph** from the menu bar. Or click the **<P>** icon in the toolbar.

To insert a Break tag, click where you want the new Break to be. Then click **Insert > Break** from the menu bar. Or click the **<BR>** icon in the toolbar.

Amend your code so that it looks like the one below:

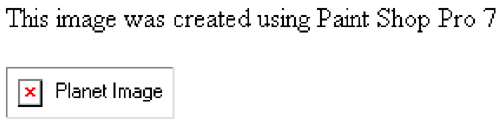


```
<HTML>
<HEAD>
     <TITLE>Insert Your Title Here</TITLE>
</HEAD>
<BODY BGCOLOR = White>
This image was created using Paint Shop Pro 7
<P>
<IMG SRC = "planet2.jpg" ALT = "Planet Image">

</BODY>
</HTML>
```

Save your work, and then view the web page. You should see the image appear below the text. To see the ALT part of the image code in action, delete a couple of letters from the file name. Something like this:

<IMG SRC = "anet2.jpg" ALT = "Planet Image">

Save your work, and then view the web page again. You should have gotten something like this in the browser:



This image was created using Paint Shop Pro 7

☒ Planet Image

The red X signifies that an image could not be found. The ALT text we typed appears in its place.

Image Alignment

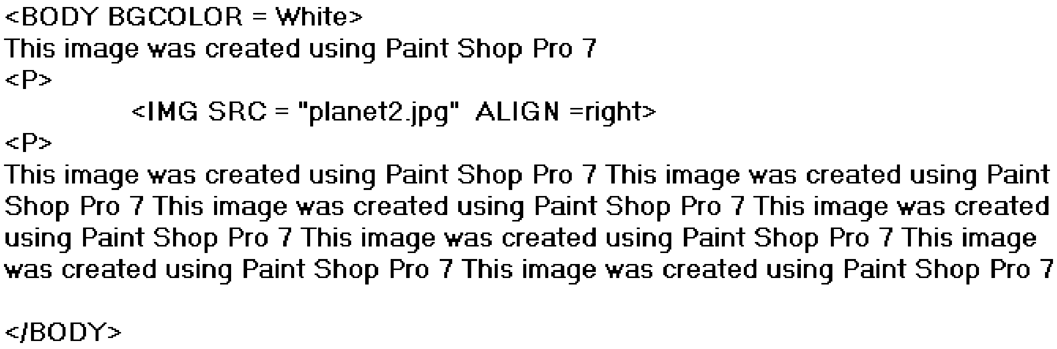
As well as extending the Image tag by adding Alternative text, you can use the ALIGN tag with the Image tag. However, when you use the ALIGN tag with the IMG tag you are aligning text with your image. Let’s try an example to clear things up.

Change your Image tag so that it looks like the one below (we’ve left out the ALT tag):

<IMG SRC = “planet2.jpg ALIGN = right>

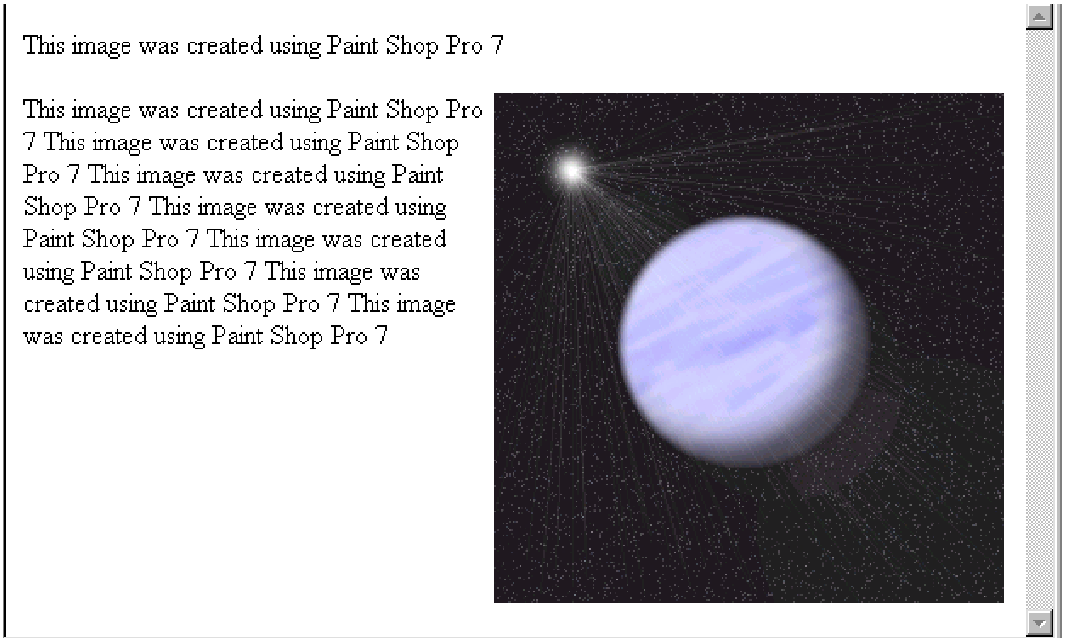
Save your work, and then view the results.

You should find that the image shifts nicely to the right hand side of the browser. However, there's a problem. Type some more text after the IMG tag, something like the one below, which is just the title copied and pasted a few times:



```
<BODY BGCOLOR = White>
This image was created using Paint Shop Pro 7
<P>
          <IMG SRC = "planet2.jpg"  ALIGN =right>
<P>
This image was created using Paint Shop Pro 7 This image was created using Paint
Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created
using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image
was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7

</BODY>
```

When you've finished, save your work and view the results. The page in a web browser should look something like the one below:



This image was created using Paint Shop Pro 7

This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7 This image was created using Paint Shop Pro 7

The problem is that although the image is aligned to the right, any text we put after that will wrap to the left of the image. Similarly, if we put ALIGN = Left, the image would appear on the left hand side, and the text would wrap to the right of it. And if we didn't want the text there, but below the image, we're stuck.

There are few solutions, one of which we'll see soon. But bear in mind if you're adding either **ALIGN = Right** or **Align = Left** to an image tag then what you are doing is wrapping text around an image. You are not aligning an image independent of text.

There are some other Aligning options you can use with images and text. Play around with these and see what they do. But cut all the text from after the image to just a couple of words, otherwise either nothing will happen or you’ll get strange results.

```
<IMG SRC = “planet2.jpg” ALIGN = Texttop>
<IMG SRC = “planet2.jpg” ALIGN = Top>
<IMG SRC = “planet2.jpg” ALIGN = Middle>
<IMG SRC = “planet2.jpg” ALIGN = Absmiddle>
<IMG SRC = “planet2.jpg” ALIGN = Bottom>
<IMG SRC = “planet2.jpg” ALIGN = Absbottom>
```

The software doesn’t add these Align tags for you, so you’ll have to type them all out by hand. This will give you some practice in amending HTML code.

Image Height and Width

Another thing you can do with the Image tag is specify a new height and width. This one is quite easy. Change your image tag to this:

```
<IMG SRC = "planet2.jpg" Height = 600 Width = 600>
```

The original image was 300 pixels high by 300 pixels wide. By changing the Height and Width to 600, we can double the size of the image. The image itself will still take the same amount of time to load into a browser because we haven’t changed the size of the JPEG file. All we’ve done is to change the height and width attributes of the IMG tag.

Changing the size down to say 100 by 100 would not be practical. Because the size of the JPEG file wouldn’t change. All you’ll do is use up more download time to get a smaller image. So when changing image size with the Height and Width attributes, go up in size and not down.

What you’re really doing is just zooming in. Unfortunately, the more you zoom in the more the quality will decrease. So don’t increase the size of an image too much.

To end this section, let’s recap on the different ways to insert images into your web pages.

Insert a Background image

- If you want to use a background image instead of a colour, the code goes into the BODY tag:

<BODY BACKGROUND = ImageName.FileType>

- The two most popular File Types used on web pages are JPEG and GIF images:

<BODY BACKGROUND = ImageName**.jpg>**

<BODY BACKGROUND = ImageName**.gif>**

- When telling the browser where your image is, use Relative referencing rather the Absolute referencing:

Relative <BODY BACKGROUND = **planet.jpg**>

Absolute <BODY BACKGROUND = C:\Images\planet.jpg>

But bear this in mind: These two are different image files:

planet.jpg
Planet.jpg

The first is lowercase “p” and the second a capital “P”. If you get it wrong, the image won’t appear. File names are case sensitive.

Normal Images

- The code to insert a normal image is this:

<IMG SRC = “ImageName.FileType”>

A common mistake is this:

<IMGSRC = “planet.jpg”>

And this is another common mistake:

<IMG SCR = “planet.gif”>

(If you can't see where the mistakes are, an email will get you the answer.)

- There are a number of ways to extend the Image tag. Here's a list:

ALT	The alternative text to use if the image doesn't display: <img src = planet.jpg **ALT** = "Planet Image">
ALIGN	Used for text wrap. The two basic options are Align = Left and Align = Right. <img src = planet.jpg ALIGN = Left> <img src = planet.jpg ALIGN = Right>
HEIGHT	Specify a new Height for your image <img src = planet.jpg Height = 600>
WIDTH	Specify a new Width for your image <img src = planet.jpg Width = 600>

Other ways to extend the Image tag:

Border	Adds a border around your image when used with links. To switch the border off, set it to zero <img src = planet.jpg **BORDER** = 5>
HSPACE	Adds a bit of space above and below an image when used with the ALIGN tag <img src = planet.jpg Align = Left **HSPACE** = 20>
VSPACE	Adds a bit of space to the sides an image when used with the ALIGN tag

	<img src = planet.jpg Align = Right **VSPACE** = 20>
USEMAP	Specifies an image map to use with an image <MAP NAME = “Recs”> <AREA SHAPE = “rect” COORDS ="0, 0, 100, 100" HREF = page1.html> <AREA SHAPE = “rect” COORDS ="0, 200, 100, 100"HREF = page1.html> </MAP> <img src = planet.jpg **USEMAP** = “#Recs”>

One last thing before Review Number 2. You’ll remember that using the Align tag is for text wrapping, and that it was stated there are a number of ways to align an image if you don’t want text wrapping around your picture. Here’s one solution using your HTML editor.

- Take out any align tags in the image
- Highlight the image you want aligning
- From the menu bar, click on Format
- From the drop down menu, click on Align Right
- The result will be this:

```
<Div Align = "Right">
        <IMG SRC = "planet.jpg">
</Div>
```

Now, your image will align to the right, but you can put text above and below the image, without the text wrapping to the left or right.

And now on to Review Number 2

Review Number Two

Create a web page with the following:

- A background image
- A few paragraphs of text
- A normal image or images of your choice
- Alternative text if the image doesn't appear

Try to create something around a theme: your life, family member, a celebrity, a hobby, etc.

If you haven't got any images of your own, you can either use the ones you download at the start of the book, or why not go on the internet and download some images that take your fancy? You can save images on web pages by doing the following:

Internet Explorer

- Click on the image with your right hand mouse button
- From the menu that appears, click on **Save Picture As**
- When the Save As dialogue box pops up, choose a location on your computer

Netscape/Mozilla/Firefox

- Click on the image with your right hand mouse button
- From the menu that appears, click on **Save Image**
- When the Save As dialogue box pops up, choose a location on your computer

Text Formatting

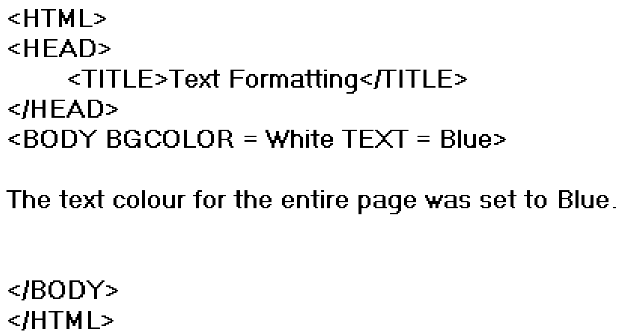
In this section, you’ll learn about the various ways you can format text using HTML code. Let’s make a start.

Getting text into your web page is quite easy – just start typing between the two BODY tags. When you save your work and view it in a browser, the text will appear without any special formatting. You don’t need to specify a type of font to use, or a colour, or a size. This is because browsers are set up to use defaults when no instructions are given. Text formatting, in html terms, is when you give the browser instructions on how your text should look. These instructions come in the form of tags, and you’ll learn these now.

Text Colour for the entire web page

You can specify which colour you want to use for all the text on your web page. The browser default is black, so you only really need to specify a colour if you don’t want black. Here’s how to do it.

- Start your HTML Editor
- Click **File > New**
- Set a background colour and click OK
- When you see the code, locate the first BODY tag
- Click your cursor after the “Y” of BODY
- Press the spacebar on your keyboard
- Type in TEXT =
- Then type a colour for your text
- In between the two BODY tags, start typing you text
- Save your web page, and then view the results
- In the code below, the text has been set to Blue



```
<HTML>
<HEAD>
     <TITLE>Text Formatting</TITLE>
</HEAD>
<BODY BGCOLOR = White TEXT = Blue>

The text colour for the entire page was set to Blue.


</BODY>
</HTML>
```

Headings

You’ve already met the Headings tag. If you’ll remember, the Heading tag uses the letter “H” followed by a number. Like this:

<H1>My Heading</H1>

or like this:

<H4>My Heading</H4>

The heading tag increases or decreases text size above or below the default size. The default size is 3. The biggest size is 1, and the smallest 7.

The heading tag will automatically insert a line break for you, so you don’t have to insert a <P> or <BR> tag after it to start a new line.

To insert a Heading tag with the HTML Editor, click on **Insert > Heading** then choose a Heading size. Only size 1 to 4 are used in the editor. But bear in mind, you can go as low as size 7.

When you click on a heading size, the beginning and end Tag will be inserted with the cursor flashing between the two:

<H1> </H1>

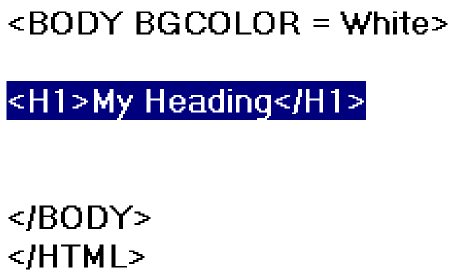
The forward slash, remember, tells the browser to stop formatting a tag. In this case, stop formatting the H1 heading tag.

With the cursor flashing between the two tags, type your text for the heading.

Centring Text

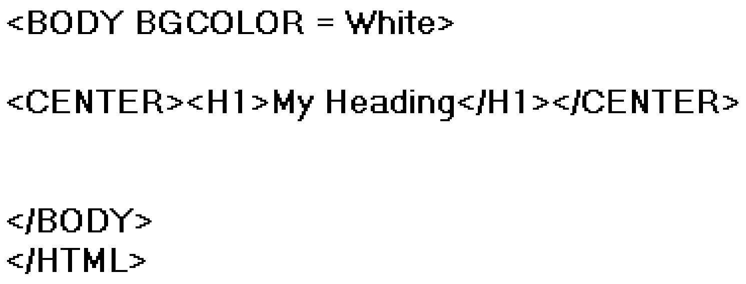
To centre text (and anything else on your page), the CENTER tag is used. Note the American spelling, with “ER” at the end and not “RE”. To centre something with the HTML Editor, do the following:

- Highlight what you want to centre (in the image below, a H1 heading tag is highlighted)



```
<BODY BGCOLOR = White>

<H1>My Heading</H1>


</BODY>
</HTML>
```

- From the menu bar, click on **Format > Centre**
- The two CENTER tags will surround your highlighted text
- Save your work and view the results in your browser

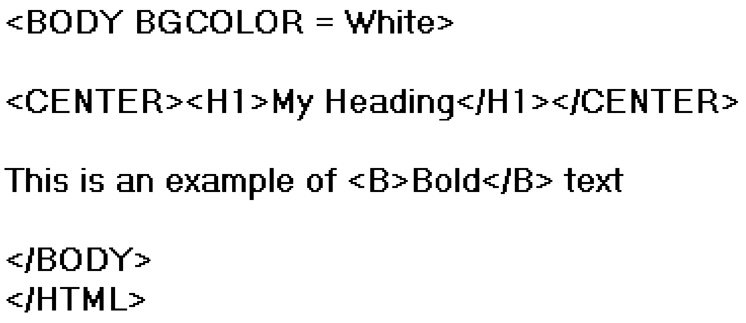


```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>


</BODY>
</HTML>
```

Bold Text

You can make text stand out by using the bold tag. The process is the same as above:

- Highlight the text you want to make bold
- From the menu bar, click on **Format > Bold**
- The two **<B></B>** tags will surround your highlighted text
- Save your work and view the results in your browser

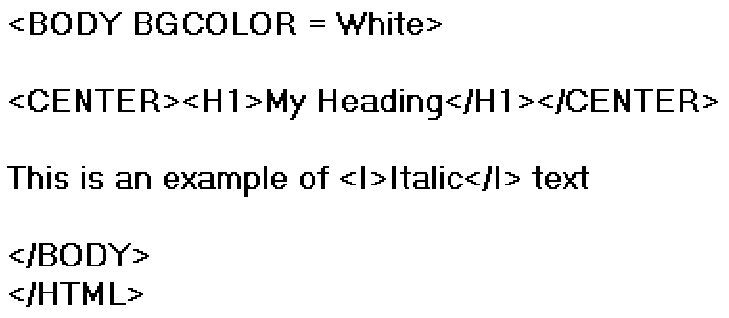


```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>

This is an example of <B>Bold</B> text

</BODY>
</HTML>
```

Italics

The HTML tag to make text italic is as simple as the <B> tag for bold text. With your HTML editor, do the following:

- Highlight the text for the italics
- From the menu bar, click on **Format > Italics**
- The two **<I></I>** tags will surround your highlighted text
- Save your work and view the results in your browser



```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>

This is an example of <I>Italic</I> text

</BODY>
</HTML>
```

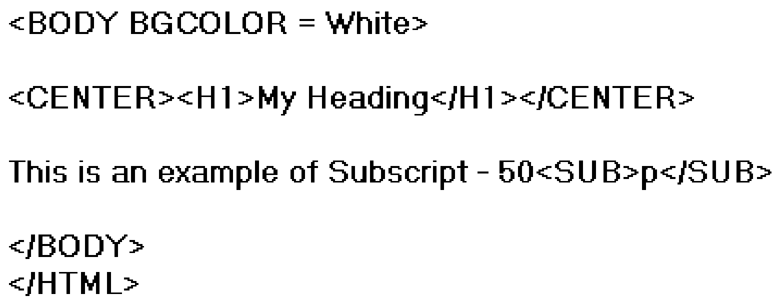
Subscript

Subscript is text that appears below normal text. Like this:

$$\mathbf{50_p}$$

You can have Subscript in your HTML code quite easily. To format text as subscript using your HTML editor, do this:

- Highlight the text you want to turn into subscript
- From the menu bar, click on **Format > Subscript**
- The two **<SUB></SUB>** tags will surround your highlighted text
- Save your work and view the results in your browser



```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>

This is an example of Subscript - 50<SUB>p</SUB>

</BODY>
</HTML>
```

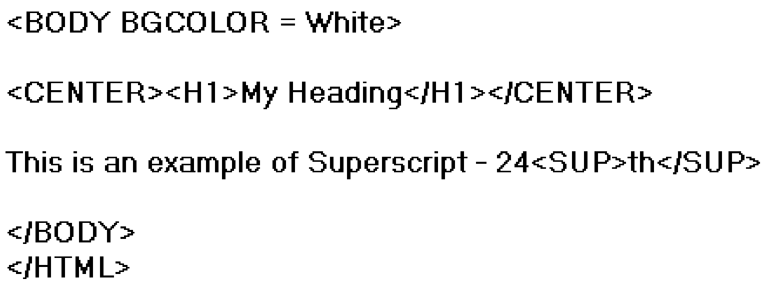
Superscript

Superscript is text that is raised slightly above the centre. Like this

24th

To format text as superscript using your HTML editor, do this:

- Highlight the text you want to turn into superscript
- From the menu bar, click on **Format > Superscript**
- The two **<SUP></SUP>** tags will surround your highlighted text
- Save your work and view the results in your browser



```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>

This is an example of Superscript - 24<SUP>th</SUP>

</BODY>
</HTML>
```

The Font Tag

The Font tag lets you specify how you want you font formatted. You can specify the size of the font, the type of font, and the colour of the font. You can specify all three in the same tag, or just use one or two.

The basic font tag is this:

<FONT> </FONT>

But that won't work, were you to surround some text with those two tags. That's because you need to specify at least one of the three font formatting attributes.

To specify the size of your font only, use this:

<FONT SIZE = 3> </FONT>

Just like the Heading tag, the Font Size comes in 7 flavours. This time, however, size 1 is the smallest, and size 7 the biggest. The default is size 3. This means that if you leave the size out, you’ll get size 3 font for your text.

To specify the type of font you want, the word FACE is used:

```
<FONT FACE = “Arial”></Font>
```

You can specify more than one font face to use. The point of doing this is that the person viewing your page might not have your font on his or her computer. The other fonts you specify will then be used as alternatives, if the first choice is not available. Each font face is separated by a comma:

```
<FONT FACE = “Arial, Helvetica, sans serif”></Font>
```

In the case above, if the person hasn’t got Arial, a Helvetica font will be used; if no Helvetica font is available, then any sans serif font will be used.

To specify a colour for your font, the American spelling is used: COLOR. The colour you use can be the name of a colour, or the colour code. So both of these are valid:

```
<FONT COLOR = RED> </FONT>

<FONT COLOR = #FF0000> </FONT>
```

Note the hash (#) symbol before the colour code. Miss it out, and you don’t get the colour you asked for.

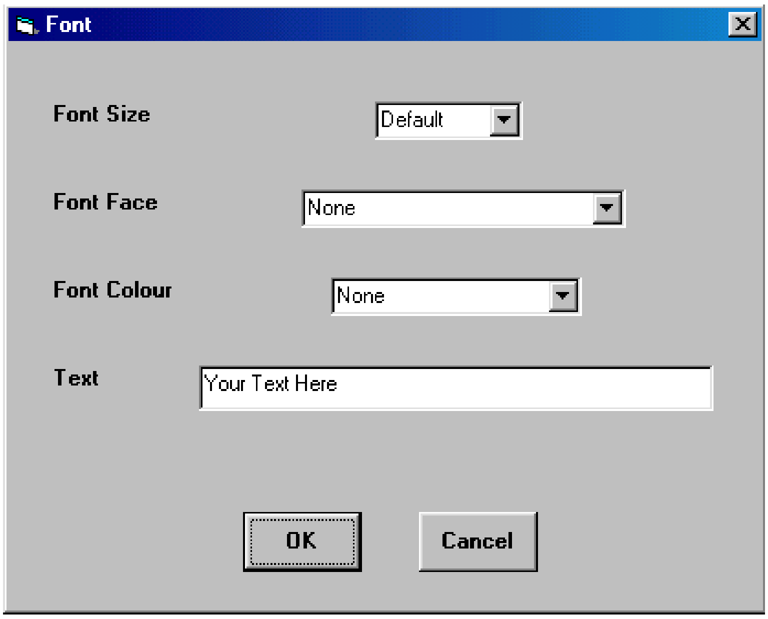
To specify all three in the same tag, the code is this:

```
<FONT SIZE = 5 FACE = “Arial, Helvetica, sans-serif” COLOR = Red>
                Your Text Here
</FONT>
```

Let’s use the HTML Editor to insert some Font formatting.

- Click anywhere in between the two BODY tags
- From the menu bar, click **Insert > Font**
- The Font dialogue box appears

The Font Dialogue Box



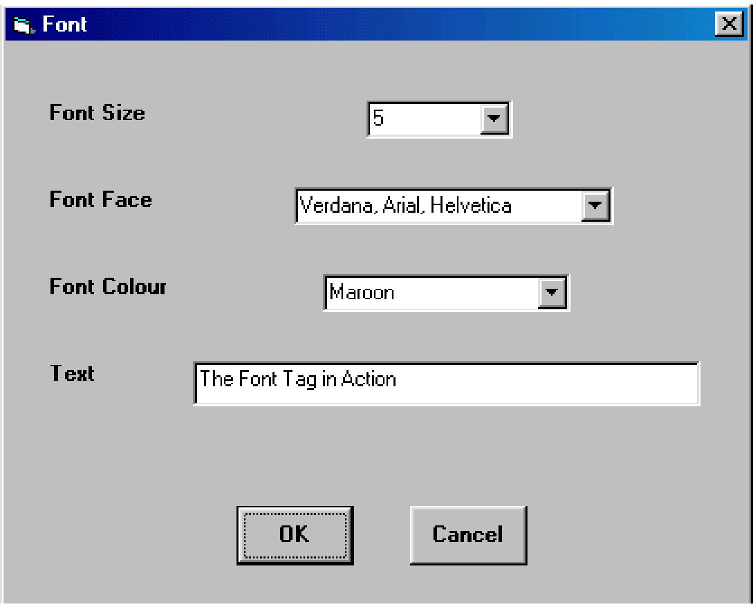
Font Size: Default

Font Face: None

Font Colour: None

Text: Your Text Here

- Click the black arrow on the drop down boxes, and choose a Font Size, a Font Face and a Font Colour. If you'd rather just have one of them (or two) leave the others set to None or Default
- Delete the text from the text box, and type in your own text



Font Size: 5

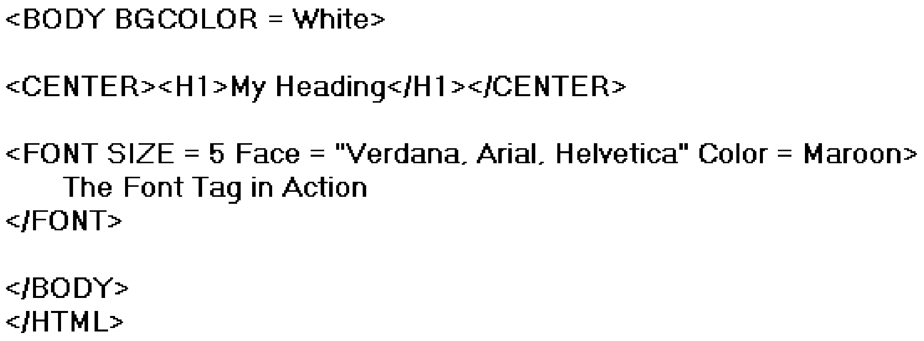
Font Face: Verdana, Arial, Helvetica

Font Colour: Maroon

Text: The Font Tag in Action

- Click OK

When you're done, the code should look like the one below:



```
<BODY BGCOLOR = White>

<CENTER><H1>My Heading</H1></CENTER>

<FONT SIZE = 5 Face = "Verdana, Arial, Helvetica" Color = Maroon>
    The Font Tag in Action
</FONT>

</BODY>
</HTML>
```

Save your work, and view the results in your browser.

The Font tag can be used on a single letter, a single word, or blocks of text. It can be quite effective.

Text Colour

If all you want to do is to change the colour of some text, your HTML editor has a quick way of doing that.

- Highlight the text whose colour you want to change
- From the menu bar, click on **Format > Text Colour**
- The Colour dialogue box appears
- Choose a colour by clicking on a colour square
- Click OK

Lists

You can insert a bulleted list onto your page with HTML. There are two types of lists to choose from: Ordered List and Unordered. An Ordered list is one that uses number, letters, roman numerals, or a combination. Like this:

i. Item One
ii. Item Two
iii. Item Three

A. Item One
B. Item Two
C. Item Three

An Unordered List is one that uses bullets. Like this:

- o Item One
- o Item Two
- o Item Three

- Item One
- Item Two
- Item Three

To get a list with numbers or letters, this tag is used:

<OL>

</OL>

The OL stands for Ordered List, of course. But those two tags won't get you a list. You need to include another tag, the <LI> tag. LI stands for List Item. You need one <LI> tag for every item in your list. So for three items, the code is this:

```
<OL>
        <LI>Item One
        <LI>Item Two
        <LI>Item Three
</OL>
```

To get the bulleted list, the <UL> tag is used. UL stands for Unordered List. It's used in exactly the same way. Just substitute the OL tags for UL tags.

To use the HTML Editor to insert a list, do the following:

- Click where you want the list to start
- From the menu bar, click on **Insert > List**
- Choose either Ordered or Unordered
- The editor will insert the code
- Delete the default text and type your own

However, the editor will give you only one <LI> tag, so you'll have to type the others yourself. Note that the <LI> tag doesn't need an end tag.

For both the Ordered and Unordered list, you can specify which type you want to use for the bullets or numbers. The types are these:

Ordered List Types (Numbers is the default)

Capital Letters	A
Lowercase letters	a
Capital Roman Numerals	I
Lowercase Roman Numerals	i
Numbers	1

You use the Types like this:

```
<OL TYPE = A>
        <LI>Item One
        <LI>Item Two
        <LI>Item Three
</OL>
```

You can specify a start, as well. But the start has to be a number:

```
<OL TYPE = A Start = 7>
        <LI>Item One
        <LI>Item Two
        <LI>Item Three
</OL>
```

So that List will be uppercase letters, starting at the 7th letter, or “G”.

Unordered List Types

There are three types of bullets you can use:

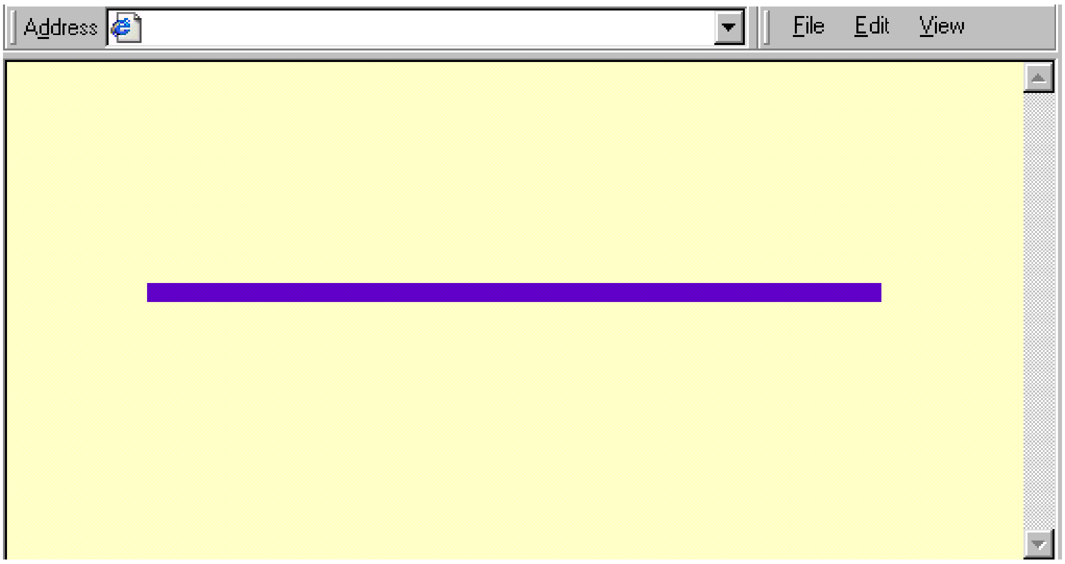
Disc
Circle
Square

```
<UL TYPE = Circle>
        <LI>Item One
        <LI>Item Two
        <LI>Item Three
</UL>
```

Try out the various lists in your editor, and add the different types to the code. That way you will get a feel for how they are used, and what they look like.

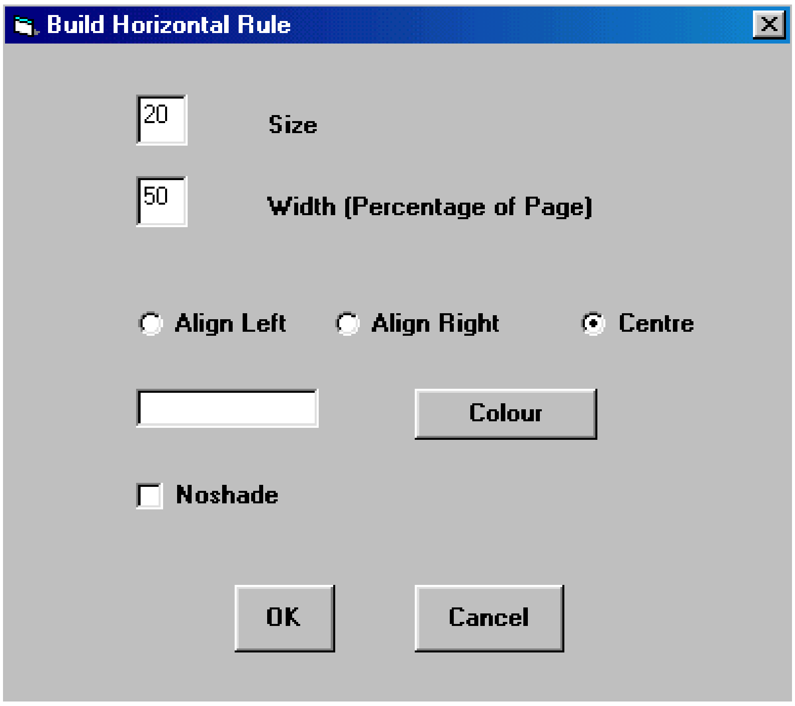
Horizontal Rules

A Horizontal rule can look quite nice on a page, if it’s coloured rule. The Rule is used to divide one section from another. Here’s an image of a coloured Horizontal Rule, with nothing else on the page.



To insert a Horizontal Rule using your Editor, do the following:

- Click in between the two BODY where you want your Horizontal Rule to appear
- From the Editor’s menu bar, click on **Insert**
- From the drop down menu, click **Horizontal Rule**
- From the sub menu, click **Build HR**. The following dialogue box appears:



Build Horizontal Rule

20 Size

50 Width (Percentage of Page)

○ Align Left ○ Align Right ● Centre

□ Noshade

- Type in a Size for your Rule. This is the height
- Type in a width
- Choose an alignment option: Left, Right or Centre
- Click on the Colour button and choose a colour for your Horizontal Rule
- The Noshade option is supposed to either add or remove a shadow from the rule, but has mixed results with coloured rules. Just ignore it
- Click OK when you’re done
- The following code will be added to your page:

```
<HR SIZE = 10 WIDTH = 75% ALIGN = Center  COLOR = #6600CC>
```

You can have the HR tag just by itself, which is why the Editor has a Basic HR option as well as the Build HR option. Try it out and see what it looks like on a web page.

For reference purposes, here’s a list of the Text formatting tags you have met in this section. There’s also some you haven’t met, too. Try them and see how they work.

TEXT	Used inside the BODY tag to specify a text colour for all the web page <BODY BGCOLOR = White **TEXT** = Blue>
<H1></H1>	Used for a Heading. The sizes are numbered 1 to 7, with 7 being the smallest. **<H1>Heading</H1>**
<CENTER>	Used for centring things on your web page **<CENTER>**Some Text**</CENTER>**
<P>	The paragraph tag. HTML doesn't insert a paragraph break for you, so you must do it yourself. Use the <P> tag when you want to start a new paragraph. You can also align things using the <P> tag. Like this, for an address: **<P ALIGN = RIGHT>** 49 Falkland Street <BR> Middlesbrough <BR> TS1 4HJ **</P>** Or put the address in the centre with **<P ALIGN = CENTER>**
<BR>	The line break. The browser will automatically give you a line break if your text overlaps to the right, so there's no need to insert a <BR> tag to start every line of text. Use the <BR> tag when you want to force a single line of space. You can't extend this tag like you can the <P> tag.

<B> </B>	Used for making text Bold **<B>**Some Text**</B>**
<I></I>	Used for turning text into italics **<I>**Some Text**</I>**
<SUB> </SUB>	The Subscript tag **50<SUB>P </SUB>**
<SUP> </SUP>	The Superscript tag 24**<SUP>**th**</SUP>**
<FONT SIZE =	Sets the size of some text. The biggest is size 7, and the smallest size 1 **<FONT SIZE = 5>Some Text</FONT>**
<FONT FACE =	Sets the type of font you want. It is better to set more than one font, just in case the person viewing the web page doesn't have the font you've specified. **<FONT FACE =Arial, Helvetica, san-serif>** Some Text **</FONT>**
<FONT COLOR =	Sets the colour of the font you want to use. **<FONT COLOR = Blue>** Some Text **</FONT>**
<OL>	Used for an Ordered List. Must include the List Item tag, <LI> **<OL>** **<LI>**Item One **<LI>**Item Two **</OL>**

OL TYPES	Specify a Type to use for your Ordered List. You can specify a starting position with the START attribute, but it must be a number Capital Letters A Lowercase letters a Capital Roman Numerals I Lowercase Roman Numerals i Numbers 1 **<OL TYPE = 1 Start = 5>** **<LI>**Item One **<LI>**Item Two **</OL>**
<UL>	Used for an Unordered List. Must include the List Item tag, <LI> **<UL>** **<LI>**Item One **<LI>**Item Two **</UL>**
UL Types	You can specify what type of bullet you would like. Disc Circle Square <UL **TYPE = Square>** <LI>Item One <LI>Item Two </UL>

Here are some other Text Formatting tags you can use in your code. Try them out and see how they work

<BASEFONT SIZE = >	Used for setting the size of the text on the entire page. The sizes range from 1 to 7, with 7 being the biggest. The default is size 3. Put it on the line after

	the first BODY tag <BODY BGCOLOR = White> **<BASEFONT SIZE = 4>**
<U></U>	The Underline tag **<U>**Some Text**</U>**
<STRIKE>	Used for striking through text **<STRIKE>Some Text</STRIKE>**
<EM></EM>	Short for emphasis. The result is text with italics **<EM>**Some Text**</EM>**
<STRONG>	An alternative to the Bold tag **<STRONG>**Some Text**</STRONG>**
<BIG>	Makes text bigger, usually by one size above the default, or the BASEFONT, if specified **<BIG>**Some Text**</BIG>**
<SMALL>	Makes text smaller, usually by one size below the default, or the BASEFONT, if specified **<SMALL>**Some Text**</SMALL>**
<BLOCKQUOTE>	Used for setting off a block of text from the main body of text. **<BLOCKQUOTE>** To be or not to be, that is the question. Whether ‘tis nobler in the mind to suffer the slings … **</BLOCKQUOTE>**

<PRE>	Use to retain formatting. For example, when you've copied lots of text from a word processor, and don't want to insert lots of paragraph tags. **<PRE>** Some text Some text Some text Some text Some text Some text Some text Some text Some text Some text Some text Some text **</PRE>**
<DL> </DL>	Used to set up a definition list. The definition list is useful for things like glossaries, and even CV's Two other tags go with the <DL> tag, <DT> and <DD>. The DT tag is the Definition Text, and the DD tag is the Definition Description. You use them like this: <H1>A CV</H1> **<DL>** **<DT>**<B>2345 – 2346</B> **<DD>** Accidentally Blew up the Starship Enterprise **<DT>**<B>2346 – 2347</B> **<DD>** In Prison for accidentally Blowing up the Starship Enterprise **</DL>**

And now that you have a good idea of how to format text on your page, here's another Review.

Review Number Three

Read these two letters below.

Dear Sir or Madam,

It has come to my attention that our local council has decided to close down our one and only library. While it is true that no new books have been bought for seven years, £12.50 was recently raised in a local raffle. This is nearly enough for an entire Delia Smith! Do you not think that your decision is a little premature?

Yours truly

Mr Irate

The council of the tiny village of Evercrease had this to say in reply

Dear Mr Irate,

The council did not take the decision to close the library lightly. It was a difficult decision, and one based entirely on financial priorities: constable Tucker must have his salary; Firewoman Jones must have her salary; and Ambulance Driver Henderson certainly must be paid for her heroic duties. Not forgetting, of course, the enormous cost of Mr Cobble's refuse collection. There is, quite simply, no more money left for libraries.

As for the subject of the £12.50 raised towards a Delia Smith, the council feels that there are enough of these cookery books in existence already. There is simply no need to purchase another. If you are lacking in the Delia Smith department, my chauffer can easily lend you his copy.

I hope that this matter is now closed.

Your truly,

Councilman Todd

Needless to say, Mr Irate does not feel that the matter is closed. Write his reply for him, in the form of a web page. Your web page must have the following:

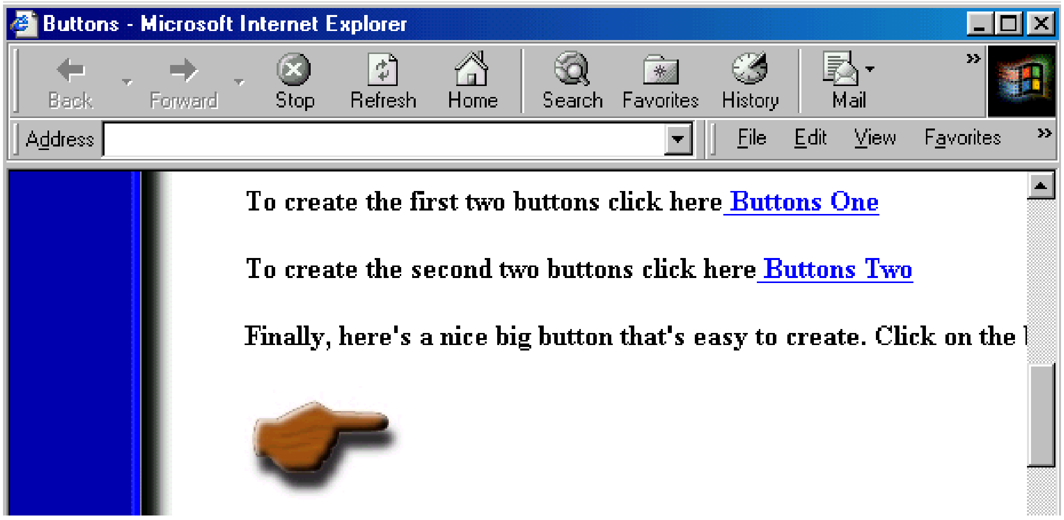
- An address aligned to the right
- A List
- Centred text
- Some coloured text

- Some bold text
- Some italics
- A change of font size and font face
- Underlined text

You can use whatever colour or image you like for the background. Your reply letter can be as serious or as witty as you like.

Hyperlinks

Hyperlinks are the backbone of the Web. They provide a means to connect one piece of information (a web page, for example) to another piece of information. If you have designed two web pages, a hyperlink will provide a quick way to jump from one page to the other. If you have one long web pages, a hyperlink can be used as a bookmark to help people jump from one part of the page to another, and back again. This is what a hyperlink looks like on a web page, the blue underlined text:



To create the first two buttons click here Buttons One

To create the second two buttons click here Buttons Two

Finally, here's a nice big button that's easy to create. Click on the

But let’s discuss what happens when the link is clicked on. (You can skip this discussion, if you like: it’s not mission critical!)

What is a Hyperlink?

When you click on one of those links (assuming you are connected to the internet), either Buttons One or Buttons Two, this is what happens.

Your browser gathers the information about the link and sends the request to something called a naming server. The naming server translates the link text (www.homeandlearn.co.uk, for example) into a series of numbers. These numbers are called the IP address. These are needed because computers don’t speak in a written language. So the computer needs something it can understand. An IP address is a set of four numbers separated by full stops. Each set of numbers is between

0 and 255. So when you click the link, the text address will be translated into an IP address, something like **213.209.156.97.**

The IP address will be used to identify a particular computer. If the computer, usually the naming server, doesn’t have the address in its database, it will pass the address further up the naming server food chain. If no naming server can find the IP address, the failure is passed back down to your browser. At this stage you’ll probably see a 404 error message.

If the address is found, however, the IP address is sent to your browser. The browser then contacts the web server that has the web page you requested. The page is then sent to your browser. However, that’s not the end because requests are done one at a time. If the web pages has images, the browser will see this and then request that the images be brought back to the web page as well. One image at a time.

Uniform Resource Locator

A uniform resource locator, or URL, is commonly called an address. The URL of our web Page is http://www.homeandlearn.co.uk/ Let’s break this down a bit.

http://www.homeandlearn.co.uk/

http://	**http** stands for hypertext transfer protocol. A protocol is a set of standards that one computer uses to speak to another. There are quite a lot of different protocols. For web communications, the two most common protocols are hypertext transfer protocol and file transfer protocol (FTP). There’s another layer of protocols underneath this called Transmission Control Protocol/Internet Protocol (TCP/IP). The http in a web address is followed by a colon and two forward slashes.

http://www.homeandlearn.co.uk/

www.homeandlearn.co.uk	This is the Domain Name, the part that gets translated into an IP address. The domain name is split into three parts, separated by full stops: **WWW** – the host name **homeandlearn** – the enterprise domain name **co.uk** – the top-level internet domain name. Others are **.com** (commercial), **.org** (organisation), **.gov** (government)

http://www.homeandlearn.co.uk/

/	If no html page is specified, the forward slash tells the server to look for the default web page. This is usually index.html. The index.html page is one that you have created. That's why naming the first page of your internet site index.html is so important.

OK, we've learnt what a link is, and what happens when you click one, but let's see how to create them.

*If you skipped the discussion, rejoin here.

Anchors

Linking in HTML code is done with the anchor tag, the <A> tag. The letter "A" in the tag is then followed by an attribute. For a link to another web page, the "A" is followed by "HREF". To set a bookmark in the same page, the "A" is followed by "NAME".

Links to other web pages

This is a link to the popular search engine Google:

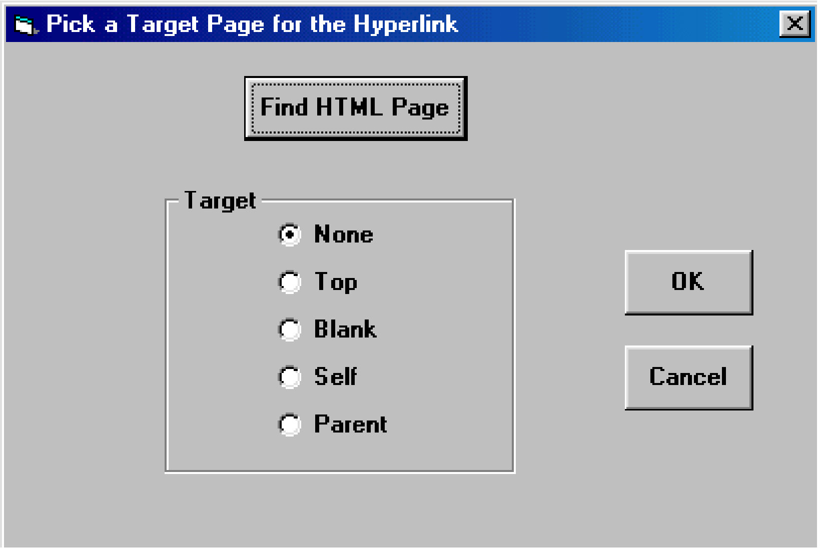
<A HREF = http://www.google.com/>Google Search Engine</A>

Notice where all the angle brackets (< >) are in the link. After the first one, we have the “A” part of the tag. Then we have the “HREF” part, signifying a link to another web page. After that comes an equals sign (=). After the equals sign comes the address of the web page itself. The address is case sensitive, so if there is a capital letter in the address, make sure to include it. This address **www.google.com** is different from this address **www.gOOgle.com**.

After the address comes the right angle bracket (>). Next comes the text that people see, the text you want them to click on. To close an anchor link, you use the end anchor tag. Which is this: **</A>**

Let’s get some practical work done.

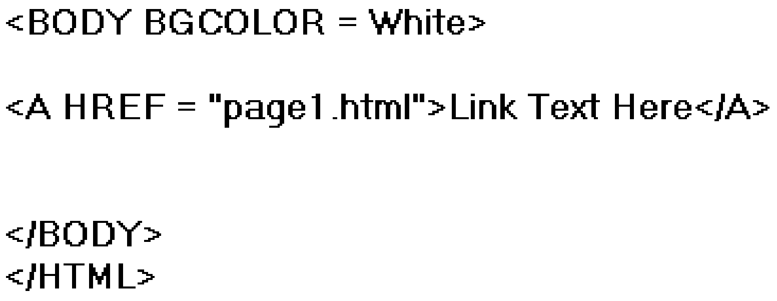
- Load up your HTML Editor and create a New web page
- Click inside the two body tags, so that the cursor is flashing there
- From the menu bar, click on **Insert > Link > Target Hyperlink**. The hyperlink dialogue box appears:



Target
● None
○ Top
○ Blank
○ Self
○ Parent

We’ll get on to the Target part of the dialogue box shortly, but what the screen is asking you for is the name of the web page that will appear in the browser when the link is clicked. In other words, the link itself.

- Click on **Find HTML Page** to locate a web page on your own computer
- The Windows Open dialogue box appears
- If you click the black down arrow next to Files of Type, you'll notice that two types are used: .**html** and .**htm**. So if you can't find a page you're looking for, switch to the other type
- Locate the web page you want to link to
- Click Open to get back to the Hyperlink dialogue box
- Click OK
- Your code will now look something like this:



```
<BODY BGCOLOR = White>

<A HREF = "page1.html">Link Text Here</A>


</BODY>
</HTML>
```

- Change the default "Link Text Here" to anything you like
- Save your web page and view the results in your browser
- Click on your link to make sure it works

Actually, there's a good chance that your link didn't work! That's because the HTML editor uses Relative referencing for hyperlinks. This will be a lot better for you, when you come to upload your web pages on the internet. If Absolute Referencing was used, your links would be like this:

C:\Webpages\links.html

That would be no good on the internet, for the obvious reason that the other person's browser would look on that person's computer for the web page that you created.

So a Relative Reference is used. The relative reference in the example was this:

"page1.html"

With this type of reference, the browser will look for the web page in the same folder that you saved your work to. If **page1.html** is not in this folder, the browser won't look anywhere else. It will just give you an error message.

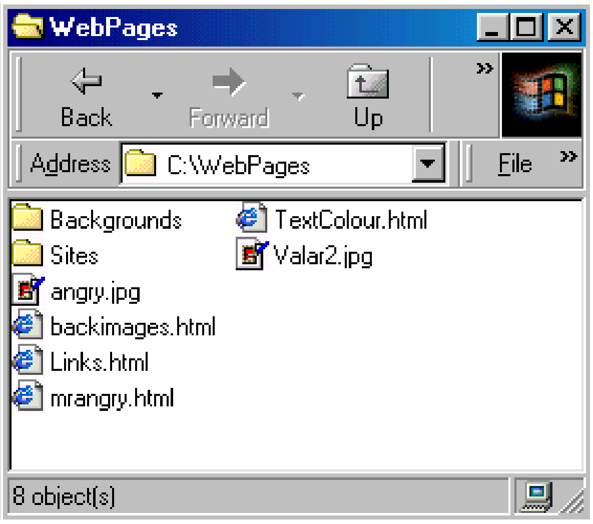
To solve the problem, either copy the page you're linking to in the same folder as your current web page, or type in the absolute reference.

Another solution is to use the forward slash before the web page, and precede that by the name of a the folder where the web page is. Like this:

"Sites/page1.html"

Now, the browser will go up one folder from where the current web page is, and look for a folder called "Sites". It will then search this Sites folder for the page called **page1.html**

To illustrate this graphically, take a look at this:



This is showing all the files and folders inside the folder called WebPages. We have two folders inside the WebPages folder: Backgrounds and Sites. Four .html files are showing and two JPEG image files. The page with our newly-created hyperlinks on it is called Links.html . The link itself, the one the HTML editor inserted, was this:

"page1.html"

With Relative Referencing, the browser will look for this page in the same folder as our Links.html page. This page is in the WebPages folder. There are only three other web pages in this folder: backimages.html , mrangry.html and TextColour.html . The browser won’t look anywhere else for page1.html. It won’t search the Backgrounds folder and it won’t search the Sites folder. However, if the link was this:

“Sites/page1.html”

Then the browser would look for a folder called “Sites”. But it will only look for this folder inside the WebPages folder , the folder where our current page is, Links.html .

But there is a folder called Sites, so the browser will search that folder for a page called page1.html. If the page is there, we have a successful link; if it’s not there, the browser will give up searching and return the error page.

The simplest solution for you is to save all your web pages and images into one folder. Then when you’re inserting a link to another web page, link to one of these web pages in the folder you create. That way, all your links will work.

For an absolute reference, we could have used this:

<A HREF = C:\Webpages\page1.html>My Text</A>

You can do that kind of thing as well. But careful where you’re pointing: if you upload a link like that to the internet, it won’t work. And if you send in work to be marked with a link like that, the link won’t work, and you’ll get your Review sent back for you to try again.

Links to Pages on the internet

The code for a link to somebody else’s web page that is on the web is similar to what you’ve just used. This time, you just add the web site address as the link. Like this:

<A HREF = www.google.com>Google Search Engine</A>

A quick way to insert a Hyperlink using your HTML editor is this:

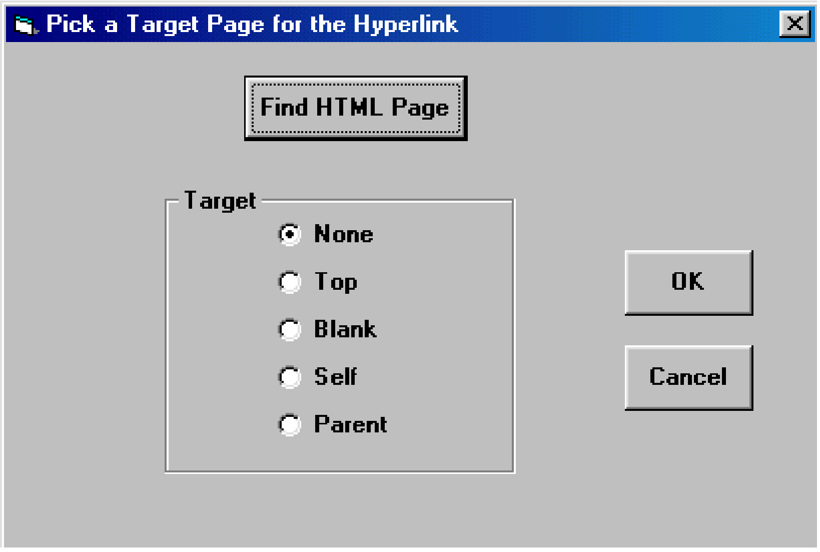
- Click where you want the link to appear
- From the menu bar, click on **Insert**
- From the drop down menu, click **Link**
- From the sub menu that appears, click **Blank Hyperlink**
- A Link like this will be inserted into your code

<A HREF = “”></A>

Type your link between the two double quotes. The text that people click on then goes between the angle brackets (> <).

Target Hyperlinks

You will have noticed on the Hyperlink dialogue box, that there was an area called Target.



Target
● None
○ Top
○ Blank
○ Self
○ Parent

The default is None. The other four you will meet when we move on to Frames. But the Blank option can be quite useful. If you have a link on your web page to somebody else’s site, you can have their web page open up in a brand new browser window. That way, your visitors won’t be leaving your site to view somebody else’s. Try out the Blank option and see how it works.

Bookmark Links

A bookmark link (commonly called an Anchor link) is useful when your web pages is rather long, and users have to scroll down to read it all. You can insert Bookmark links to aid navigation. When users click on your links, they will jump to different section of your web pages. You can even use a bookmark link to jump to a different web page, and back to the same spot where they left.

There's an example of a book-marked web page in the files you downloaded at the start of the book, called BookmarkExample.html (in the HTML folder). The bookmark link itself is at the bottom of the page and reads "Back to the Top". Here's how they work.

There are two parts to the bookmark: The clickable link itself, and the place where you want to jump to.

The place where you want to jump to, the destination for the click, again uses the <A> tag. This time, the added attribute is not HREF but NAME. You then surround some text or image with the tag. Like this:

<A NAME = "section1">In**</a>** this first section, we'll discuss Links

We've surrounded the word "In" with our destination bookmark. The name itself, the part after the equals sign, can be anything you like. But you'll use that name in the Link part of the bookmark. Here's the actual link, the part people see and click on.

<A HREF = #section1>Click here for Section One**</A>**

Note that we're back to the HREF attribute. This time, after the equals sign, there is a hash symbol (#). After the hash symbol you type the NAME you used in step one. You then type the text that people will click on. Finally, you close the tag with **</A>**.

You can bookmark to another web page, if you like. In which case, the link would be this:

<A HREF = page2.html#section2>Click here for Section Two**</A>**

Note where the hash symbol is now –after the name of a web page. The NAME of the destination bookmark follows the hash symbol. There is no spaces between the two.

The destination link itself would then go somewhere on page2.html. If you had another bookmark on page two, you could have the user jump back to the same spot where they left.

To insert a bookmark using your HTML Editor, do the following:

- Click where you want the bookmark to appear
- From the menu bar, click on Insert
- From the drop down menu, click **Link > Bookmark Link**
- The Bookmark Link dialogue box appears
- Type in the Name for your bookmark, and click OK

The HTML Editor will only insert the basic tags for you, in this format:

<A NAME = Section1></A>

<A HREF = #Section1>Bookmark Link Text</A>

There is still some work left for you to do. But it's quite easy. Cut and paste the two sections where you want them. Remember: The NAME part is the destination, where users will end up when they click on the HREF part.

For the NAME part, you need to surround some text or an image. Like this:

<A NAME = Section1>**Your Text Here**</A>

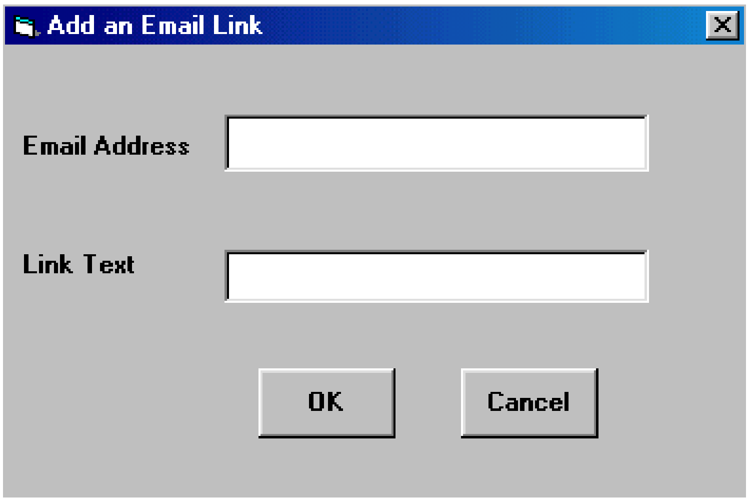
For the HREF part, just paste it somewhere in your web page, and replace the default "Bookmark Link Text" with your own.

Email Links

An email link can be added to web pages, too. When people click on this email link, their Email software will start up. In the To box of the email software will be your email address.

To add a email using your HTML Editor do the following:

- Click on **Insert** from the menu bar
- From the drop down menu, click **Link > Email Link**
- The Insert Email Address dialogue box appears, and looks like this:



- Type your email address in the first text box
- In the Link Text box, type the text that people will click on to send you an email
- Click OK
- The editor will insert code something like this one:

```
<A HREF = "MailTo:online@homeandlearn.co.uk">Email us</A>
```

Notice that the link is the HREF type. After the equals sign you have this:

Mailto:

Mailto is all one word, and is followed by a colon. Then you put your email address after the colon, without typing a space. You then type a right angle bracket (>) before typing the text you want people to click on to send you an email. The Anchor closing tag </A> finishes it all off.

And that's it for email address – quite simple! If you want to centre the email address, highlight the entire email code and click **Format > Centre**. Or centre it with a paragraph tag. Like this:

```
<P ALIGN = Center>
        <A HREF = "MailTo:online@homeandlearn.co.uk">Email us</A>
</P>
```

Try the email code out in a web page, save your work and see what happens when you view the web page and click the email link.

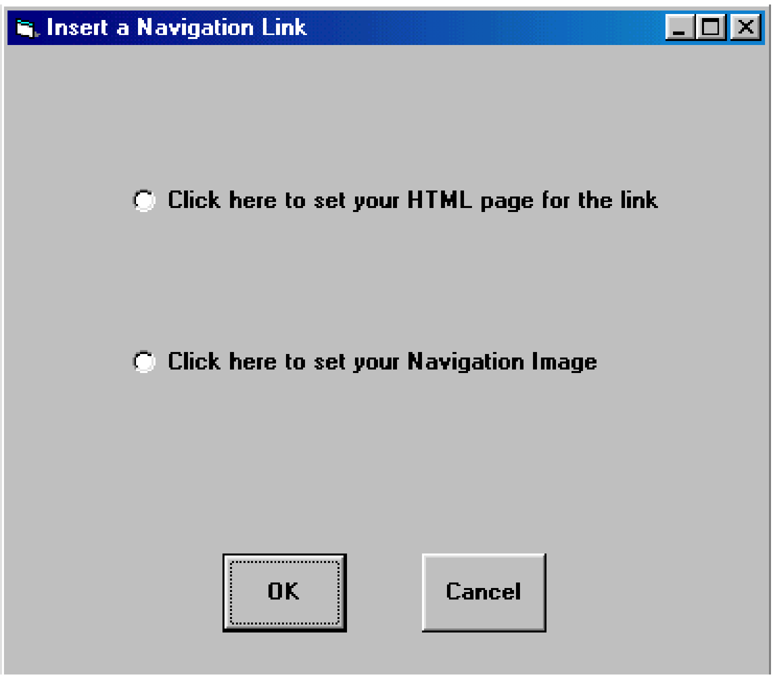
Using Navigational Icons As Links

You can use images as navigational icons on your web page, to help your viewers get around your site. There is a web page in the files you downloaded at the start of the book called NavIcons.html (in the Navigation folder). Load it into your browser to see an example of what navigational icons are.

You can copy any of the images in the Navigation folder to your own Web Pages folder, and use them on your site, or in the tutorial that follows.

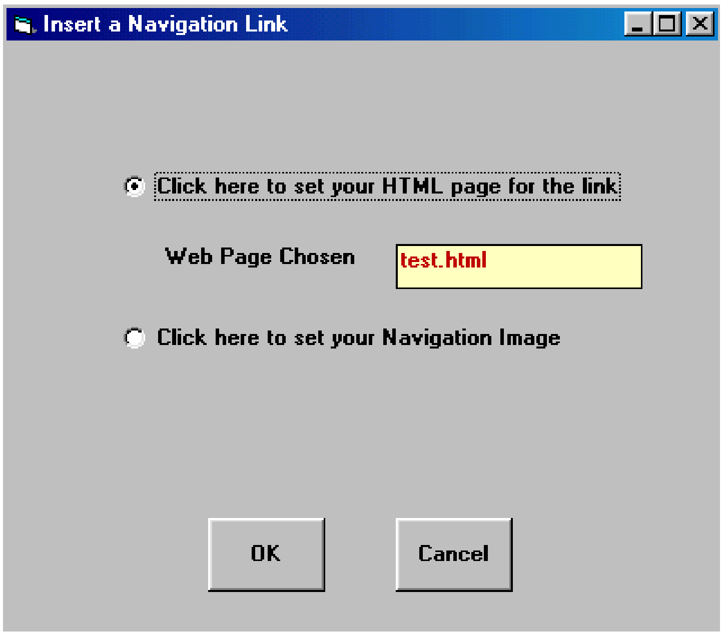
To insert a navigational link with your HTML Editor, create a new page and do the following:

- Click anywhere in between the two BODY tags
- From the menu bar, click **Insert**
- From the drop down menu, click **Link** > **Navigation Link**
- The Insert a Navigation Link dialogue box appears, and looks like this:


Insert a Navigation Link
○ Click here to set your HTML page for the link
○ Click here to set your Navigation Image
OK
Cancel

Inserting a navigation link is a two step process: setting the HTML page the people will go to when the link is clicked; setting the image that is clicked on. So:

- Click on the first of the options on the dialogue box (actually, the order you click on doesn’t matter. You can click the Navigation Image first, if you prefer)
- The Open HTML page dialogue box appears
- Locate the web page you want to use as a link, and click OK
- The dialogue box will now look like this:



Insert a Navigation Link

● Click here to set your HTML page for the link

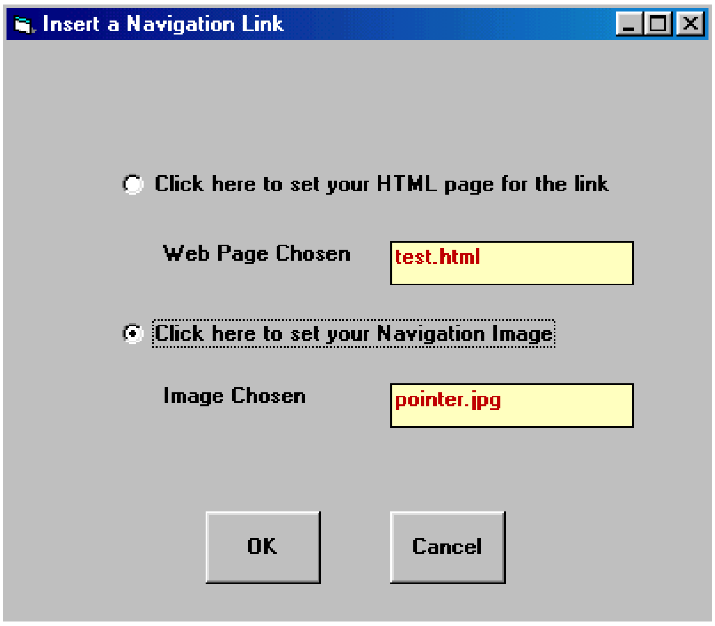
Web Page Chosen test.html

○ Click here to set your Navigation Image

Next, you need to select an image to use. The image will be the thing that is clicked on.

- Click on the second option, “Set your Navigation Image”
- The Open Image dialogue box appears
- Locate the image you’re going to use for navigation
- Click Open
- The Open Image dialogue box disappears, and your Insert a Navigation Link dialogue box will look like this:

Dialogue Box



○ Click here to set your HTML page for the link

Web Page Chosen: test.html

● Click here to set your Navigation Image

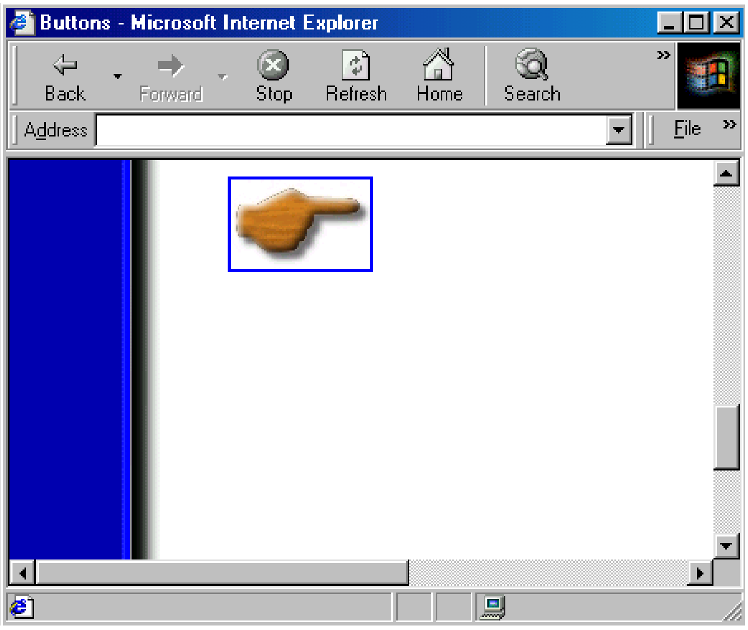
Image Chosen: pointer.jpg

When you have BOTH your web page and image set, click OK. The code will be inserted onto your page, and will look like this:

```
<A HREF = test.html>
        <IMG SRC = "pointer.jpg">
</A>
```

As you can see, the image code has gone in between the Anchor link code. With the image surrounded by the link, the image itself will then be the link. All you're doing is setting a picture to use as a link instead of some text. Once again, though, the HTML Editor uses Relative Referencing, so careful where you're pointing! If you get a red X instead of an image, make sure the web page and the image are in the same folder.

If you're successful, you might have a page that looks like this one:



Notice the blue square around the image. That's the HREF part of the code doing that, to indicate that the image is a link. What it's doing is putting a border around the image. You can remove the border by adding the BORDER attribute to the image code. Like this:

```
<A HREF = test.html>
        <IMG SRC = "pointer.jpg" BORDER = 0>
</A>
```

A space is typed after the image name. You then type BORDER = and follow that by a number. The number 0 means no border. If you want a thick border, experiment with the Border number, see what happens.

The default border colour is Blue. But you can change that. Unfortunately, there's no Border Colour attribute that you can add to the Image tag. To change the colour of the border, you have to remember that it is the Anchor link that is putting the border around your image. So the thing to change is the Link Colour of the Anchor tag.

You set the colour of links with the LINK tag. There are three of them, and they usually go in the BODY tag. Here's the three LINK tags:

LINK	Set the colour of a link before it has been clicked on

ALINK Set the colour of a link when the link is clicked on

VLINK Set the colour of a link after it has been clicked on

You can put all three in the BODY tag. Like this:

<**BODY LINK** =Blue **ALINK** =Red **VLINK** =Red>

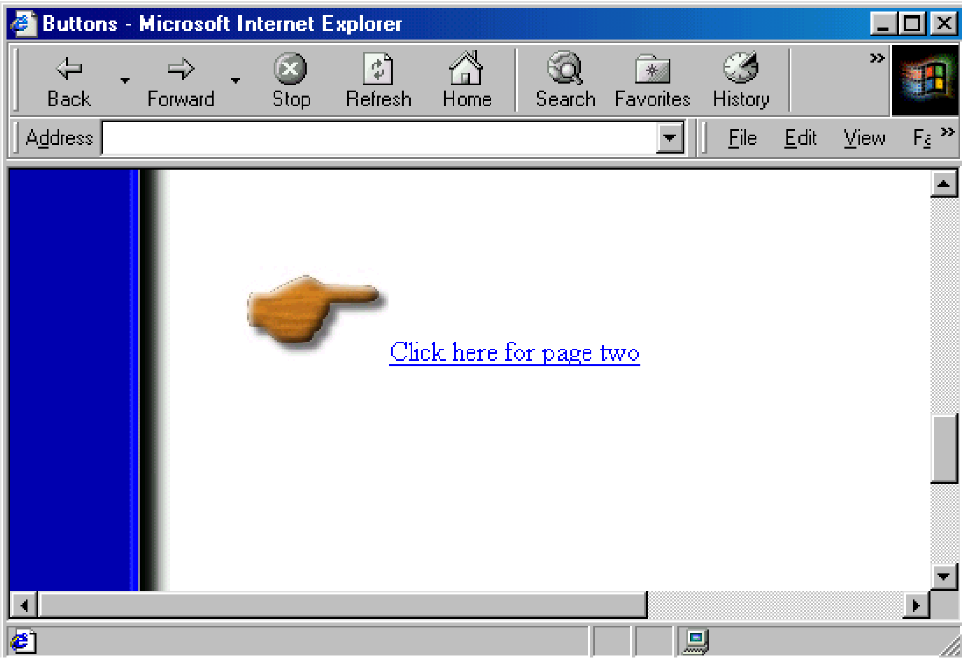
Here, we’re setting all links that have not been clicked on to blue. When a user clicks on a link, and after the link has been click, the colour will be red.

However, setting link colours often has mixed results. That’s because the user can manipulate the browser settings to override the colours you specify.

Some text can be added along side your navigation icon, for those who have images turned off in their browsers. Like this:

<A HREF = test.html>
 <IMG SRC = "pointer.jpg" **BORDER = 0**>Click here for page two
</A>

The end product might be a web page that looks like this one:



Buttons - Microsoft Internet Explorer

Click here for page two

Experiment with the navigation images you downloaded. Use navigation icons sparingly, though: too many and you not only irritate your visitors, but the entire page will take an eternity to download over a 56k modem.

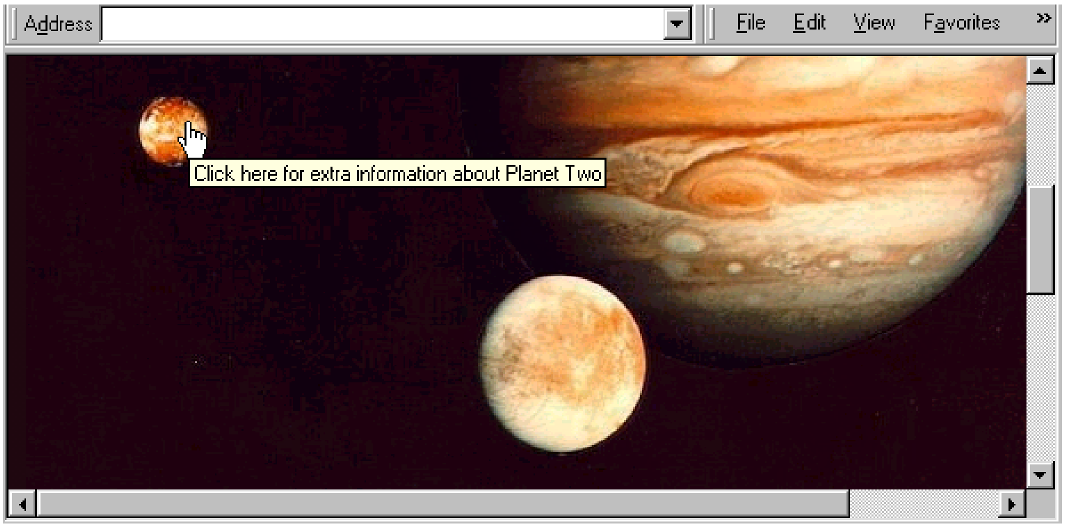
Image Maps

To help your visitors find their way around your site, you can use something called an image map. An image map is a graphic on your web page that has been split into different areas. Each area is a separate link. There is a web page in the files you downloaded that has been turned into an image map. The page is called **imagemap.html** and is in the folder called "imagemap". Load it into your browser to get a good idea of how they work.

(The image, by the way, is far too big in size to go on the internet. You wouldn't use this one on a web page. At 250 kilobytes in size, your visitors would be long gone before it's finished loading.)

When the web page is loaded into your browser, hold your mouse over one of the planets. You should see a little pop-up box with some more information.

The pop-up box will look like this one:



Click here for extra information about Planet Two

Only the planets are clickable links. Move your mouse on a black area, and nothing happens when you click. Click on a planet and the idea is that you would be taken to a new web page (though you stay on the same page with this image map).

You'll now learn how to construct your own image map. Unfortunately, your HTML Editor will not insert the code for you, so this is something you'll have to do on your own. You can, of course, use your editor to start a new html page and add the code by hand. Off we go then.

An image map consists of two parts: the actual image itself, and the coordinates to use for the links. Both parts go between the two BODY tags. Here's the code for the image:

<IMG SRC = imagemap.jpg **USEMAP = #planets** border =0>

You met most of that code before. The only thing you haven't met is the **USEMAP** attribute:

USEMAP = #planets

The word USEMAP is followed by an equals sign, then the hash symbol (#). After that comes the name of the image you want to use. This name is something you pick yourself, and can be anything you want. The name is then used for the second step of the process.

The second step of the process is where you set up the coordinates of your map. In the web page on the disk, we wanted only the planets to be links to web pages. So we need to "tell" the browser where the planets are. This how you "tell" the browser which parts of your image you want to turn into links:

<MAP NAME = "planets">

<AREA **SHAPE** = "" **COORDINATES** = "" **HREF** = "" ALT = "">

</MAP>

Only the basic tags have been inserted. Here's an explanation.

<MAP NAME = >	Here is where you add the name from the USEMAP part of your image. We called our map "planets". (The name is case sensitive.)

<AREA	Set up an area to use for your map. You need one for every part of the map. We used five areas in our image map. So five lots of area tags were needed.

SHAPE =	You have three choices here: Rectangle, Circle, and Polygon. The actual code for the three is:
	SHAPE = "rect"
	SHAPE = "circle"
	SHAPE = "poly"
	We used four circle shapes and one rectangle shape (Yes, you can mix them).

COORDS	Shorts for Coordinates. The coordinates can be quite tricky to set up. The coordinates for a rectangle use this format: **COORDS = "x1, y1, x2, y2"** The first two (x1 and y1) are the coordinates of the top left of your rectangle; the second two (x2 and y2) are the bottom right of your rectangle (we'll see an easy way to get these coordinates shortly.) The coordinates for a circle use this format: **COORDS = "x, y, r"** The x and y are the centre of the circle, and the r is the radius. The coordinates for a Polygon use this format:

	COORDS = “x1, y1, x2, y2, x3, y3” Each pair of coordinates is a point on your polygon
HREF =	This is where you add the name of the web page that people will betaken to when they click an area of your image map
ALT =	Set the text you want to use when people hold the mouse over you link
</MAP>	Finish the whole thing off with the end map tag.

The code from the planets image map was this.

```
<BODY BGCOLOR = #210711>

<MAP NAME = planets>

<AREA SHAPE = "circle" COORDS = "430, 210, 100" HREF = "imagemap.html" alt = "Click here for extra information about Planet One">

<AREA SHAPE = "circle" COORDS = "80, 243, 20" HREF = "imagemap.html" alt = "Click here for extra information about Planet Two">

<AREA SHAPE = "circle" COORDS = "135, 580, 80" HREF = "imagemap.html" alt = "Click here for extra information about Planet Four">

<AREA SHAPE = "circle" COORDS = "300, 377, 50" HREF = "imagemap.html" alt = "Click here for extra information about Planet Three">

<AREA SHAPE = "rect" COORDS = "520, 580, 650, 760" HREF = "imagemap.html" alt = "Click here for extra information about Planet Five">
```

```
</MAP>

    <img src = imagemap.jpg USEMAP = #planets border =0>

</BODY>
```

Notice that the MAP part comes first, and the Image second. Let’s take the code for the first planet, and see how it works:

```
        <AREA SHAPE = "circle" COORDS = "430, 210, 100"
HREF = "imagemap.html" ALT = "Click here for extra
information about Planet One">
```

After typing the word AREA we then set the shape we want to use:

```
SHAPE = "circle"
```

The coordinates of the circle come next:

```
COORDS = "430, 210, 100"
```

The first number, 430, is how far left the centre of the circle is on the image (the measurements are all in pixels). The second number, 210, is how far from the top the centre of the circle is on the image. The third number, 100, is the radius of the circle.

Next, we have a HREF attribute:

```
HREF = "imagemap.html"
```

This is quite simply the name (and address) of the web page that people go to when our circle link is clicked on.

Finally, we have the ALT attribute:

```
ALT = "Click here for extra information about Planet One">
```

If you hold you mouse over the first planet, you’ll see this text pop up in a little box. Notice that the right angle bracket (>) comes at the very end.

Four more AREA tags were needed, one for each or the other four planets. The final one uses a Rectangle as the SHAPE, and it's worth examining how it works. The code was this:

<AREA SHAPE = "rect" COORDS = "520, 580, 650, 760" HREF = "imagemap.html" ALT = "Click here for extra information about Planet Five">

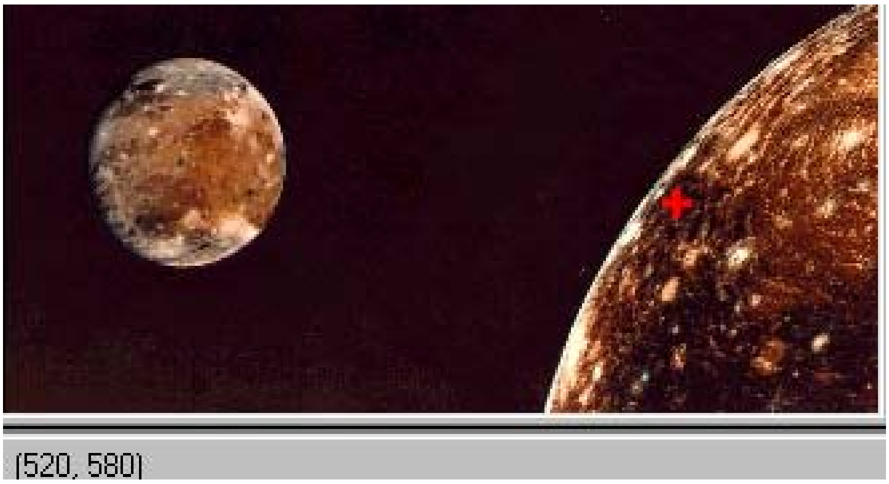
The HREF and ALT parts of the code are fairly straightforward. Notice that the name of the SHAPE is "rect" and not "Rectangle". (Like all attribute values, however, the "rect" is not case sensitive. So you can have it in Uppercase, lowercase, or a mix of both, if you wanted. The double quotes are not needed either.)

The Coordinates were these:

COORDS = "520, 580, 650, 760"

The first two, 520 and 580, define the top left hand corner of the rectangle. The 520 is how far left of the image the top left corner is; the 580 is the how far from the top the top left corner is. The final two coordinates, 650 and 760, define the bottom right of the rectangle. The 650 is how far left of the image the bottom right corner is; the 760 is the distance from the top of the image to the bottom right corner.

You can easily grab the coordinates of a rectangle, if you have image editing software, like Paint Shop Pro. Here's a screenshot from Paint Shop Pro 7 showing the coordinates of the top left hand corner of the rectangle:



(520, 580)

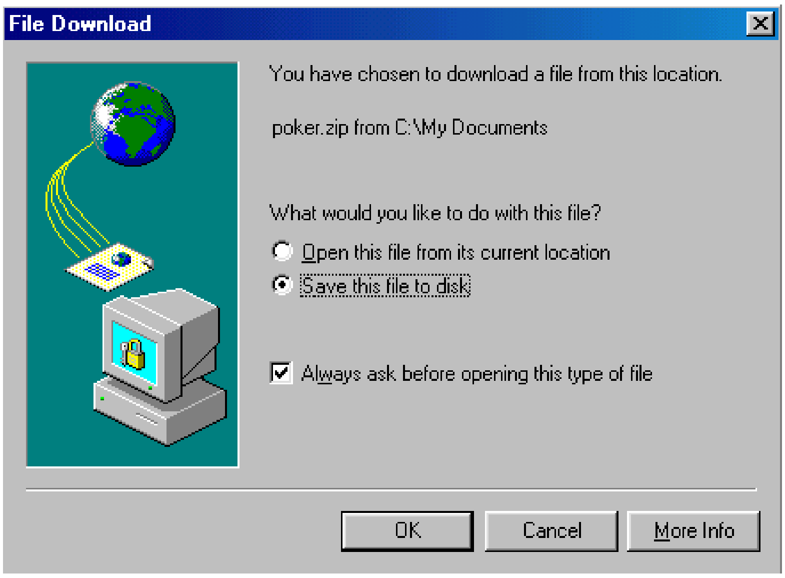
The red X in the picture represents the mouse pointer when the selection tool is selected. You move your mouse to the top left corner of the rectangle and the coordinates will show up in the bottom left of the screen, on the grey task bar (520, 580). Move your mouse to where you want the bottom right of your rectangle, and jot down the coordinates.

Links to other Files

You can place a link on your web page to files other than images and web pages. A link can be made to all sorts of different file types. For example, if you have recorded a piece of music and it's in the MP3 format, you can share it with the world by inserting this link:

<A HREF = "MyMasterpiece .mp3">Download my music</a>

When the link is clicked on, the browser will see the file extension **.mp3** and try to open it up. If you have Internet Explorer, the Windows Media Player will probably be started, and the MP3 file will open up in that piece of software. If the browser can't find a piece of software to open the file, it will give you this dialogue box (on a Windows 9x machine):


File Download

You have chosen to download a file from this location.

poker.zip from C:\My Documents

What would you like to do with this file?

○ Open this file from its current location
● Save this file to disk

☑ Always ask before opening this type of file

As you can see, you are giving the option to either open the file from where it currently is, or saving it to your own computer.

You don’t have to add anything special to have people download files like MP3’s, or video files, or zipped files. Just a normal link with the name of the file after the HREF part will do it.

And that concludes this section on links. Time for a Review.

Review Four

Design a web site with the theme of Space. There are some images in the files you downloaded at the start of the book to help you with this, but by all means use any image of your own.

Your web site must have at least three pages. The site must also include the following:

- At Least 5 Hyperlinks
- Of the 5, one must be an Email link
- All of your pages must link together (The first page should have links to page 2, and page 2 should link back to the first page.)
- At least one of your links must use an image The first page of your web site must be called index.html

Any background colour scheme (or background image scheme) can be used. Try using as many as possible of the HTML tags that you have learnt so far.

Good luck.

Tables

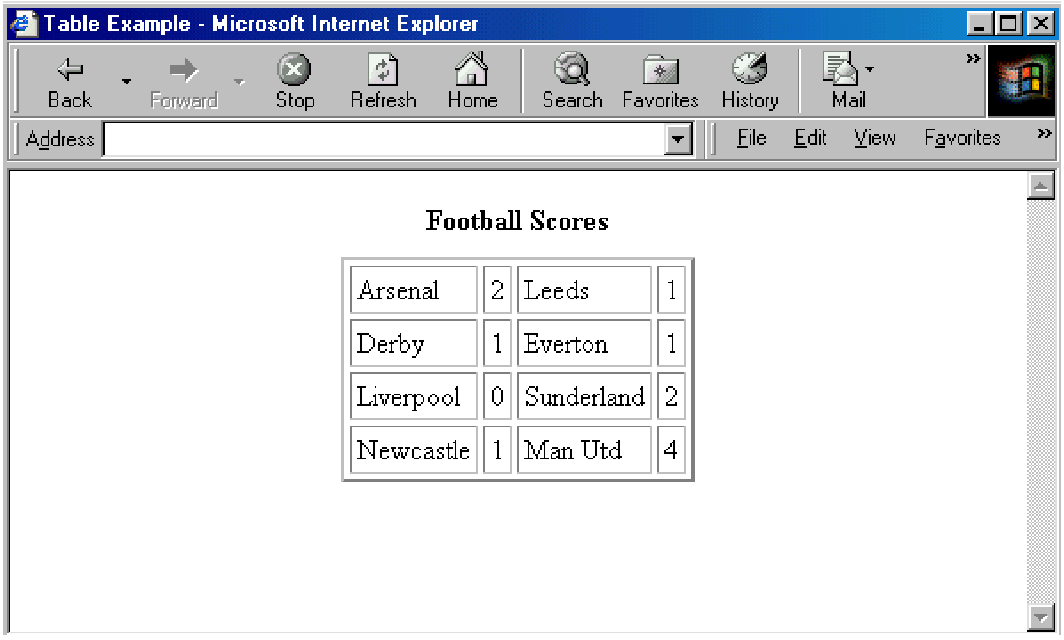
With HTML, positioning things like images and text where you want them to appear is next to impossible. Trying to position an image in the centre of the page, with images diagonally to the right of it is something you just can't do in straight HTML – unless you use Tables.

Tables were introduced to HTML as a way to make textual data look more presentable on the screen. Things like statistics could be presented in neat rows and columns, making them easier to read. Tables are still used for that purpose. But the fact that cells in tables can be coloured, height and width changed, images placed in cells, makes them far more handy than just a data presentation tool.

In this section, you'll learn how to do both: present data using a table, and using tables to get more creative. As Tables can be quite complex to set up, your HTML Editor can be used to do most of the hard work. So let's make a start.

Data Presentation

We're going to use the HTML Editor to design this table:



Football Scores

Arsenal	2	Leeds	1
Derby	1	Everton	1
Liverpool	0	Sunderland	2
Newcastle	1	Man Utd	4

That's the basic table we'll start with. Then we'll see ways to tidy it up a bit. First, the basic table tags are these:

```
<TABLE>
    <TR>
        <TD></TD>
    </TR>
</TABLE>
```

The table tags come in pairs. To set up the table, the TABLE tag is used. There is a start tag and end tag. In between these two tags are the table Row tags <TR> </TR>. For every Row in your table, you need at least one Table Data tag <TD> </TD>. The Table Data tags represents the cells in each row. In the example picture, we had a table with four rows. In each row we had a home team, a score, an away team, and the away team's score. So each Row in our table had four cells in it. For one individual Row, the code would look like this:

```
<TABLE>
    <TR>
        <TD></TD>
        <TD></TD>
        <TD></TD>
        <TD></TD>
    </TR>
</TABLE>
```

That code means set up a table with one Row, and put four cells into the Row. That, in our example, would get us the Arsenal versus Leeds score (well, it would if we typed the data in).

The data itself, the text you want people to see, goes between the two TD tags. Like this:

```
<TABLE>
    <TR>
        <TD>Arsenal</TD>
        <TD>2</TD>
        <TD>Leeds</TD>
        <TD>1</TD>
    </TR>
</TABLE>
```

If we wanted another Row to our table, we would just add another set of TR tags, and four more TD tags. Like this:

```
<TABLE>
     <TR>
          <TD>Arsenal</TD>
          <TD>2</TD>
          <TD>Leeds</TD>
          <TD>1</TD>
     </TR>

     <TR>
          <TD>Derby</TD>
          <TD>1</TD>
          <TD>Everton</TD>
          <TD>1</TD>
     </TR>

</TABLE>
```

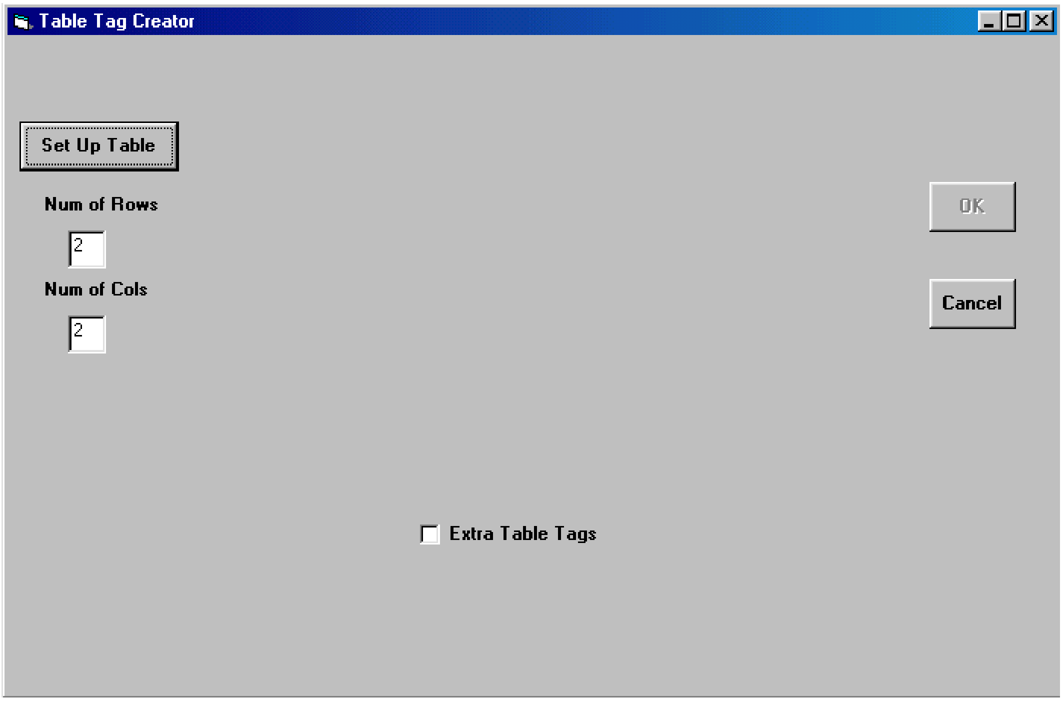
And that's a basic two Row table, with four cells in each row. There is no extra formatting added to the tags. The browser would not put a border around that simple table, nor would it add a caption.

Let's use the HTML Editor to set up that table, and add two more rows.

So, with your editor loaded, do the following:

- Click inside the two BODY tags
- From the menu bar, click on **Add**
- From the drop down menu, select **Table**
- The Table dialogue box appears, and looks like this:

Table Dialogue Box



Num of Rows

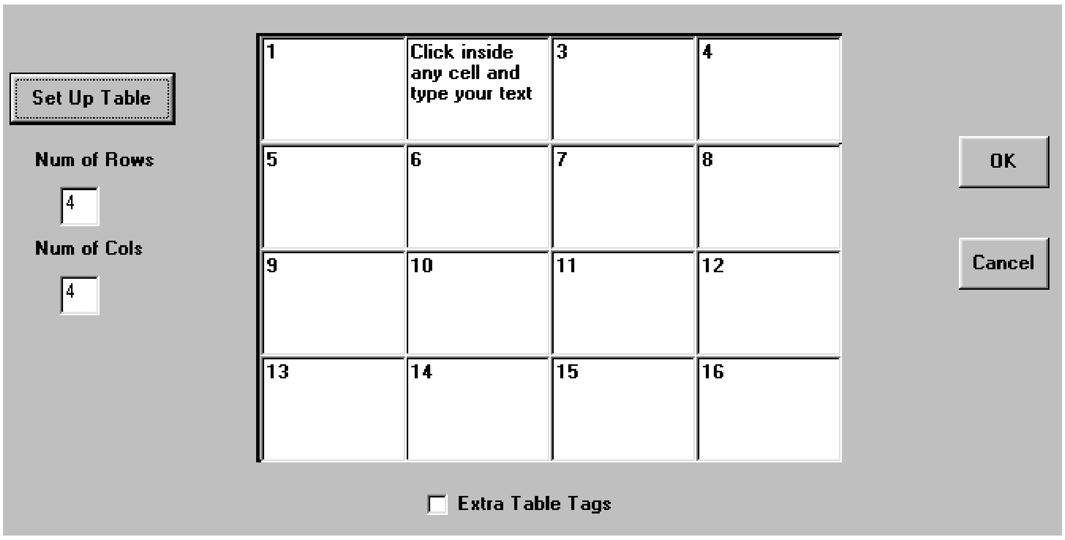
2

Num of Cols

2

The default is 2 Rows and 2 columns. (the TD tags are the columns in HTML tables, and the TR tags the rows.)

- Change both of the 2's in the Num of Rows and Num of Cols to 4's. Then click the **Set up Table** button
- Your editor will look like this:



Num of Rows

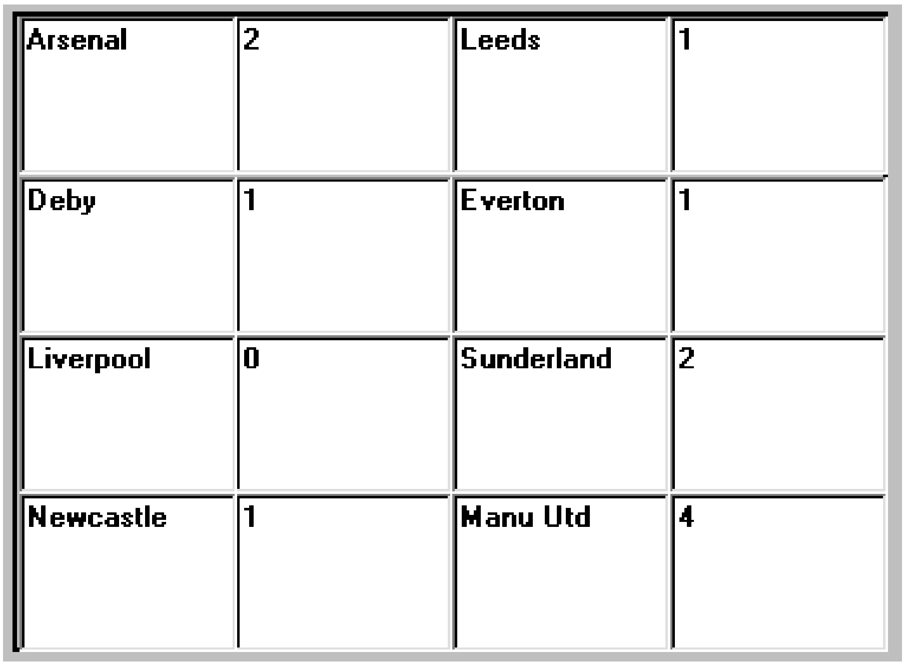
4

Num of Cols

4

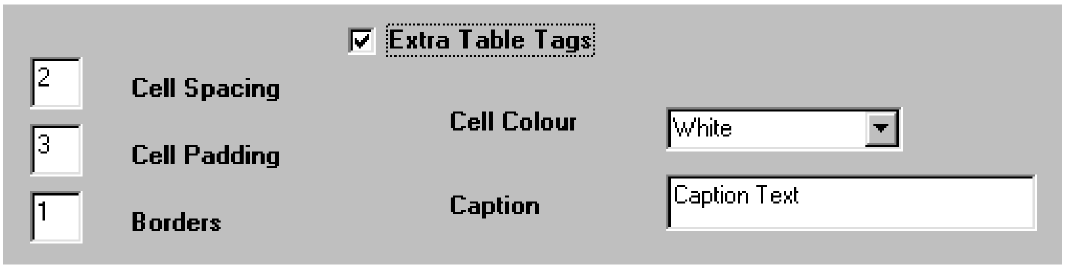
1	Click inside any cell and type your text	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Each of the white text boxes represent your TD tags. Type inside the text boxes, so that you have all four scores displayed. Like this one below:



Arsenal	2	Leeds	1
Deby	1	Everton	1
Liverpool	0	Sunderland	2
Newcastle	1	Manu Utd	4

You could go ahead and click the OK button. But click inside the box next to "Extra Table Tags". You should see the following:



☑ Extra Table Tags

2 Cell Spacing

3 Cell Padding

1 Borders

Cell Colour: White

Caption: Caption Text

The Border tag will obviously add a border around your table. To add a caption to your table, click inside the caption dialogue box. Erase the default text, and type your own. Cell Spacing will extend the walls of each cell; Cell Padding will add some space between the contents and the walls of each cell. The defaults are 2 and 3. But change them to something else, anything you like. We'll save the Cell Colour option till later.

When you're done, click the OK button. The editor will insert the code into your page.

Save your work and view the results in your browser. If you want to centre the table, just highlight the entire table code and click **Format > Centre**.

But let's look at where the extra tags have gone.

```
<TABLE  Border = 1 CellPadding = 3 CellSpacing = 2>
    <CAPTION>Caption Text</CAPTION>
    <TR>
       <TD BGCOLOR = White>Arsenal</TD>
       <TD BGCOLOR = White >2</TD>
       <TD BGCOLOR = White >Leeds</TD>
       <TD BGCOLOR = White >1</TD>
    </TR>
```

The Border, CellPadding and Cellspacing attributes have all gone in the first TABLE tag. The Caption tag comes after the TABLE tag, but before the first TR tag. Try experimenting with the Border and Cell Spacing/Padding tags. Change the border size to something like 6 or 7 and see what happens. The BGCOLOR = white part is the HTML Editor inserting the default background colour for all the cells. We'll see how to change this default colour later. For now, let's see what we can do to extend the Table tags.

Table height and Table width

You can change the size of the entire table by adding the Width and Height attributes to the TABLE tag. Like this:

<TABLE **WIDTH** =300 **HEIGHT** = 300>

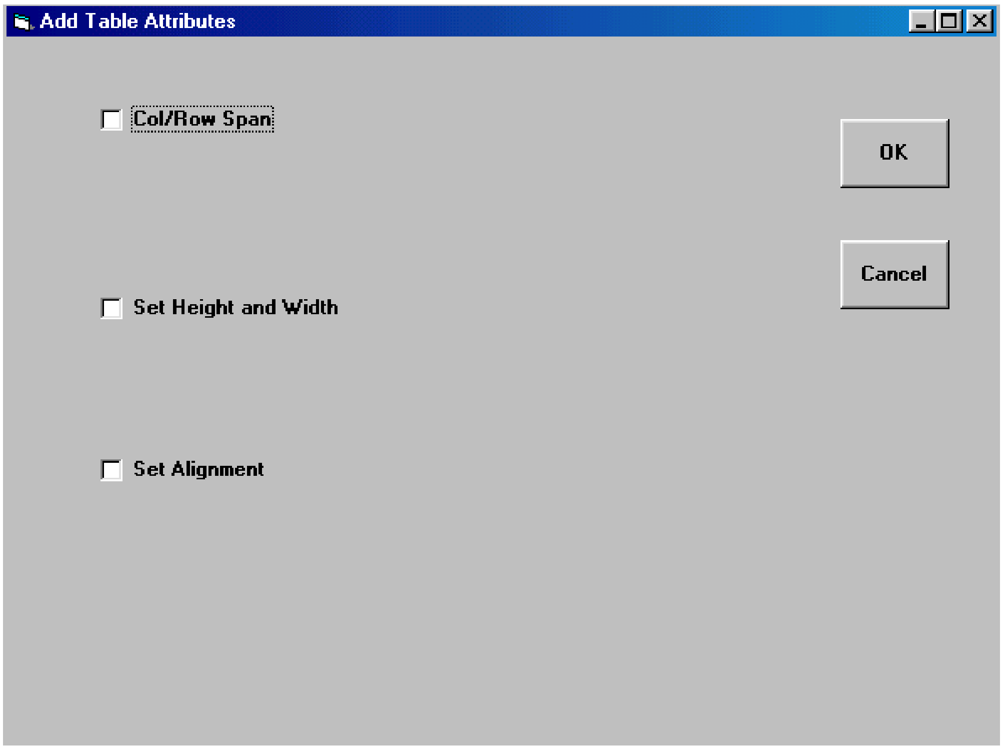
That changes the size of your entire table (the Border, CellSpacing and CellPadding tags have been left out for convenience sake. They are optional, anyway.) If you don't specify a Height and Width for your table, your browser will size the table based on the contents on the cells. The measurement above is in Pixels. A typical monitor size in pixels might be 800 by 600 or 1024 pixels by 768.

A better idea is to specify a table size as a percentage. Like this:

<TABLE **WIDTH** =100% **HEIGHT** = 100%>

The figure is the percentage of the screen you wish to take up with your table. The maximum is not 100%. Experiment with figures higher and lower than 100 and see what happens to your table. You can use your editor to add the Width and Height attributes for you. To add a Height and Width using the editor, do the following:

- Click after the “E” of “<TABLE
- Press the spacebar to add a space
- Click on **Format > Table Attributes** from the menu bar
- The following dialogue box pops up:

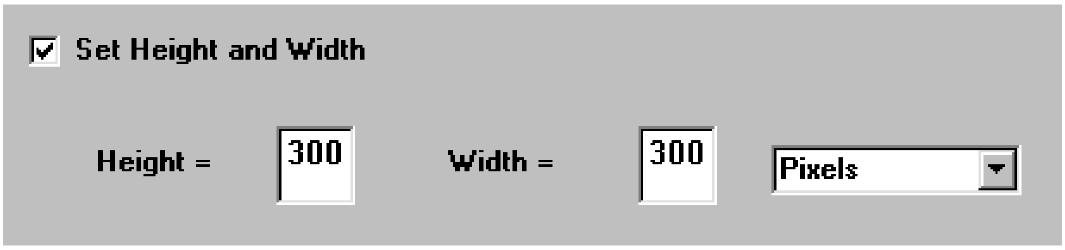


□ Col/Row Span

□ Set Height and Width

□ Set Alignment

- Click inside the box next to “Set Height and Width”
- The Height and Width options appear



☑ Set Height and Width

Height = 300

Width = 300

Pixels

- Type in a Height and Width for your table
- Choose either Pixels or Percent from the drop down box
- Click OK
- Height and Width are added to the <TABLE> tag

Row height and Row width

You can make changes to the Height and Width of not only the entire table, but to each individual cell, or row of cells. Just use your editor again to add the Width and Height attributes to the TR or TD cell. The result is this:

```
<TR Height = 50 Width = 100>
   <TD BGCOLOR = White>Arsenal</TD>
   <TD BGCOLOR = White >2</TD>
   <TD BGCOLOR = White >Leeds</TD>
   <TD BGCOLOR = White >1</TD>
</TR>
```

You can add the Height and Width attributes to individual TD cells, too, but the results are often erratic. If you want one big cell next to smaller cells, the ROWSPAN and COLSPAN tags are used.

ROWSPAN and COLSPAN

For a complex table of different cell size, you can use ROWSPAN and COLSPAN. This can get quite complex. But remember that Columns go down, and Rows go across. To have one big cell stretch horizontally across two smaller cells, the code would be this:

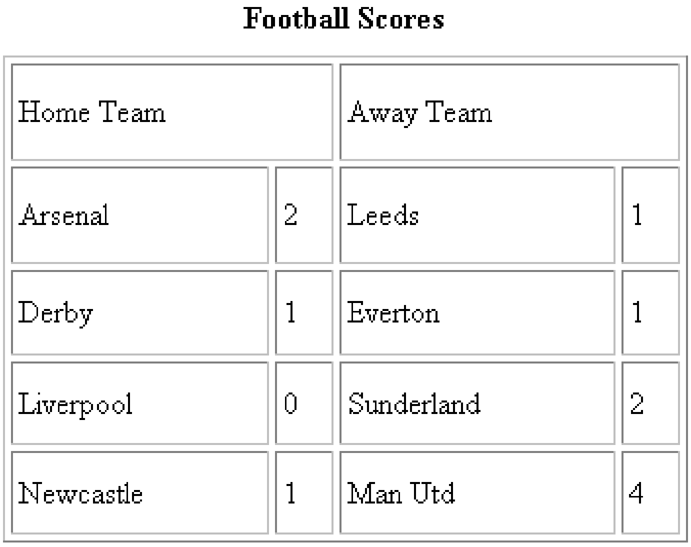
```
<TR>
      <TD COLSPAN = 2>Home Team</TD>
      <TD COLSPAN = 2>Away Team<BR></TD>
</TR>
```

The attribute COLSPAN has the value of two because one TD cell is going to stretch across two columns. We had 4 columns in our table, so the first TD tag will span two columns, and so will the second one. If we had some of the other table tags, you might get a better idea of what’s going on.

```
<CAPTION><B>Football Scores</B></CAPTION>

     <TR>
          <TD COLSPAN = 2>Home Team</TD>
          <TD COLSPAN = 2>Away Team<BR></TD>
    </TR>
```

```
<TR >
      <TD>Arsenal</TD>
      <TD>2</TD>
      <TD>Leeds</TD>
      <TD>1</TD>
</TR>
```

In a browser, it looks like this:



Football Scores

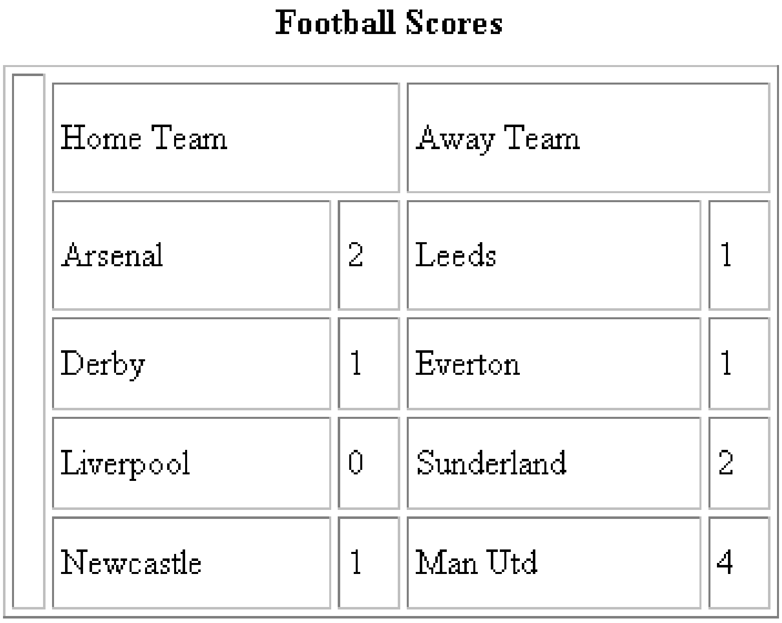
Home Team		Away Team	
Arsenal	2	Leeds	1
Derby	1	Everton	1
Liverpool	0	Sunderland	2
Newcastle	1	Man Utd	4

If we wanted one big cell to stretch over all our rows, running down the left side, we would use ROWSPAN. The code would be this (the <BR> tag gets you a blank cell):

```
 <TR>
      <TD ROWSPAN = 6><BR></TD>
</TR>
```

And this would be the effect:

ROWSPAN



Football Scores

	Home Team		Away Team	
	Arsenal	2	Leeds	1
	Derby	1	Everton	1
	Liverpool	0	Sunderland	2
	Newcastle	1	Man Utd	4

You might have noticed that although the code was this:

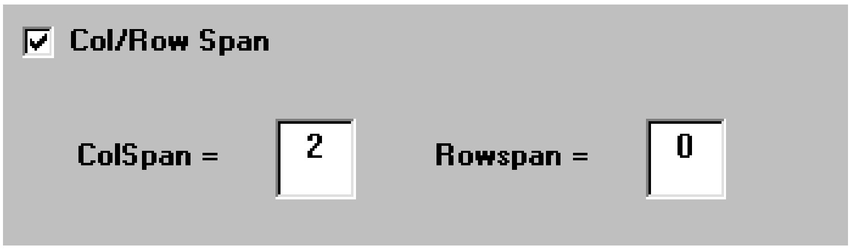
```
 <TR>
      <TD ROWSPAN = 6><BR></TD>
</TR>
```

the number of rows in the table was actually only five. So why set the ROWSPAN to 6? Well, it's because we had 5 sets of TR tags in our table, plus the one TR tag we added for the ROWSPAN making 6 in total. The code for the whole table now looks like this:

```
<TABLE  Border = 1 CellPadding = 3 CellSpacing = 3 Height = 250
Width = 350>
    <CAPTION><B>Football Scores</B></CAPTION>

         <TR>
                <TD ROWSPAN = 6><BR></TD>
        </TR>

        <TR>
               <TD COLSPAN = 2>Home Team</TD>
               <TD COLSPAN = 2>Away Team<BR></TD>
        </TR>
        <TR>
                <TD BGCOLOR = White>Arsenal</TD>
                <TD BGCOLOR = White>2</TD>
                <TD BGCOLOR = White>Leeds</TD>
                <TD BGCOLOR = White>1</TD>
         </TR>

         <TR>
                 <TD BGCOLOR = White>Derby</TD>
                 <TD BGCOLOR = White>1</TD>
                 <TD BGCOLOR = White>Everton</TD>
                 <TD BGCOLOR = White>1</TD>
         </TR>

        <TR>
                <TD BGCOLOR = White>Liverpool</TD>
                <TD BGCOLOR = White>0</TD>
                <TD BGCOLOR = White>Sunderland</TD>
                <TD BGCOLOR = White>2</TD>
         </TR>

         <TR>
                 <TD BGCOLOR = White>Newcastle</TD>
                 <TD BGCOLOR = White>1</TD>
                 <TD BGCOLOR = White>Man Utd</TD>
                 <TD BGCOLOR = White>4</TD>
    </TR>

</TABLE>
```

Again, you can use the HTML Editor to add the Colspan and Rowspan tags for you:

- Click after the TR of a Table Row tag, and add a space
- Click on Format > Table attributes (If it's greyed out it means you haven't added a table yet)
- The Add Table attributes dialogue box appears, the one you've just met
- Click inside the box next to "Col/Row Span"
- The Col and Row options appear



☑ Col/Row Span

ColSpan = 2 Rowspan = 0

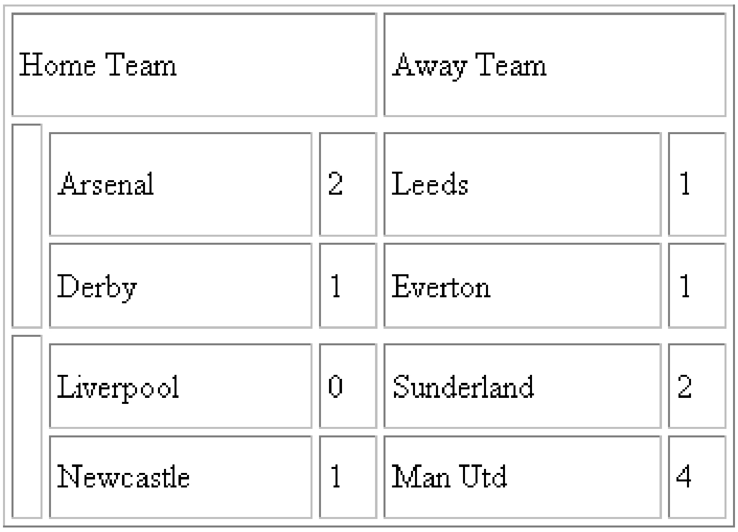
- More times than not, you'll add either one or the other. Here, we're setting the Colspan to 2. The Rowspan is left at zero, meaning we don't want to add the Rowspan attribute.
- Click OK when you're done. The attribute will be added to the tag.

There's no doubt about it though – setting up a complex table with cells spanning across other cells can be a tricky business when you're coding by hand. Practice will ensure that you understand the process of CELLSPAN and ROWSPAN. To that end, here's an exercise.

Using CELLSPAN and ROWSPAN, create a table that looks like this one:

Exercise

Football Scores



Home Team			Away Team	
	Arsenal	2	Leeds	1
	Derby	1	Everton	1
	Liverpool	0	Sunderland	2
	Newcastle	1	Man Utd	4

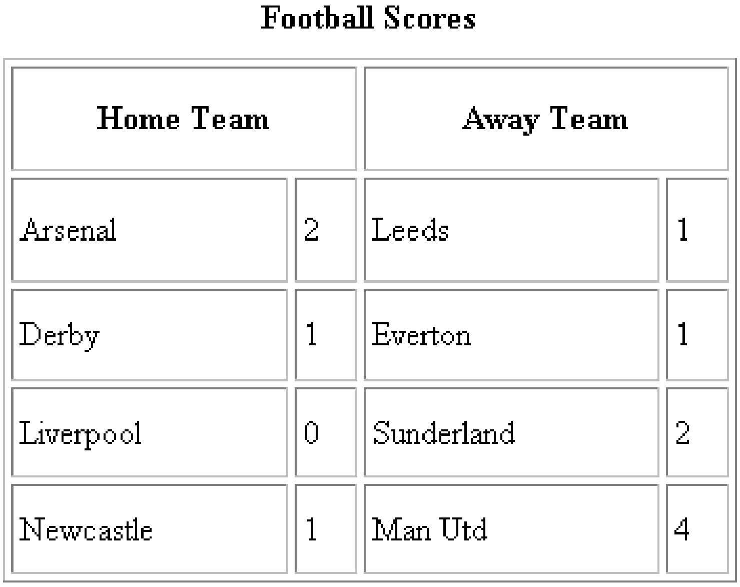
Aligning contents in a cell

We can align the contents in a cell, so they look more presentable. To align a cells contents, the ALIGN attribute is used in the TD part of the tag. To centre align our headings, the code would be this:

```
<TR  Height = 50 width = 100>
    <TD COLSPAN = 3 ALIGN = Center>Home Team</TD>
    <TD COLSPAN = 2 ALIGN = Center>Away Team</TD>
</TR>
```

Again, note the American spelling of centre – “ER” not “RE”. You can add a splash of bold to the text in the cell by highlighting the text and clicking on **Format > Bold**.

In a browser, our table would now look like this:



Football Scores

Home Team		Away Team	
Arsenal	2	Leeds	1
Derby	1	Everton	1
Liverpool	0	Sunderland	2
Newcastle	1	Man Utd	4

As you can see, the centring has made the table look a lot better. There are a lot more alignment options for cell contents than CENTER. The three basic horizontal alignment options are: LEFT, CENTER and RIGHT.

```
<TD ALIGN = Left>
<TD ALIGN = Center>
<TD ALIGN = Right>
```

If you want, you can use vertical alignment instead of horizontal. You use the VALIGN tag for vertical alignment. The positions for vertical alignment are: TOP, MIDDLE and BOTTOM.

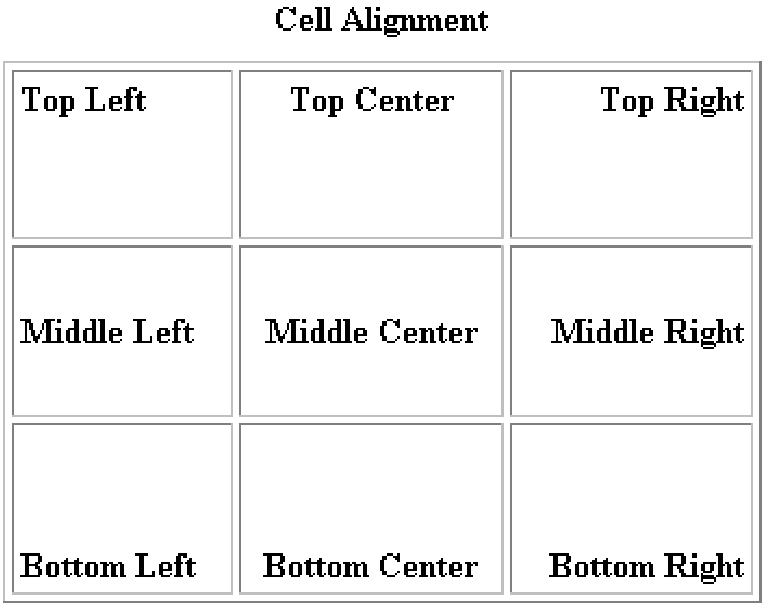
<TD **VALIGN = Top**>
<TD **VALIGN = Middle**>
<TD **VALIGN = Bottom**>

You can combine horizontal and vertical alignment to give nine positions in all:

<TD **VALIGN = Top Align = Left**>
<TD **VALIGN = Top Align = Center**>
<TD **VALIGN = Top Align = Right**>

<TD **VALIGN = Middle Align = Left**>
<TD **VALIGN = Middle Align = Center**>
<TD **VALIGN = Middle Align = Right**>
<TD **VALIGN = Bottom Align = Left**>
<TD **VALIGN = Bottom Align = Center**>
<TD **VALIGN = Bottom Align = Right**>

In a browser, here's what a table of all nine positions would do to text:

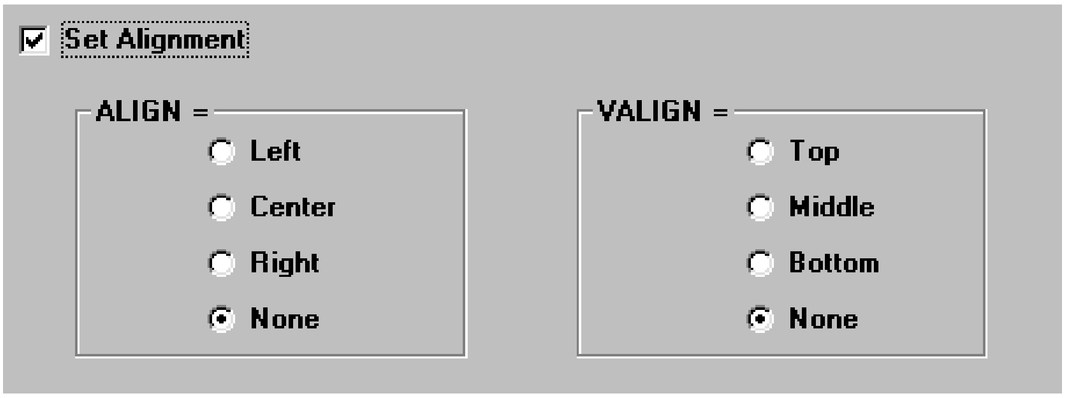


Cell Alignment

Top Left	Top Center	Top Right
Middle Left	Middle Center	Middle Right
Bottom Left	Bottom Center	Bottom Right

You can use your HTML Editor to Align the football scores in your table so that they are all in the centre of the cells. Here's how:

- Click after the “D” of a <TD> tag
- Type a space
- From the menu bar, click **Format > Table** Attributes
- The Add Table Attributes dialogue box appears once again
- This time, click the box next to “Set Alignment”. The alignment options appear:



☑ Set Alignment

ALIGN =
○ Left
○ Center
○ Right
● None

VALIGN =
○ Top
○ Middle
○ Bottom
● None

- The defaults for the ALIGN and VALIGN options are set to None
- Select an alignment option from the choices and click OK
- The code is added to your TD tag:

<TD ALIGN = Center VALIGN = Middle>2</TD>

Changing the background colour of Table and Cells

Individual cells can have their background colour changed. The background colour of the entire table can be changed, too. To change the colour of the entire table, just add the BGCOLOR tag to the TABLE tag. Like this:

<TABLE BGCOLOR = Lime>

Because the HTML Editor adds a default colour of white to each individual cell, however, these would have to be deleted from the TD tag to see the full effect.

So if the original code was this:

```
<TR>
        <TD BGCOLOR = White>Arsenal</TD>
        <TD BGCOLOR = White>2</TD>
        <TD BGCOLOR = White>Leeds</TD>
        <TD BGCOLOR = White>1</TD>
</TR>
```

By deleting the BGCOLOR code, you'd have this:

```
<TR>
        <TD>Arsenal</TD>
        <TD>2</TD>
        <TD>Leeds</TD>
        <TD>1</TD>
</TR>
```

With the BGCOLOR tag gone from all your cells, our Football Scores would look like this in a browser – actually, we won't show you it in a browser. Add the BGCOLOR attribute to your own <TABLE> tag, and load it up into your browser to see what it looks like.

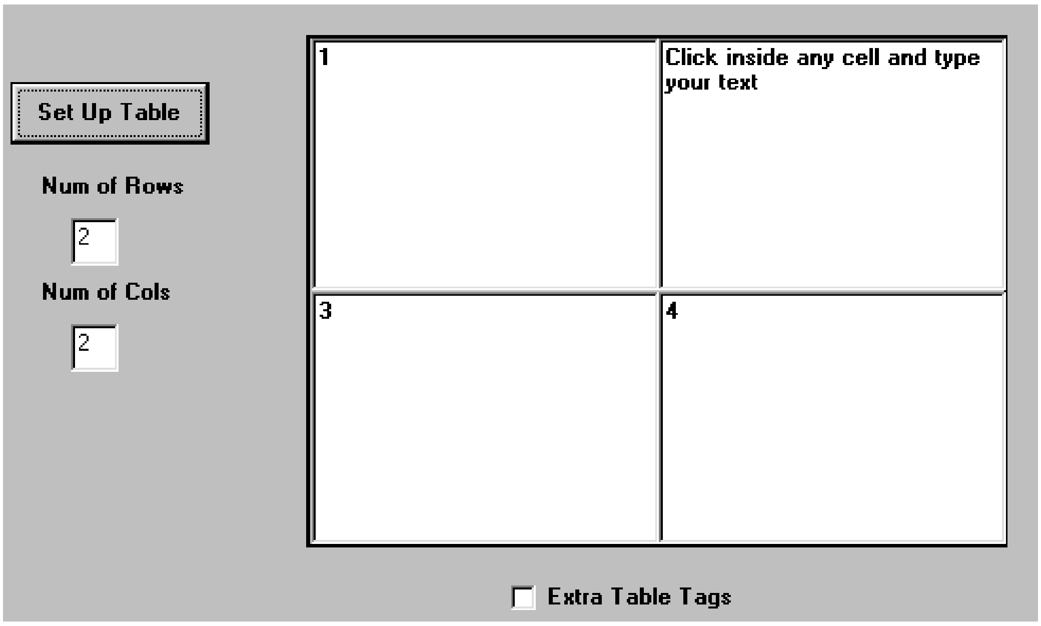
Colouring individual cells

You can add a splash of colour to individual cells. The BGCOLOR attribute is just added to the TD tag whose colour you want to change. Like this:

<TD BGCOLOR = "Yellow">Arsenal</TD>

Much easier is to use the HTML editor to add the colour to cells for you.

- Create a new table with two rows and two columns
- The Table dialogue box will look like this:



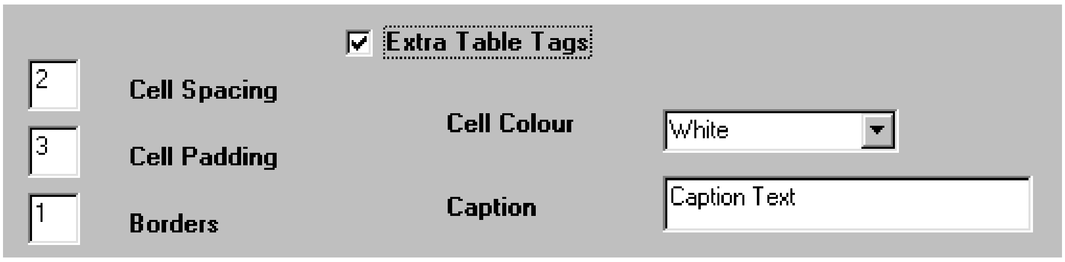
Num of Rows 2

Num of Cols 2

1	Click inside any cell and type your text
3	4

□ Extra Table Tags

- Click inside the box next to “Extra Table Tags”
- Extra Table tag options will appear:


☑ Extra Table Tags

2 Cell Spacing

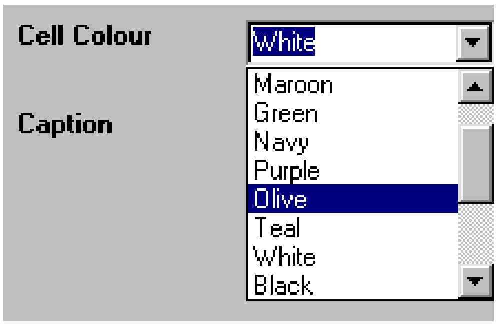
3 Cell Padding

1 Borders

Cell Colour White

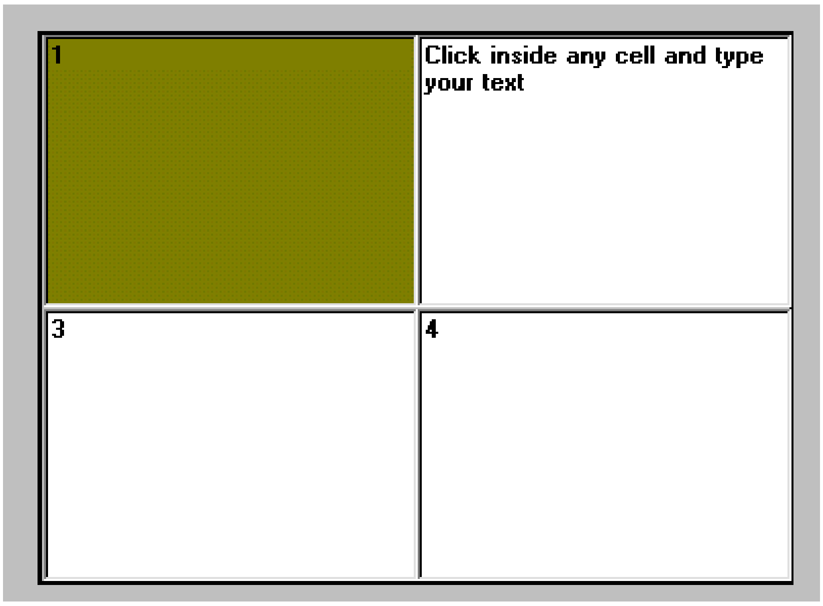
Caption Caption Text

- You may, at this point, want to delete any text or numbers from the text boxes and your own (or even leave them blank)
- Click the black down-pointing arrow of the Cell Colour box to see a list of colour options. Select one of them:

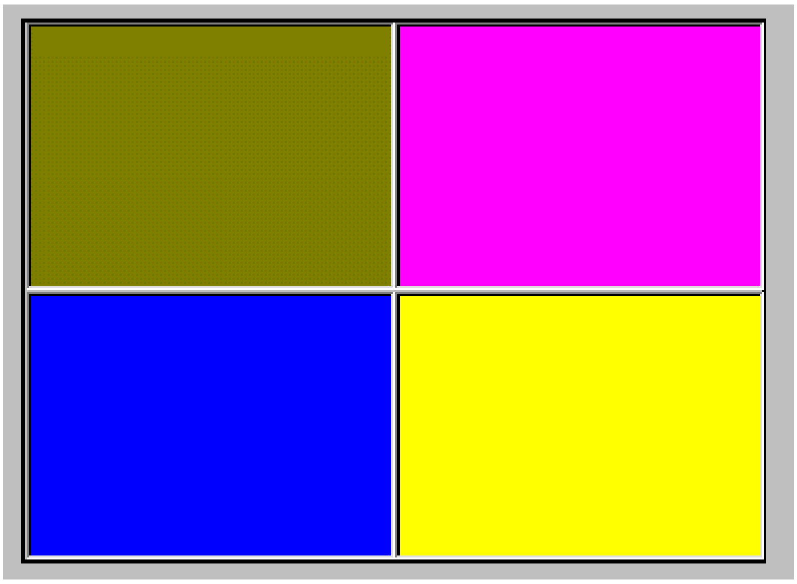

Cell Colour White

Caption

- Next, click inside one of your text boxes
- The background colour will change



- Select another colour from the drop-down box, and click another text box
- Continue until all four text boxes are coloured:



- Finally, click the OK button
- Your code will look like this (some attributes have been deleted, so that you can see the new code more easily):

```
<TABLE  Border = 1 Height = 200 Width = 200>

    <TR>
        <TD BGCOLOR = Olive><BR></TD>
        <TD BGCOLOR = Magenta><BR></TD>
    </TR>
    <TR>
        <TD BGCOLOR = Blue><BR></TD>
        <TD BGCOLOR = Yellow><BR></TD>
    </TR>

</TABLE>
```

- Add a new height and width for your table
- Save your work, and view the results in a browser. OK, it might look awful, but it does demonstrate how individual cells can have their colours changed.

If you want to use a colour other than those in the drop down box, highlight the colour you want to change. From the menu bar, click on **Format > Body Colour**. From the colour dialogue box that pops up, click a colour that takes your fancy.

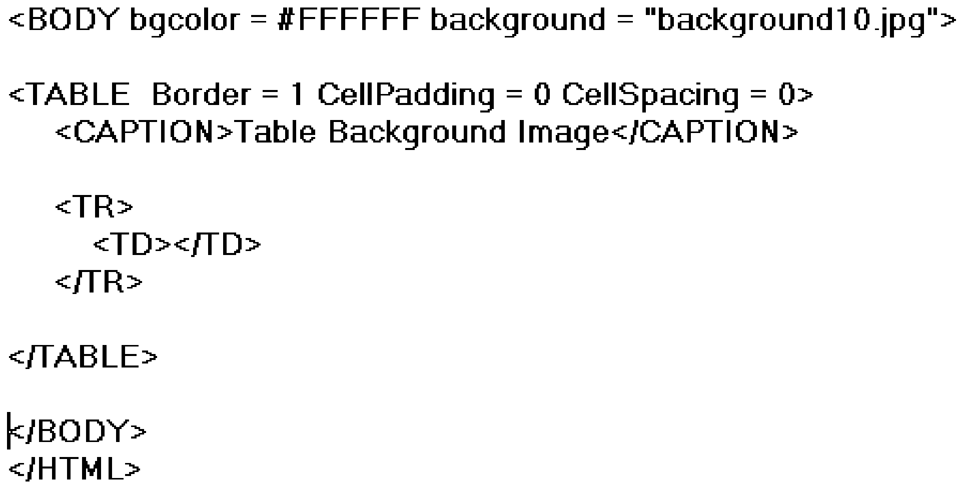
Using images in tables

Images can be used in tables, either as the background image of the table, or in individual cells.

Start your browser, and load up the web page called **tableimage.html** in the files you downloaded at the start of the book. The web page shows an example of a table with patterned background. Let’s see how to create it. It’s quite easy.

- Start a new web page in your HTML Editor
- Set the background colour or image to anything you like
- Click OK
- With the basic HTML code displayed, click in between the two BODY tags
- From the menu bar, click on **ADD > Add a Table**
- The Add a Table dialogue box appears

- Set the Number of Rows to one, and the number of columns to one
- Click the “Set Up Table” button
- You can either just click OK at this stage, or add some Extra Table Tags, a Border and a Caption, perhaps
- Your HTML Editor will now look like this:



```
<BODY bgcolor = #FFFFFF background = "background10.jpg">

<TABLE  Border = 1 CellPadding = 0 CellSpacing = 0>
   <CAPTION>Table Background Image</CAPTION>

   <TR>
     <TD></TD>
   </TR>

</TABLE>

</BODY>
</HTML>
```

You can delete the CellPadding and CellSpacing tags, if you want. Delete the Border tag as well, and you’d have this:

<TABLE>

Type a space after the “E” of table, and type **BACKGROUND =**

<TABLE BACKGROUND = >

- From the menu bar, click on **Insert > Image**
- The Insert Image dialogue box appears
- Click on **Insert a Background Image**
- The Open Image dialogue box appears
- Locate the image you want to use as a background (in our case it was background12.jpg from the downloaded files, in the Backgrounds folder)
- Click Open, then click OK when you’re returned to the **Insert a Background Image** dialogue box. Your code will look like this:

<TABLE BACKGROUND = "background12.jpg">

- Highlight the entire table
- Click **Format > Centre**
- Save your work and load it into a browser to see the final results

Images in Cells

You can have an image in a Table TD cell, rather than text. To insert an image into a cell with your HTML Editor, do this:

- Click in between a Pair of <TD> tags
- From the menu bar, click on **Insert > Image**
- From the Insert an Image dialogue box, click on **Search for an Image**
- Locate the image you want to use and click OK
- Align the image with the ALIGN and/or VALIGN attributes
- Save your work and view the results

You can turn the image into a Link, if you want. Or just add a link instead of an image or text.

Nested Tables

Tables can be quite effective when you nest one inside the other. Load up the web page called **nestedtables.html** in the Tables folder of your downloads to see an example of a nested table (though Firefox users may not see the full effect). To create this web page for yourself, do this:

- Create a new page in your HTML Editor
- Repeat the steps from the **Using images in tables** section a few pages back
- You might have something like this:

```
<CENTER>
<TABLE  Height = 400 Width = 500  background = background12.jpg">

        <TR>
                <TD><BR></TD>
        </TR>

</TABLE>
</CENTER>
```

To nest a table, you need to insert a new TABLE inside one of the TD Tags. In other words, instead of having text or an image or a link in the Table Data tag, you're putting a new table in there.

- So click in between the two <TD> tags of your table (where the <BR> tag is above, which you can delete)
- From the menu bar, click **Add > Add a Table**
- From the table dialogue box, set the Number of Rows to 1 and the Number of Columns to 1
- Click OK, and your code will look like this:

```
<CENTER>
<TABLE  Height = 400 Width = 500  background = background12.jpg">

        <TR>
              <TD>

                        <TABLE BGCOLOUR = White>
                              <TR>
                                    <TD></TD>
                              </TR>
                        </TABLE>

              </TD>
        </TR>

</TABLE>
</CENTER>
```

Notice where the nested table has gone – in between two <TD> tags. A white background colour has been added to the new TABLE tag, and our new nested table also has one row and one cell in the row.

- Add some text in between the new TD tags:

```
        <TABLE BGCOLOUR = White>
                <TR>
                        <TD><B>This table with the white
background has been nested inside the one with the patterned
background</B></TD>
                </TR>
        </TABLE>
```

Save your work, and then view the results in a browser. You should have something similar to the one from the downloaded files.

We’ve covered quite a lot of ground in this Tables section, and there’s a lot to remember. But there’s no doubt about it – tables are an essential tool for budding and professional web designers; they are worth getting the hang of.

And now on to a Review.

Review Number 5

Web Page One

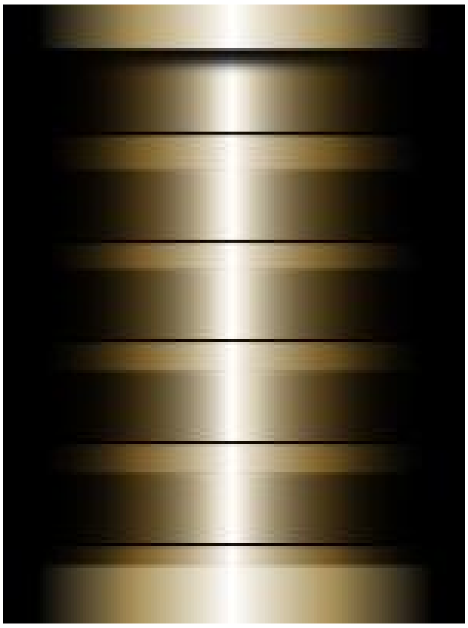
Use a table to present the following data:

World Population Figures – 1750 to 1999

	1750	**1800**	**1850**	**1900**	**1950**	**1999**
Asia	502	635	809	947	1402	3634
Africa	106	107	111	133	224	767
Europe	163	203	276	408	547	729
Southern America	16	24	38	74	166	511
Northern America	2	7	26	82	172	307
Oceania	2	2	2	6	13	30
World	791	978	1262	1650	2524	5978

Web Page Two

In the Tables folder of the files you downloaded there are 5 spool gif images (spool1.gif, spool2.gif, etc). The 5 pieces are part of a navigation bar. Use a table to assemble the pieces so that you end up with this navigation bar below:



Each separate part of the navigation bar needs to go inside its own TD cell. At the moment, there is no text on each of the bars. You could add some text, if you have an image editing package, and then turn each part into a clickable link. But this is not necessary for a Pass. A correctly assembled navigation bar is, however. (Images are often sliced like this, by the way, because it speeds up download times.)

A suitable background image or colour would complete the web page.

Forms

HTML forms are a way of gathering data from visitors to your page. Forms typically have text boxes for data input, radio buttons, drop down boxes, and buttons so that users can Submit the form. A reset button is also handy, just in case mistakes are made filling out the form. The best way to understand what a form does is to see one in action. There is a folder amongst the files you downloaded called **Forms**. Inside this folder is a web page called form1.html. Open this web page and have a play about with the form. Fill out the form and click the Submit button (you won't be sending it anywhere). Test the Reset button, too.

You will shortly be using your HTML Editor to create the form1.html web page. But let's start with a few explanations.

The Form Tag

You don't have to tell your browser about any form elements on your web page. If you just want a simple form element like a text box, you can insert the text box tag by itself. But if you want to do something with the information on your form, like send it somewhere or to someone, you have to "tell" your browser about the form elements on your page. You do this with the Form tag:

```
<FORM>
</FORM>
```

Like most HTML tags, the FORM tag comes as a pair, the forward slash preceding the second FORM tag to indicate that the form tag ends. Any form elements you need then go between these two FORM tags.

A NAME attribute is a useful addition to the FORM tag. When the form has a name, you can then refer to it in scripts. You'll see how to process a form with scripting in a later section. For now, just be aware of how to use the name:

```
<FORM NAME = “frmSurvey”>

</FORM>
```

To give your form a name, type a space after FORM then type the word NAME, followed by an equals sign. Finally, add a name for your form. You can call it anything you like. Here, we've called the form "frmSurvey".

If you want your form to go somewhere or to someone, two other attributes are needed in the FORM tag: METHOD and ACTION. Like this:

<FORM NAME = "frmSurvey" **METHOD = "post" Action =mailto:me@me.com>**

</FORM>

METHOD is way to send the data. There are two options, Post and Get. **Post** sends the data in single lines of text; **Get** squashes all the data in a single line and adds it the URL in the Action part. If the URL was an internet address, you'd see all the data you're sending in the address bar of your browser. This sort of thing:

ProcessSurvey.html?text1=John&text2=Smith

The first name John was typed into the text box called "text1" and the surname Smith went into the text box called "text2". That is a direct result of using the Get method to send data. The Post method doesn't add all that code to the address bar of your browser.

You should use Post to send your data, rather than Get.

ACTION is used to specify the address where you want to send the data. Here, we're using an Email link to send the data straight to an email address:

ACTION = mailto:me@me.com

But the form can be processed by a CGI script or an ASP script. In which case you'd specify the address of the script in question:

ACTION = "ProcessSurvey.asp"

To ensure that data in your forms is not sent anywhere, you can just add a pair of double quotes to the ACTION attribute:

ACTION = “”

Form Elements

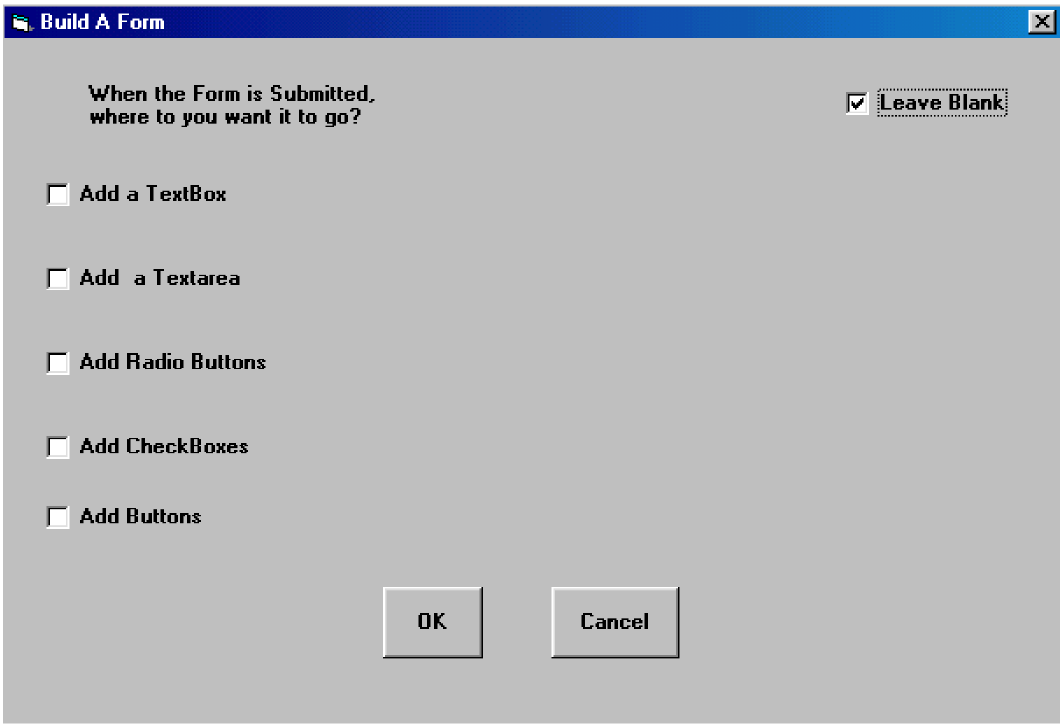
There are quite a few different form elements you can add to a form:

Text Boxes
Text Areas
Option Buttons
Check Boxes
Drop down List/Fixed Lists
Password Boxes
Command Buttons
Submit Buttons
Reset Buttons
Image Command Buttons
Hidden Form Values

We’ll explain them in more detail as we go along. But let’s do some practical work. We’ll start designing the Survey Form using the HTML Editor.

So, open your editor and create a new HTML page, any colour background you like. Click in between the two BODY tags and then do this:

- From the menu bar, click on **Add**
- From the drop down list, click **Add a Form**
- The Form dialogue box appears



Build A Form

When the Form is Submitted,
where to you want it to go?

☑ Leave Blank

□ Add a TextBox

□ Add a Textarea

□ Add Radio Buttons

□ Add CheckBoxes

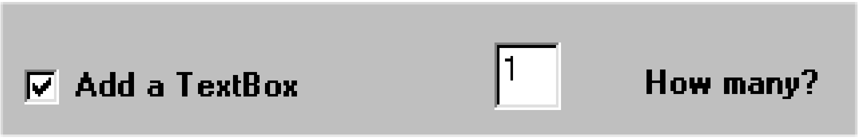
□ Add Buttons

The Build a Form dialogue box lets you add 5 basic form elements, and allows you to add a URL for the ACTION attribute of the FORM tag (The editor will automatically add the METHOD attribute).

If you want to add an ACTION attribute, un-tick the box “Leave it Blank”, and then type in your URL (this can be an email address or a website address). Let’s start by adding a couple of Text Box.

Text Boxes

- Tick the box next to **Add a Textbox**
- The following appears:



☑ Add a TextBox 1 How many?

The dialogue box is obviously asking how many text boxes do you want on your form. Our form had two of them, so enter a 2 in the text box.

If you were to click OK now, this is the code you’d get for a single text box:

<INPUT TYPE = "Text" Size = 20 Value = "" Name = text1>

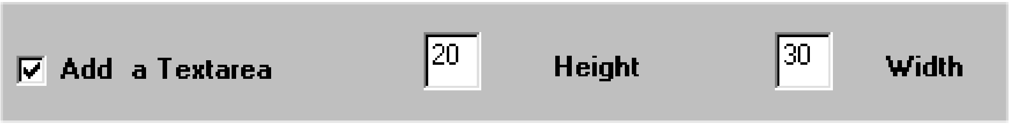
For most form elements, the word INPUT is used to set up the element. Next, you type a space followed by the word TYPE. This tells the browser what type of form elements to draw on your page. If you want a text box, the TYPE to use is “Text”. (With or without the quotes.)

Next, you add the attributes you want. The HTML Editor will add a size attribute. This is set to 20 pixels long. But you can change this value, if you want a longer text box. (Width and Height don’t work for form elements.) The Editor will also add a name for your text box. The default names are “text1”, “text2”, “text3”, etc. But you can change the names to anything you like. The name of the form elements can then be used in scripts to manipulate data from the text box.

Notice that there is no end tag for INPUT. You can add some default text to your text box by typing the text you want after the VALUE attribute.

Text Areas

With your text boxes set to 2, click inside the box next to “Text Area”. You’ll see this:



☑ Add a Textarea 20 Height 30 Width

The Height and Width values are set to 20 and 30 in the editor, but this is a bit too big for an address box. (Although Height and Width are used in the Editor, these are not added to the code.) If you were to click the OK button with just a text area added, the code would be this:

```
<TEXTAREA Rows =7 Cols = 30 NAME = "TextArea1">
            Your Address Here
</TEXTAREA>
```

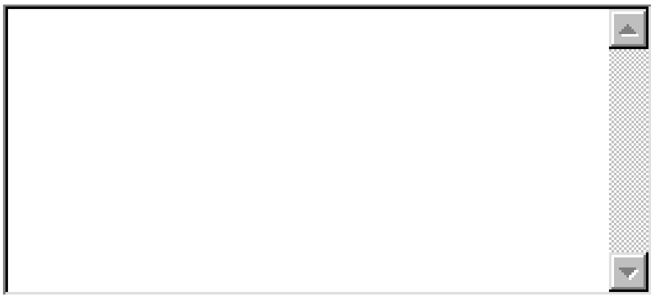
TextArea (all one word) doesn’t use the INPUT tag. And you need an end TextArea tag as well **</TEXTAREA>**. After typing a space, you specify how big your Rows and Columns are going to be. These are the Height and Width from the Dialogue Box. The Height equates to Rows, and Width to Cols (not Columns). 7 Rows and 30 Cols is a nice size to use for an address text area.

Again, the Editor gives your element a default Name, which you can change.

After typing a right angle bracket (>) you can then type in some text to be used as a default. You don’t have to add any text, if you don’t want to. In which case, it would be just this:

```
<TEXTAREA Rows =7 Cols = 30 NAME = "TextArea1">
</TEXTAREA>
```

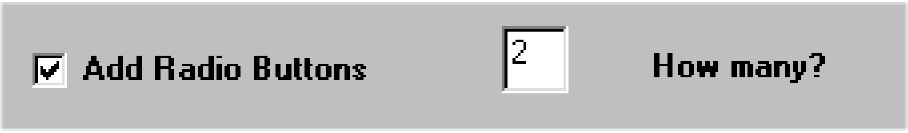
The text area above would then look like this:



Option Buttons

Option buttons are sometimes called Radio Buttons, and they force the user into choosing only one item in a list. In the form1.html sample form, Option Buttons were used to limit your choice of Operating Systems. A popular use of Options Buttons is the Male/Female option.

With the Form Dialogue box still open, tick the box next to “Radio Buttons”. You’ll see this:



☑ Add Radio Buttons 2 How many?

Again, it’s a simple matter of specifying how many radio buttons you want. We added 5 of them for the Operating Systems, so delete the 2 and type in a 5.

If you were to just have Radio Buttons selected, and you clicked OK, the code would be this for the minimum 2 radio buttons needed:

```
<INPUT TYPE = Radio Name = R1 Value = "Radio1" Checked>Windows 95

<INPUT TYPE = Radio Name = R1 Value = "Radio2">Windows 98
```

After typing the INPUT tag, the word TYPE comes next. For Option Buttons, the type is “Radio”. The NAME is definitely needed here, and note that the NAME for both is “R1”. You use the same name for each group of option buttons you are adding to your form. So if you wanted Male/Female option buttons, the code might be this:

```
<INPUT TYPE = Radio Name = R2 Value = "Radio1" >Male

<INPUT TYPE = Radio Name = R2 Value = "Radio2">Female
```

This time, each radio button has the name “R2”. The reason you keep the same name for each group of option buttons is simply to distinguish one group of option buttons from another.

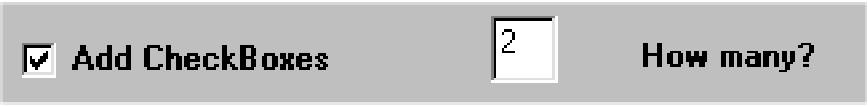
The VALUE attribute is quite useful. The Editor puts in a default value of Radio1, Radio2, Radio3, etc. But you should set the value to the same text as the user sees. So the Values in the codes above should be “Windows 98” and “Windows 95”, and not “Radio1” and “Radio2”. When the user submits the form to you using the Submit button, these VALUES are going to be returned. If you’ve just got Radio1 and Radio2, you won’t know (or won’t remember, more likely) which option the user has selected. Manipulating values with scripts is also a lot easier if the Value is the same as the text the user sees.

If you want to have a default option button selected, use the word “Checked”.

To complete the tag, type the right angle bracket (>). After the right bracket, you then type the text that the user will see. In our case, the name of an Operating System: Windows 95, Windows 98, Windows ME, Windows XP, Other.

Check Boxes

Check boxes are used to give your users the opportunity to select more than one option from a range of options. With the Radio Buttons, you could only select one item; with check boxes, you can select them all. So on your Form dialogue box, tick the box next to “Add Checkboxes”. You’ll see this:



☑ Add CheckBoxes 2 How many?

Again, you are being asked how many of the element do you want. The default is 2. We need 6 check boxes. So delete the 2 and enter a 6. If you were to click the OK button with only the Checkbox option ticked, the code would be this for the default 2 boxes:

```
<INPUT TYPE = Checkbox Name = CheckOne Value = "Check1">Printer
<INPUT TYPE = Checkbox Name = CheckOne Value = "Check2">DVD
```

Check boxes and Radio buttons work in a similar way, and the HTML code is similar, too. The same points made about Radio buttons apply to check boxes. Note the TYPE used, though: Checkbox.

Submit and Reset buttons

If you want your form to be sent somewhere, a Submit button is needed (though you can write code for your own submit button – the browser’s own Submit button is not required to send data somewhere.) When the Submit button is clicked, the browser will check the ACTION attribute of the FORM tag to see where you want the data sent. It then checks the METHOD attribute to see what method you want to use, Post or Get. The browser will then try to send the form’s data for you.

The code for a Submit button is this:

```
<INPUT TYPE = Submit Value = "Submit">
```

This time, the TYPE is set to “Submit”. The VALUE attribute is the text that will appear on the button itself. The width of the button is determined by the width of the text for the VALUE attribute. If you wanted a really wide button, you can use this old trick:

Value = "   Submit   ">

Here, spaces are added to the left and right of the text. The browser will see the spaces as text and adapt the width accordingly.

Reset

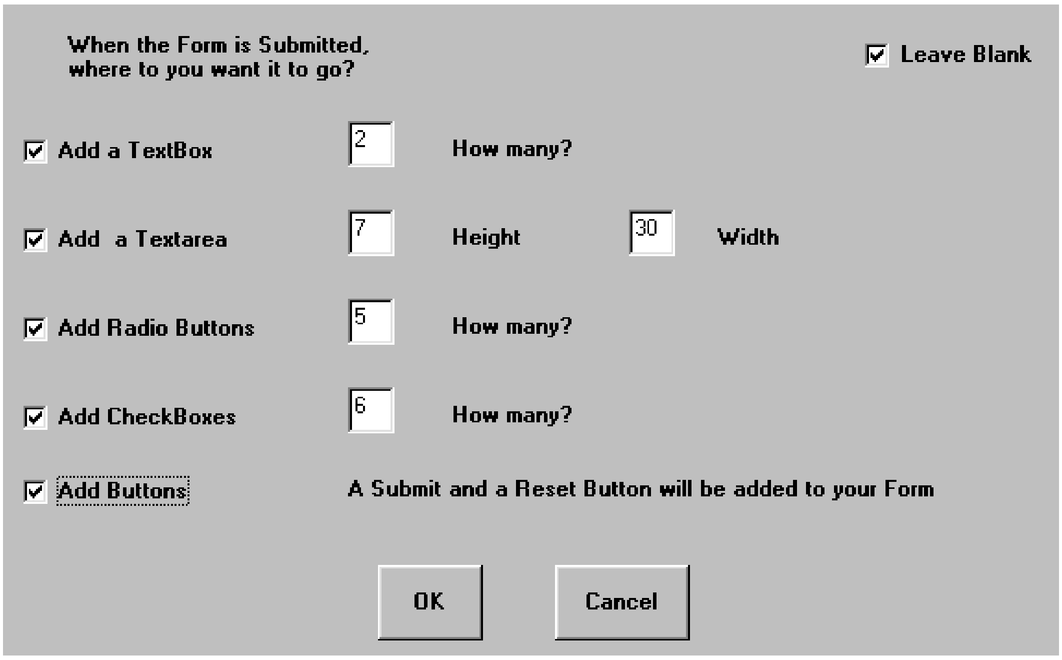
The Reset button returns the form to the state it was originally in when it was loaded. Any default values you added will be retained. The code for a rest button is this:

<INPUT TYPE = Reset Value = "Clear">

Note the TYPE is now “Reset”. The value attribute works in the same way that the Submit button does – it’s the text that will appear on the button. So you could have this, and your Reset button will still work:

<INPUT TYPE = Reset Value = "Oh no – What a mess!">

By now, you should have all 5 boxes on the Form dialogue box ticked. Your dialogue box will then look like this:



When the Form is Submitted,
where to you want it to go?

☑ Leave Blank

☑ Add a TextBox 2 How many?

☑ Add a Textarea 7 Height 30 Width

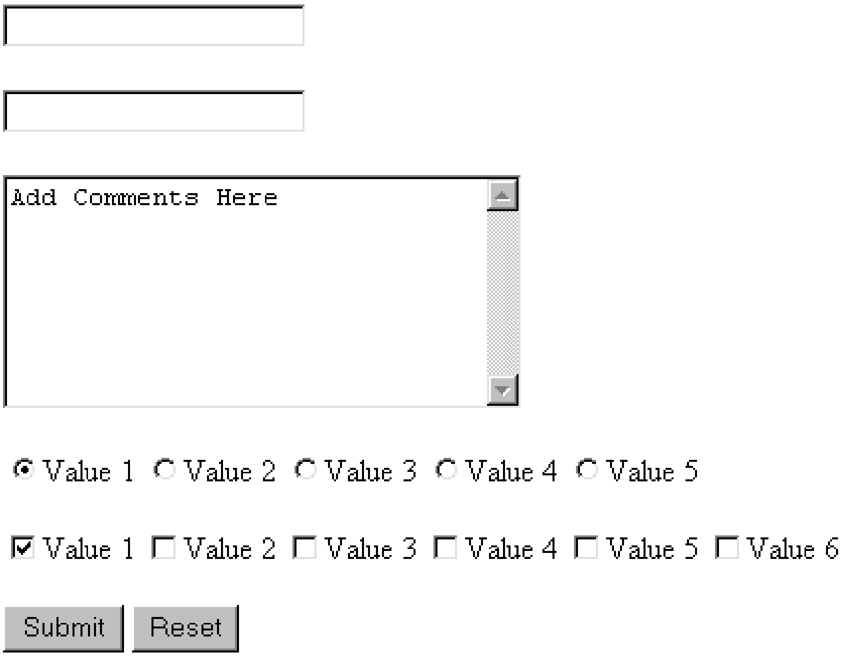
☑ Add Radio Buttons 5 How many?

☑ Add CheckBoxes 6 How many?

☑ Add Buttons A Submit and a Reset Button will be added to your Form

When your dialogue box looks like the one above, click OK. The code will be added to your document.

If you save your work right now, and then load the page into a browser by clicking **View > View Web Page** or by clicking the glasses on the toolbar, you browser will look something like this one:



Add Comments Here

● Value 1 ○ Value 2 ○ Value 3 ○ Value 4 ○ Value 5

☑ Value 1 □ Value 2 □ Value 3 □ Value 4 □ Value 5 □ Value 6

The form at the moment is not much good to anybody. The default values need to be changed and some captions added, so that people will know what the heck it is they're supposed to do.

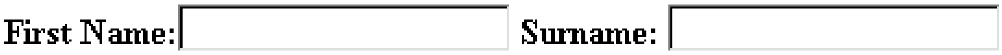
But you should have a form on your page, and see how it easy it is to add one with your Editor. We'll tidy it up a bit. It's not difficult.

You can add some captions to the text boxes, in order to identify them. But there are no special tags needed. You just type out what you want to use as a caption before the code for the element. Like this:

First Name: <INPUT TYPE = "Text" Size = 20 Value = "" Name = text1>

Surname: <INPUT TYPE = "Text" Size = 20 Value = "" Name = text2>

To get these two text boxes on the same line, just delete the paragraph tag <P> inserted by default between the two. The text boxes on your form will then look like this:



For the Text Area, we can change that default text. It will be this in your code (though yours will be on one line):

```
<TEXTAREA Rows = 7 Cols = 30 NAME = "TextArea1">
        Add Comments Here
</TEXTAREA>
```

Simply delete the text “Add Comments Here” and type some text in its place, something like “Type your Address here”.

The code for the Radio and Option buttons needs some work, though: Yours will be like this for the Radio buttons:

```
<INPUT TYPE =Radio Name =RadioOne Value ="Radio1" Checked>Value 1
<INPUT TYPE = Radio Name = RadioOne Value = "Radio2">Value 2
<INPUT TYPE = Radio Name = RadioOne Value = "Radio3">Value 3
<INPUT TYPE = Radio Name = RadioOne Value = "Radio4">Value 4
<INPUT TYPE = Radio Name = RadioOne Value = "Radio5">Value 5
```

You can add some text to serve as a caption in the same way you did for the text boxes – just type something before the first radio INPUT tag. You then need to change the text that people will see, those Value 1, Value 2, etc, parts. Change the Value attributes (Value = Radio1, Value = Radio2, Value = Radio3, etc) to match your new text. So your new code would be this:

```
Please select your Operating system:<P>
<INPUT TYPE =Radio Name =RadioOne Value =" Windows 95"
Checked>Windows 95

<INPUT TYPE = Radio Name = RadioOne Value = " Windows 98"> Windows
98

<INPUT TYPE = Radio Name = RadioOne Value = " Windows ME ">
Windows ME

<INPUT TYPE = Radio Name = RadioOne Value = " Windows XP ">
Windows XP

<INPUT TYPE = Radio Name = RadioOne Value = " Other "> Other
```

For the Check boxes, make changes in the same manner as the changes you’ve just made to the Radio Buttons.

The Submit and Reset buttons can have their text changed, if you want.

You can make any text (like First Name or Surname) bold, or change the FONT style and colour, or centre your form in the normal manner. When you're finished, your form in a browser should look something like the one you download at the start of this book. (Except your Submit button won't cause a pop up message box to appear.)

Aligning Form Elements

The centred form looks a little unprofessional as it is. You can align the elements on your form better by putting them into a table.

The best way to do this is by inserting a form with no form elements on it (Click **Add > Add a Form**, then just click OK when the dialogue box pops up.) This code will be appear in your editor:

```
<FORM Method = Post Name = frmOne Action = "">

</FORM>
```

Click in between the two Form tags, then click **Add > Add a Table**. Set how many rows and columns you want in your table, then click OK. In this example, we'll set the Rows to 1 and the columns to 4.

Next, add the Form elements individually with your Editor. We'll see how to do this now.

Assume the code for the Form and Table is this (with only the first row of tags showing):

```
<FORM Method = Post Name = frmOne Action = "">
<TABLE>
    <TR>
        <TD>1</TD>
        <TD>2</TD>
        <TD>3</TD>
        <TD>4</TD>
    </TR>
</TABLE>
</FORM>
```

- To insert a Text box individually, delete the text from the TD tag. In the code above, delete the number 2 (The text First Name is going where the 1 is.)
- From the menu bar, click **Add**
- From the drop down menu, click **Add a Form Element**
- From the sub menu, click **Text Box**
- A single Text Box form element tag appears in your code
- The result will be this:

```
<TD>
  <INPUT TYPE = "Text" SIZE = 20 NAME = "" VALUE = "">
</TD>
```

Add another text box where the number 4 is in our TD tags. Replace the number 1 with First Name and the number 3 with Surname. Your code for the first row will now be this:

```
<FORM Method = Post Name = frmOne Action = "">
<TABLE>
    <TR>
         <TD>First Name</TD>
         <TD>
<INPUT TYPE = "Text" SIZE = 20 NAME = "" VALUE = "">
         </TD>
         <TD> Surname </TD>
         <TD>
<INPUT TYPE = "Text" SIZE = 20 NAME = "" VALUE = "">
          </TD>
    </TR>
</TABLE>
</FORM>
```

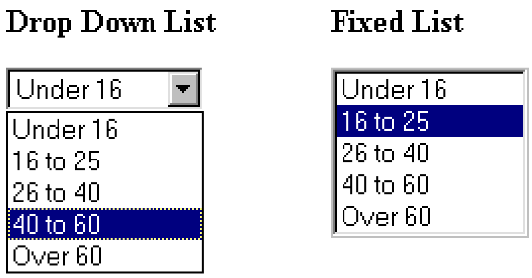
All the form elements you’ve learnt so far can be added to a table TD tag using the same method: Click **Add > Add a Form Element**, then click on a form element from the sub menu.

You will have noticed a few form elements in the sub menu that haven’t been discussed yet. We’ll go through them now.

List Boxes

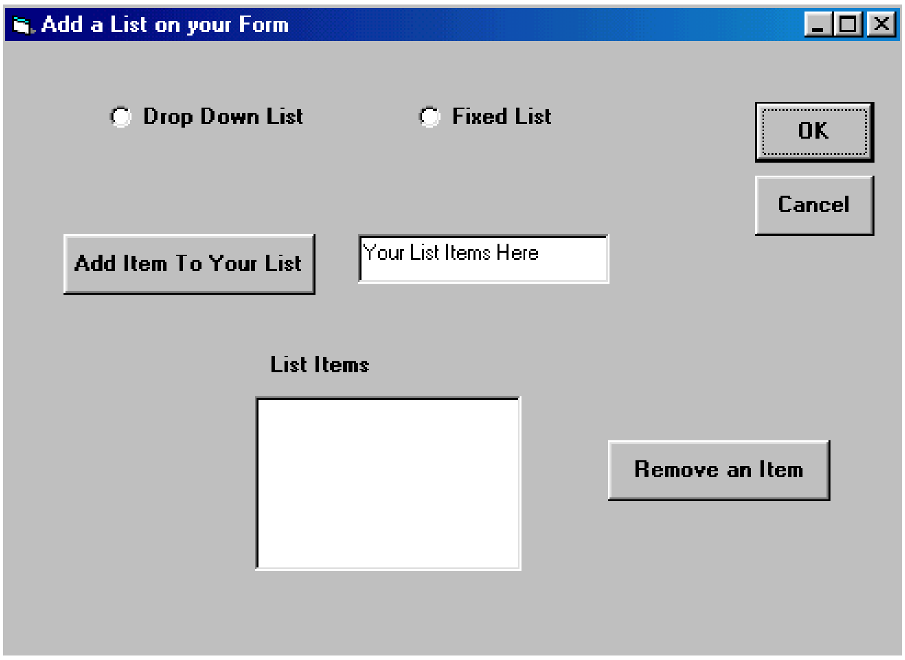
You can have a list box of items on your forms, either in a drop down format or as a fixed list. Here's what the two look like on a web page:

Drop down and fixed Lists:



To add a List to your form or web page using your Editor, do this:

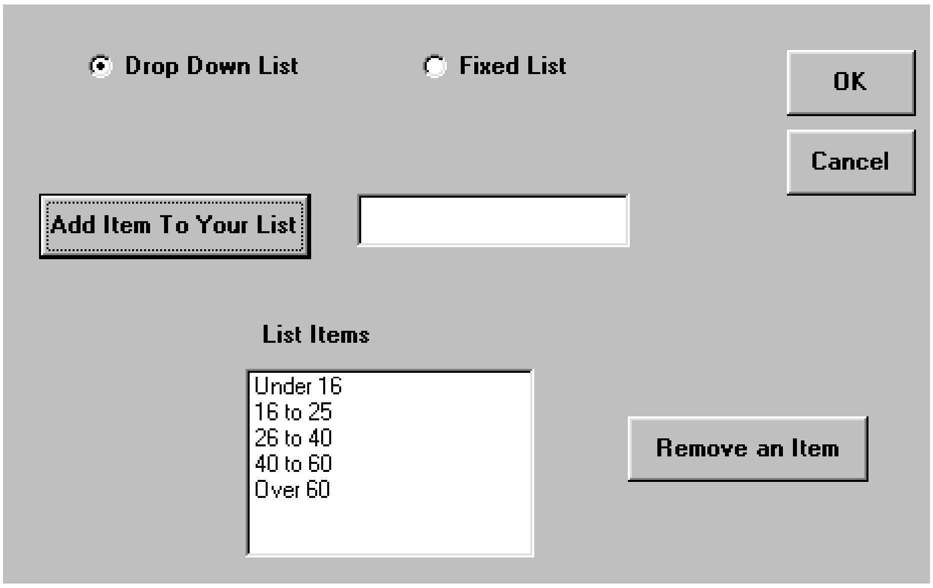
- From the menu bar, click on **Add**
- From the drop down list, click on **Add a Form Element**
- From the sub menu, click **Dropdown/Fixed List**
- The following dialogue box appears:



- Click either of the option buttons at the top: Drop down list or Fixed list

- Delete the default text “You List Items Here” from the text box
- Type in an item for your list
- Then click the button “Add Item To Your List
- The item appears in the box below. The text in text box will disappear, ready for another entry
- If you want to delete an item from your list, click on the item you want to remove, then click the button “Remove an Item”
- When you’re done, click OK

The image below shows what your dialogue box might look like before you click OK:



● Drop Down List ○ Fixed List

List Items

Under 16
16 to 25
26 to 40
40 to 60
Over 60

When you click OK, the code will look like this:

```
<SELECT NAME = List1 STYLE = "Width: 100">
        <OPTION Value = Under 16>Under 16</OPTION>
        <OPTION Value = 16 to 25>16 to 25</OPTION>
        <OPTION Value = 26 to 40>26 to 40</OPTION>
        <OPTION Value = 40 to 60>40 to 60</OPTION>
        <OPTION Value = Over 60>Over 60</OPTION>
</SELECT>
```

List boxes are called Select boxes in HTML and use the <SELECT> tag to set them up. The NAME and STYLE are not needed, and you could just have this:

```
<SELECT>
        <OPTION Value = Under 16>Under 16</OPTION>
        <OPTION Value = 16 to 25>16 to 25</OPTION>
</SELECT>
```

Change the NAME and Width values to anything you like.

Each item in your list needs an OPTION tag.

<OPTION Value = Under 16>Under 16</OPTION>

You don't need the closing OPTION tag, if you don't want it, but it's here for neatness sake. The text you want to appear in the list, the text that people see and click on, goes after the first right pointy bracket (>). The Value is not strictly needed, either. When the form is submitted, the option the user selected will be returned to you. If the Value attribute is missing, the text itself will be returned.

If you want one of the items in your list selected by default, just choose the item and add selected as an attribute. Like this:

<OPTION Value = "Under 16" **SELECTED**>Under 16

The only difference between the drop down list and the Fixed list is one attribute in the <SELECT> tag – SIZE

<SELECT SIZE = 5>

The SIZE is the number of item in your list. If you add this attribute, you'll get a Fixed list instead of a drop down list, which is the default.

If you want your users to be able to choose more than one item from the list, the attribute to add to the SELECT tag is MULTIPLE

<SELECT SIZE = 5 **MULTIPLE>**

Password Boxes

You can add a password box to you form. This is identical to a text box in its appearance. The only difference is that when you type in the box the characters are replaced by asterisks (*). It's important to remember that although password box hides the text from prying eyes, the data is not encrypted in any way. If you use METHOD = Get instead of Post, the password would show up in the address line of the browser.

The code for a password box is this:

<INPUT TYPE = Password NAME = “” SIZE = 20 MAXLENGTH = 8>

The MAXLEGTH attribute is the maximum number of characters that can be entered into the password text box.

To add a Password text box to your Form with the Editor, do this:

- From the menu bar, click on **Add**
- From the drop down menu, click **Add a Form Element**
- From the sub menu, click **Password Box**
- The password box code is inserted onto your page

The NAME and MAXLENGTH attributes will be blank. Insert your own values for these two attributes, or delete them if you prefer. Only the TYPE attribute is necessary.

Command Button

A command button looks just like the Submit and Reset buttons that you’ve already added to a form. The difference is that the Submit and Reset buttons have code built-in to them. This code will do the work of submitting the form or resetting the values to their defaults. A command button element has no code built-in to it. You write the code yourself for these buttons. Let’s see what the HTML code is for inserting one onto your form. It’s this:

<INPUT TYPE = Button NAME = "" VALUE = "Button Text">

Note the TYPE is “Button”. The NAME is used as a reference to distinguish it from any other buttons on your form. The VALUE attribute is the text that appears on the button itself. This can be anything you like.

To add a Command Button to your Form with the Editor, do this:

- From the menu bar, click on **Add**
- From the drop down menu, click **Add a Form Element**

- From the sub menu, click **Command Button**
- The Command Button code is inserted onto your page

Image Button

Command buttons look rather dull and grey. You can replace the grey command buttons with an image of your own. Then when the image is clicked on, any code underneath your button will get executed in the normal manner. The code for an Image Button is this:

<INPUT TYPE = Image NAME = "" SRC = "">

Note that TYPE is now not “Button” but “Image”. If you’re using an image as a button, the browser needs to know the source. In other words, the place where the image is stored. So you need the SRC attribute. Something like:

SRC = “MyButtonImage.jpg”

Hidden Elements

You can have something called an Hidden Element on your forms. Hidden elements are for your benefit, not your users. You can store information in a hidden element, and then pass the information to a second web page or form.

The Code for a hidden element is this:

<INPUT TYPE = Hidden NAME = "" VALUE = "">

As the name of the element implies, any data stored in the VALUE attribute does not show up on the page. However, if a viewer clicks **View > Source** in Internet Explorer, your code can be seen, and that includes any values you’ve hard-coded into the Value attribute of the Hidden element. So you wouldn’t use the Hidden element to store sensitive data.

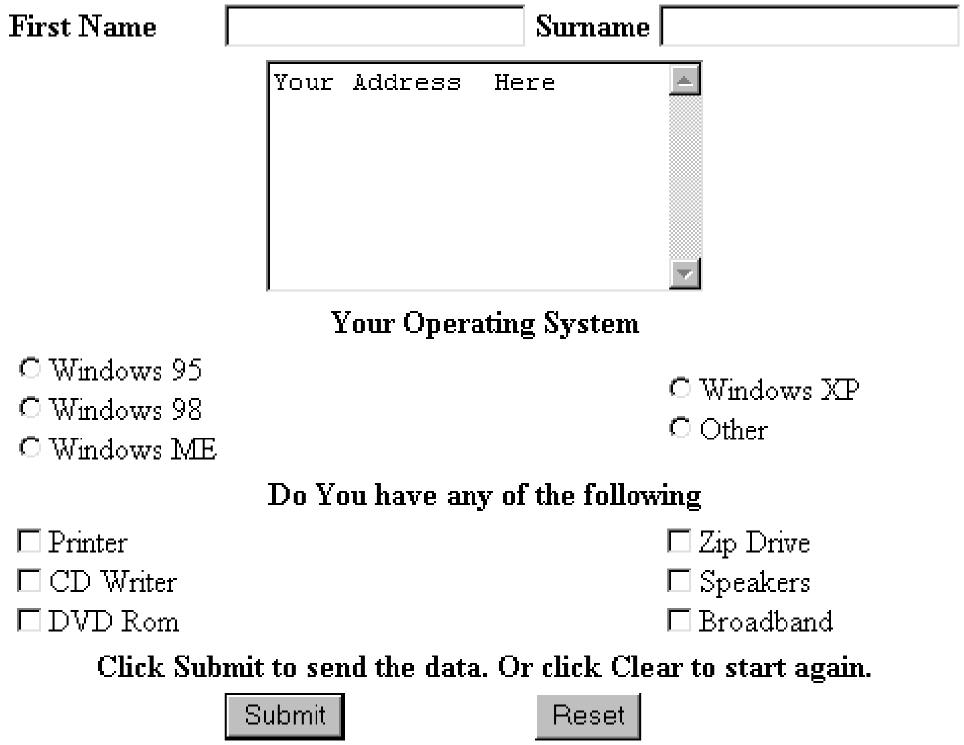
To add a Hidden Element using your Editor, do this:

- From the menu bar, click on **Add**

- From the drop down menu, click **Add a Form Element**
- From the sub menu, click **Hidden Form Element**
- The Hidden Element code is inserted onto your page

Review Number 6

Tidy up the form your created for this section. The new form should be put into a table, so that the elements are aligned. Create something like the one below:



First Name ▭ **Surname** ▭

Your Address Here

Your Operating System

○ Windows 95
○ Windows 98
○ Windows ME
○ Windows XP
○ Other

Do You have any of the following

□ Printer
□ CD Writer
□ DVD Rom
□ Zip Drive
□ Speakers
□ Broadband

Click Submit to send the data. Or click Clear to start again.

Though, as you are the web designer, you can add any colour scheme that you like. The end result should be a form with elements all nicely lined up and balanced.

Frames

The best way of understanding what a frames document is and does is to see one in action. So load up the web page called frames1.html that is in the Frames folder of your downloads. Click a hyperlink on the left hand side and watch what happens. Click a link on the right hand side, and again watch what happens.

What is happening is that 2 web pages have been joined together, or put into a frame. Frame targeting is then used with the hyperlinks to insert a web page into either of the 2 we started with.

If none of that makes sense, don't worry – it will become clear when you design your own frame document.

Creating a frames document can be split into two stages – setting up the frame document itself, and hyperlink targeting.

Setting up the frame document

When you open up a normal web page, you are requesting that a single page be placed in the browser window. With a frame document, you are requesting that 2 or more web pages be placed in the browser window. To do this, you need some special instructions for the browser. To tell the browser that you want to split the single window into frames, you need to use FRAMESET code. This is the code for the frames1.html frame document used as an example.

```
<HTML>

<FRAMESET Cols = "25%,75%" border=0>

        <Frame NAME = frame1 SRC = "page4.html">
        <Frame NAME = frame2 SRC = "page1.html">

</FRAMESET>

</HTML>
```

You can use your Editor to insert this code for you, so you don’t have to type it out yourself. We’ll see how to do that shortly. First, an explanation of the code.

<FRAMESET Cols = "25%,75%" border=0>

After the <HTML> tag, you set up a frame by typing the word FRAMESET. This tells your browser that you want to split the single window into frames.

Next, you need to tell the browser what kind of splitting you want. Do you want to split the window horizontally, vertically, or a mixture of both? In the code above, we’ve specified that we want the window to be split vertically (in columns). The attribute we’ve used is **Cols**. After typing an equals sign, you then tell the browser how many columns you want, and what percentage of the screen to allocate each column:

Cols = "25%,75%"

We only wanted two columns. We wanted the first column to take up 25 percent of the screen, and the second column to take up 75 percent of the screen. The total adds up to 100 percent. (If you prefer, you can use pixel size instead of a percentage. But the calculations are easier to make using percentages.) You could, if you wanted, specify that the screen be split into four columns. You’d do it like this:

Cols = "10%, 40%, 40%, 10%"

If you wanted to split your screen horizontally, the attribute to use is ROWS:

Rows = "25%, 75%"

Note the use of commas to separate the two values. Miss them out and your entire frames document refuses to work.

We also added a Border attribute to the FRAMESET tag, and set the value to zero. This will ensure (in most modern browsers) that those awful grey frame borders will disappear.

Because we've split our window into two columns, the next thing we need to do is to tell the browser which pages are going to go into our two columns. We do this with the FRAME tag:

<FRAME SRC = NAME = >

We need one of these for every column or row in our frame document. After typing the word FRAME we need to tell the browser where the page is that is going to go into this first column. The first FRAME tag you use relates to the first value you use in the COLS or ROWS equals attribute. So our 25% column will take the web page specified in the first FRAME tag, and our 75% column will take the second web page.

<FRAME SRC = " page4.html" NAME = frame1>

In other words, we're telling the browser "Put the page called page4.html into the first column. This first column has a size of 25%." (You might have noticed that the SRC and the NAME attributes have swapped places. This has been done deliberately to demonstrate that the order of the attributes is not important. SRC doesn't have to come first, nor does NAME.)

The NAME attribute is very important. You use the NAME attribute in the target link. You can use anything you like for the NAME. We've called our first column "frame1". (Your HTML Editor will insert a default name for you.)

Our second column also needs a FRAME tag. It was this:

<FRAME NAME = frame2 SRC = "page1.html">

The page that is going into our second column is called "page1.html". The NAME we have given to our second column is "frame2".

To recap, then:

- You set up a frames document with the FRAMESET tag
- In the FRAMESET tag, you specify whether you want to split your frame document into Cols or Rows
- You give each Row or Column a value, which represents the area of the screen each frame will take up

- Each row or column in your frame needs a FRAME tag. The FRAME tag tells the browser which page is going into your Row or Column
- Each FRAME tag needs a SRC attribute. The SRC attribute tells the browser where to find the web page that is going into the Row or Column
- Give each FRAME tag a NAME. The NAME will be used for targeting

Frame Targeting

Once you have your FRAMESET tags all set up, you can move on to stage 2 – Targeting.

The whole point of having a frames document is so that your visitors don't have to jump from page to page in your web site. When they click on one of your links, you can keep them on the same page. The page they would have been taken to when the link is clicked on will now be displayed in one of your frames.

To insert a web page into one of your frames, you add a TARGET attribute to the link itself. A normal hyperlink will look like this:

<A HREF = “Page2.html”>Click here for exciting news</A>

When that link is clicked, the entire page that the user was looking at will disappear. This Page2.html will then take its place.

But in a frames document, you don't want the entire page to disappear. You want this Page2.html to appear in one of your frames. In our example, we had a frames document split into two columns. When a link on the left was clicked on, a new page appeared on the right hand side.

The reason it did this is because a TARGET attribute was added to all of those links on the left hand side (and the ones on the right). The TARGET attribute uses the NAME value from your FRAMESET document. When we split our page into two columns, we gave each column a NAME. The NAME of our first column, the 25% one, was **frame1.** The NAME of our second column was **frame2**. To get the web page called Page2.html to appear in the right hand column, we use the NAME of the right hand column in the link, right after the

TARGET attribute. Like this:

```
<A HREF = “Page2.html” TARGET = frame2>
            Click here for exciting news
</A>
```

The link is practically the same. The only difference is that TARGET attribute. We’re now saying to the browser “When you bring me the web page called Page2.html, don’t fill the whole window with it. Instead, put it into the frame column that has the NAME **frame2**.”

And that’s the key to frames – Using a NAME from your FRAMESET code to TARGET a particular Column or Row.

OK, that’s enough theory. Let’s use the HTML Editor to create a frames document for us.

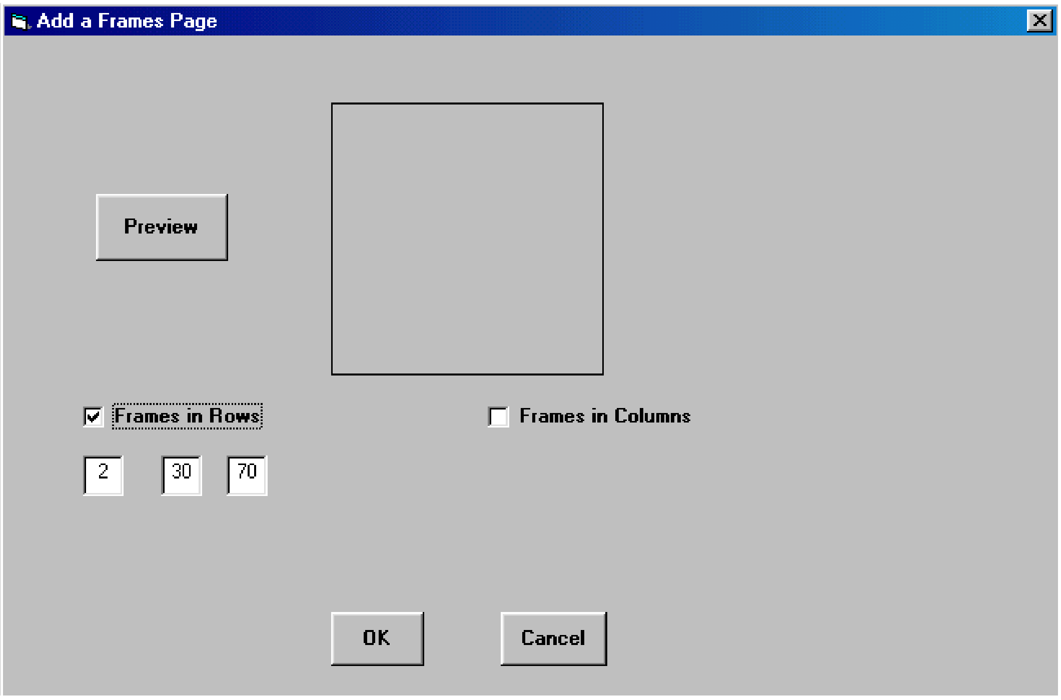
First, let’s create two pages to go in our frames, and a few pages for the links – five in all. You can do this quite easily with your editor. To create five web pages quickly, do this:

- Start your HTML Editor and create a new web page in the normal way. Set a white background colour.
- Click **File > Save As**
- Create a folder for your web page, call it Frames
- Type leftside.html as the filename
- Click **Save**. You’ll be returned to your HTML Editor
- Click **File > Save As**
- In the filename box, type rightside.html for the name of your second page. Then click **Save**
- When you get back to the editor, click **File > Save As** again
- Save the web page as page1.html, and click **Save**
- Click **File > Save As** yet again, and this time save the web page as page2.html. Click **Save**
- Create a fifth web page called page3.html

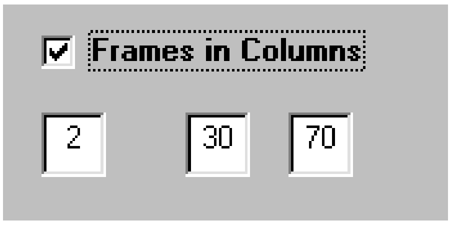
You will now have created five web pages very quickly, and we have everything we need to start creating our frames document.

To create your FRAMESET document using your Editor, do this:

- Click on **Add** from the menu bar
- From the drop down menu, click **Frames Document**
- From the sub menu, click **Add a Frames Document**
- The following dialogue box will appear



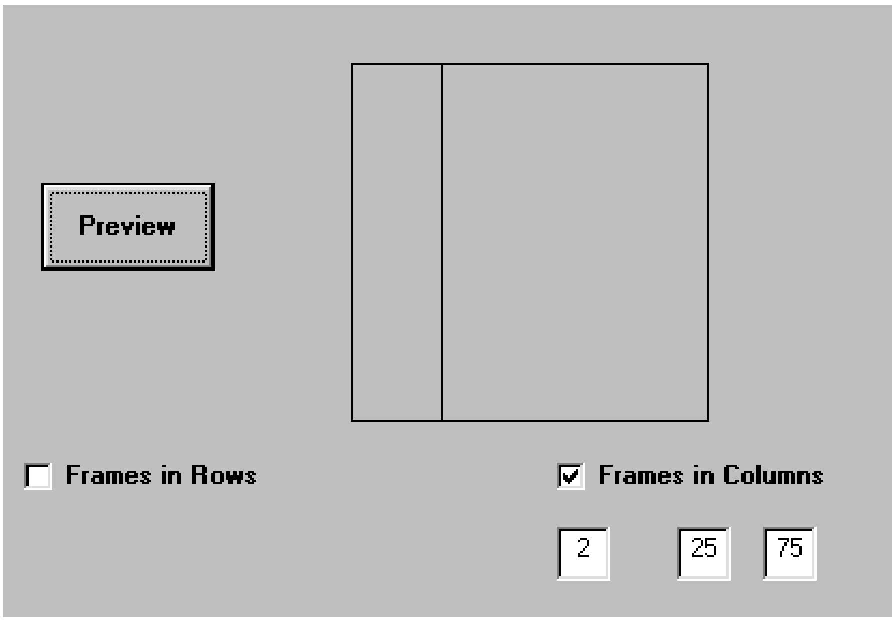
- We're creating a frames document in Columns, so click the box next to "Frames in Columns"
- The following appears



The first figure, 2, is how many Columns you want. The default value is fine for us. The next two boxes are the percentage figures. The default is to have the first column take up 30 percent of the frame and the second column 70 percent.

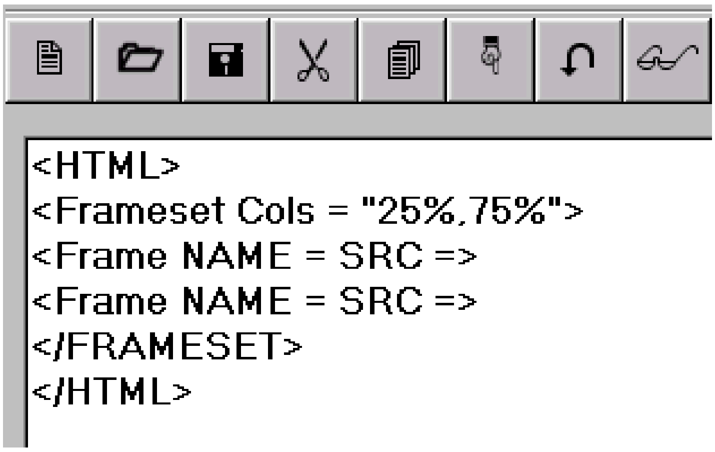
- Change the 30 to 25, and the 70 to 75
- Click the Preview button
- You'll see that square in the middle change to this:

Frames Preview



The line represent the border between your two columns. Play about with the numbers and click Preview to see what happens.

When you're happy, click the OK button. The following code will be inserted into your Editor.

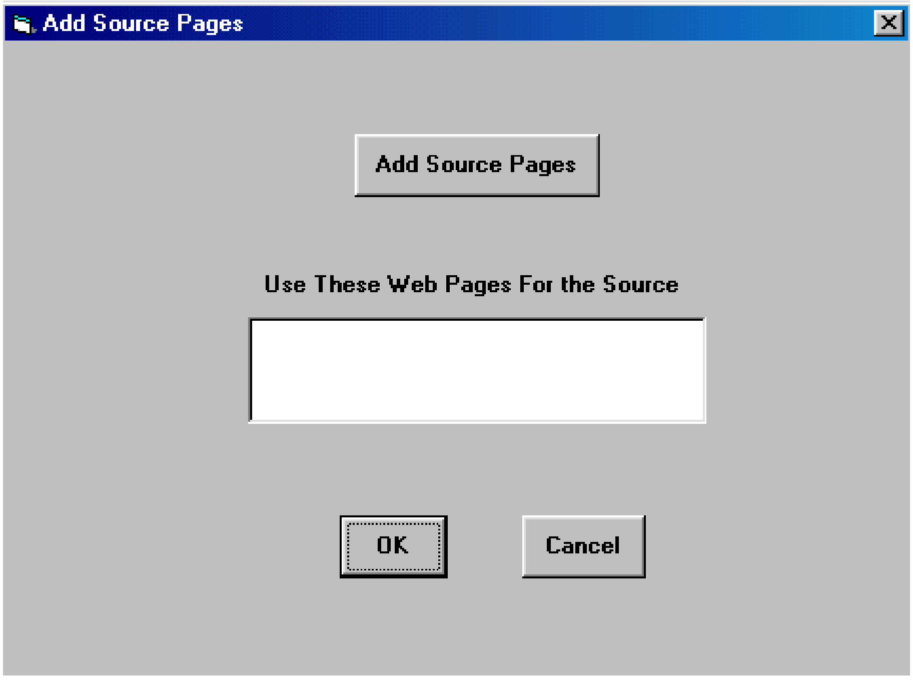


```
<HTML>
<Frameset Cols = "25%,75%">
<Frame NAME = SRC =>
<Frame NAME = SRC =>
</FRAMESET>
</HTML>
```

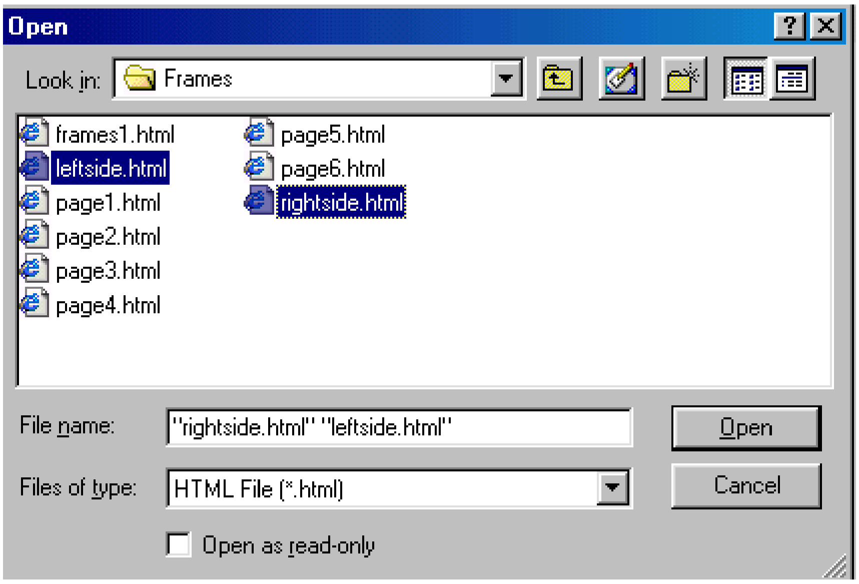
As you can see, the Cols have been set to 25 percent and 75 percent. Note the two FRAME tags. The NAME itself and the SRC are blank. We can use the Editor to set the pages going into our two columns, and set the NAME value. So:

- Click on **Add** from the menu bar
- From the drop down menu, click **Frames Document**

- From the sub menu, click **Add Source Pages**
- The following dialogue box appears:



- Click on the button “Add source pages”
- The open dialogue box appears
- Navigate to where your Frames folder is. The two web pages you saved earlier, leftside.html and rightside.html, should appear



Look in: Frames

frames1.html
leftside.html
page1.html
page2.html
page3.html
page4.html
page5.html
page6.html
rightside.html

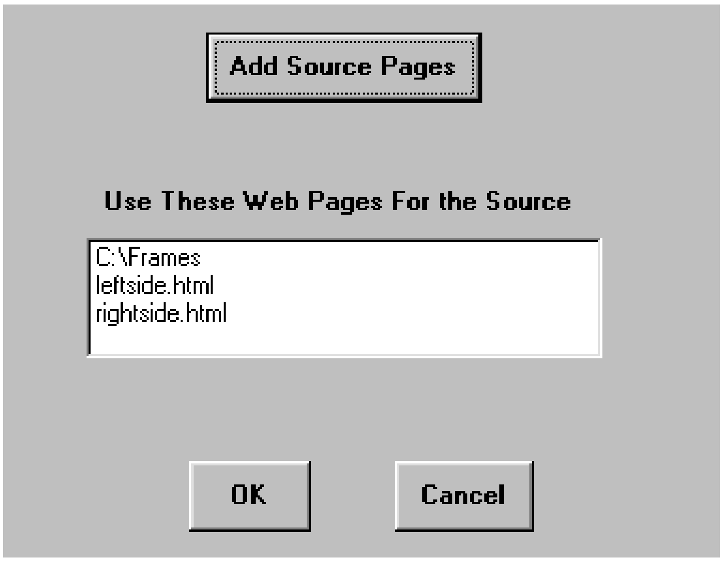
File name: "rightside.html" "leftside.html"

Files of type: HTML File (*.html)

□ Open as read-only

- To select more than one web page, hold down the **Ctrl** key on your keyboard and click on the file with your left mouse button

- Click on Open when both the leftside.html and rightside.html are selected
- Your dialogue box will then look something like this one:



C:\Frames
leftside.html
rightside.html

- The first line of the text box under “Use these web pages for the source” is the absolute reference to your html pages. Here it’s “C:\Frames”, meaning the folder called Frames on the C drive
- Click the OK button
- The editor will then amend your Frameset code to look like this:

```
<Frameset Cols = "25%,75%">


<Frame NAME = frame1 SRC ="C:\Frames\leftside.html">
<Frame NAME = frame2 SRC ="C:\Frames\rightside.html">

</FRAMESET>
```

Note that the Editor has inserted the absolute reference to your web pages - **"C:\Frames\leftside.html"**>. If you’re uploading to the internet, or sending work to be marked, you need to change this to a relative reference.

The important attributes to note are the two NAME attributes – frame1 and frame2. We’ll be using these in our targeting.

But you can now save this frameset document by clicking on File > **Save As**. Call it **frameset1.html**.

Click on **View > View web page** to see the results of your labour. You should have a frame document with two white pages in it. There should also be a grey bar running down the page, splitting your two frames into columns.

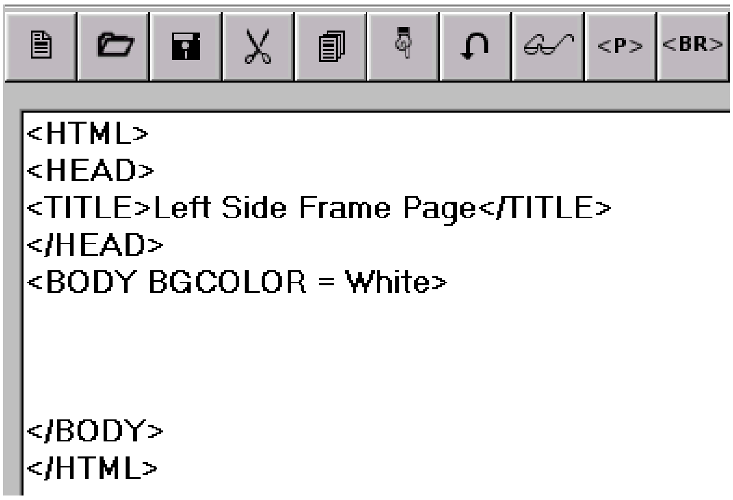
If you've got this far, congratulations – you have now created a frames page!

All right, so your frames page doesn't actually do anything. But the hard work is over. Only the targeting is left to do. For that, remember what our two columns are called? You're going to need those two names.

Inserting your Target links

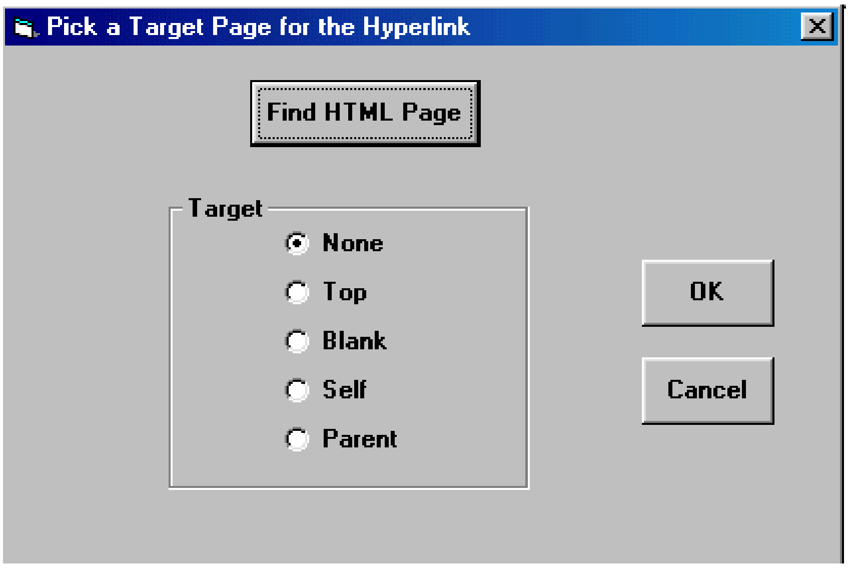
Now that we have our Frameset page working properly, we can turn our attention to the targeting.

- In your HTML Editor, click on File > Open
- The Open dialogue box appears
- Open the web page **leftside.html**
- This is just a normal web page, and should look like this:



```
<HTML>
<HEAD>
<TITLE>Left Side Frame Page</TITLE>
</HEAD>
<BODY BGCOLOR = White>



</BODY>
</HTML>
```

- Insert a Hyperlink by clicking on **Insert** from the menu bar
- From the drop down menu, click on **Link**
- From the sub menu click on **Target Hyperlink**. The dialogue box appears:



Target

● None
○ Top
○ Blank
○ Self
○ Parent

- Ignore the Target part of that dialogue box. That's for normal hyperlinks. Instead, click the button "Find HTML Page"
- The Open dialogue box appears
- Locate the web page that you saved earlier called page1.html and click Open
- The link that is inserted will look like this:

<A HREF = "page1.html">Link Text Here</A>

- Click your cursor after the **html** part and type a space
- Then type TARGET = (it doesn't have to be in capital letters)

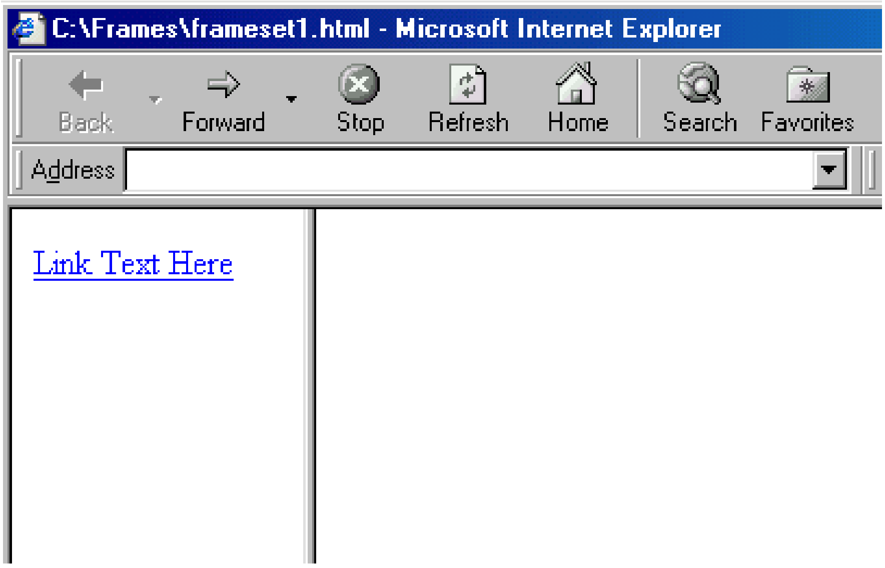
<A HREF = "page1.html" **TARGET =** >Link Text Here</A>

Now is where the NAME from your Frameset document comes into play. So type the NAME of the frame you want to target:

```
<A HREF = "page1.html" TARGET = frame2>
            Link Text Here
</A>
```

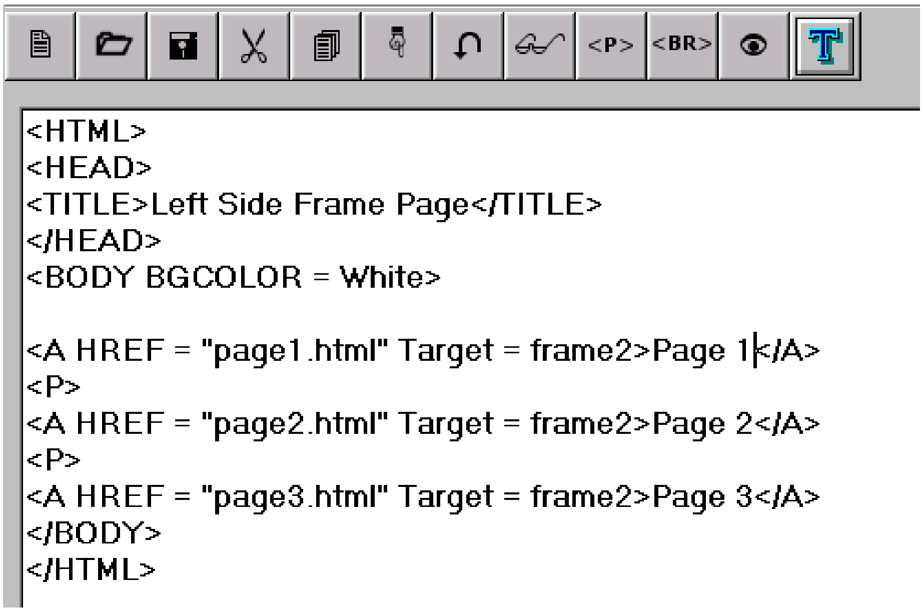
(You can tidy up your code like the one above, making it easier for you to read. This won't effect how it looks in a browser.)

Click **File** > **Save** to save your work, and reload your frame document. It should now look like this:



You can click the link to see if it work, but as your web page called page1.html is also a blank white page, you won’t notice any difference. To remedy this, open up page1.html. Type some text in between the two BODY tags. Save page1.html again, and then reload the frames document. Now click your link. You should see the text you typed appear in the right hand side frame.

Insert another two links into the leftside.html page, and the code will look like this:



```
<HTML>
<HEAD>
<TITLE>Left Side Frame Page</TITLE>
</HEAD>
<BODY BGCOLOR = White>

<A HREF = "page1.html" Target = frame2>Page 1</A>
<P>
<A HREF = "page2.html" Target = frame2>Page 2</A>
<P>
<A HREF = "page3.html" Target = frame2>Page 3</A>
</BODY>
</HTML>
```

Notice that the TARGET is exactly the same for all three hyperlinks. The difference is with the HREF attribute, which has different web pages for each link.

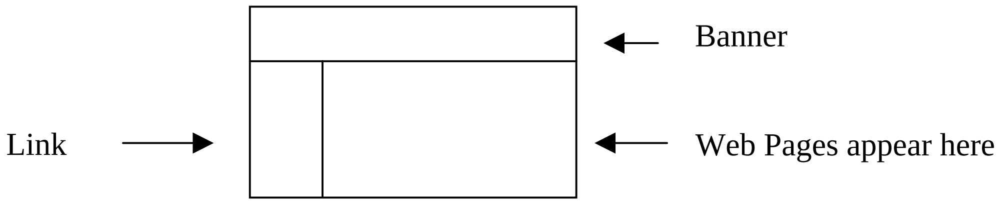
The only thing left to do is to open up page2.html and page3.html. Type some text into these two pages, save them and then reload your frames page. When you click on each link, page 1, 2 and 3 should appear in the right hand frame.

And that's it for a simple frame document. As an experiment, open up the frameset.html page in your Editor and change Cols to Rows. Save the web page and view it in your browser to see the difference between Cols and Rows.

Frames with Rows and Columns

You can mix Rows and Columns in the same Frameset document. It's not done much in web design, these days. But it's useful to know how to do it.

If you want a frame split like this one below:



Then the way to do it is to nest one set of FRAMESET tags inside another. The frame in the diagram has two Rows. The second Rows is split into two columns. So the code would be this:

```
<FRAMESET ROWS = “20%, 80%”>

<FRAME NAME = frameOne SRC = RowOne.html>

<FRAMESET COLS = “25%, 75%”>
        <FRAME NAME = frameTwo SRC =  ColOne.html>
        <FRAME NAME = frameThree SRC = ColTwo.html>
</FRAMESET>

</FRAMESET>
```

It looks a little complicated, but note that there are now two sets of FRAMESET tags, each with there own start and end tag.

The first FRAMESET Tag sets out the rows

<FRAMESET ROWS = "20%, 80%">

Into the 20% Row has gone the first FRAME tag, as normal. But the 80% row is now not a FRAME tag but another FRAMESET tag. So you are filling that 80% with two FRAME columns. The two FRAME columns will have a size of 25% and 75%

<FRAMESET COLS = "25%, 75%">

The FRAME tags for this new FRAMESET code are as normal:

```
<FRAME NAME = frameTwo SRC =  ColOne.html>
<FRAME NAME = frameThree SRC = ColTwo.html>
```

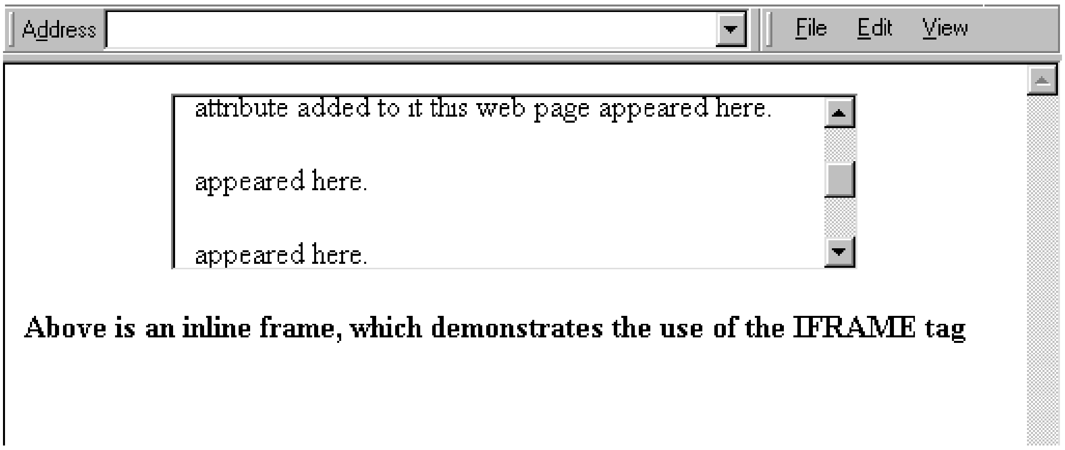
The first of those FRAME tags will go into the 25% column of the second Row, and the other one will go into the 75% column.

Finally, the two FRAMESET tags have their end tags </FRAMESET>

Inline Frames

You can add something called an inline frame, or IFRAME, to a web page. (However, the Firefox browser doesn't recognize the IFRAME tag, so you'd have to be certain that all your users will have Internet Explorer if you're going to insert an IFRAME onto your page.)

An IFRAME looks like this:



attribute added to it this web page appeared here.

appeared here.

appeared here.

Above is an inline frame, which demonstrates the use of the IFRAME tag

The code for an Inline Frame goes into a normal HTML document, between the two BODY tags. This code is this:

<IFRAME SRC =page2.html Width = 70% height = 40%>

</IFRAME>

The first IFRAME tag needs a Source (SRC) attribute and a width and height. Don't forget the end IFRAME tag </IFRAME>.

Inline Frames can present security problems, which is why Firefox is so touchy about them. They can be quite handy, though.

Other Frame Attributes

There are a number of useful attributes that you can add to FRAMESET and FRAME code. You've already met the BORDER attribute, which when set to zero will hide those awful grey bars. Here are some attributes that you should be familiar with.

Frame Margin

By default, the contents of each frame will have a border around the contents of 8 pixels. You can change this margin with the MARGINWIDTH and MARGINHEIGHT. The attribute goes inside the FRAME tag whose margins you want to change. Like this:

<FRAME NAME =f1 SRC = p1.html MARGINWIDTH = 10 MARGINHEIGHT = 20>

Or you can remove the margin altogether by setting both to zero. (Not the word zero, the number 0)

NORESIZE

If you have the grey borders showing in your frames, the size of each of your frames can be changed by the user. If the cursor is placed over the grey bar, the user can click with the left mouse button and drag the border to a new position. To prevent them doing this, use the NORESIZE attribute. It doesn't need a value. The attribute goes inside the FRAME tag whose border you don't want to be changed:

<FRAME NAME =f1 SRC = p1.html **NORESIZE**>

You can decide whether a frame should have a scroll bar or not. The attribute to use is SCROLLING followed by either Yes or No. The default for browsers is to show scroll bars only when needed. Then, they will appear only if the page contents are longer than the screen. You should only use this tag if you're certain that no scroll bars should appear at all on your pages. Again, the attribute goes in the FRAME tag whose scrolling you want to switch off:

<FRAME NAME =f1 SRC = p1.html **SCROLLING = No**>

No Frames

Some earlier browsers are just not capable of displaying frames at all. In which case, the NOFRAMES tag comes in handy. It's a rare visitor these days, though, who doesn't have a browser that can't support frames. Still, it's considered polite to include the tag, anyway. You use it like this:

…FRAMESET code here

</FRAMESET>

<NOFRAMES>
<BODY>
This page uses frames, but your browser doesn't seem to support them. Our alternative page can be found here **AltPage**

</BODY>

</NOFRAMES>

</HTML>

Notice the use of the two BODY tags inside the **NOFRAMES** tags. The text you want to display if the user doesn't have a browser that supports frames is then typed between the two BODY tags. This can be anything you want.

And that wraps up our section on frames.

And that's it for the HTML course. There is a lot of learning to do if you want to master HTML scripting. At this stage, you may have to dip in to the book quite a lot when you’re designing your pages. But that’s to be expected. Once you’re creating web pages on a regular basis, you’ll find you need the book less and less. By this stage, you’ll want to expand on what you have learned. If you do, then you can’t do better than to learn all about Cascading Style Sheets. These will really bring your site to life! We have a CSS section on our web site. So when you’re ready – tuck in!

We wish you all the very best for your coding future.