

## Introduction

OK, you have decided to learn HTML. Being a wonderful musician, writer, poet, digital artist, painter or whatnot is not enough of a creative outlet for you and the preset templates that we provide you with here at eVille do not satisfy you. You want to make your own pages. I can hear you now *“No offense, E. H., but the templates are so corporate looking, and a touch dull. We want our pages to look more like yours, dynamic and interactive to reflect our brilliance. How do we do it? How do you do it?”* Well, forget it. It took me years of intense study to learn to make pages like that...OK, I am lying to you. It took me a single afternoon to learn the basics of HTML and it won't take you any longer. After all, you have me to teach you, an advantage I certainly didn't have.

OK, by now you are undoubtedly wondering how much this is going to cost you. You have priced out Macromedia Dreamweaver and Adobe GoLive and wished that your art paid well enough to purchase either. You probably took a gaze at Micro\$oft Front Page and thought about joining the Dark Side, except that you know that people will laugh at you for using it. So, you undoubtedly want to hear me recommend something to you and are wincing at the thought of how much it is going to cost you. Have your wallet in hand? Ready to hear how much you are going to spend on a top-flight HTML editor? Nothing. Not a single red penny. HTML documents are just that, documents, and are best written inside of text editors. *“Text editors?”* you say, *“You mean like Simpletext? (or Notepad for you Windows users)”* Precisely. Though there are other better choices for text editing.

BBEdit Lite is the freeware version of BBEdit, the best editor bar none for the Macintosh. Once you are proficient at writing the code by hand, you will want to spend the money on the full commercial version, don't be afraid, it is still a text editor and not near so expensive as the visual editors. For those of you using Windows there are several excellent choices. First amongst the lot is Editpad, a strict text editor with no extras. Small, fast and stable the complete opposite of most Windows software. There is also Note Tab Lite, a program I highly recommend as it has an exhaustive list of HTML tags that it keeps docked on the left hand of your screen for you to refer to. On the high end of HTML editors is the freeware program 1st Page 2000 from Evrsoft, the best text editor bar none for Windows. Once you are comfortable writing the code by hand, you will want to use it, and best of all it's free.

OK, I can hear you complaining *“E. H., I am an artist, not a geek. Isn't there a program out there that won't require me to do any programming? Something visual like GoLive, but not a sorry joke like FrontPage?”* Well, if it is any comfort to you, I was the same way once. Terrified that learning to write HTML by hand would cause me lose my wit and style and become a pale skinned (OK, paler skinned) geek with weak eyes and large hands. OK, so I already have weak eyes and large hands, but I was still afraid of losing my wit and style. I was overjoyed, however, once I got the hang of it (grok in geekspeak), writing HTML by hand turned out to be completely natural, an extension of my artistic nature. As a writer, I would envision the scenes for my tales in my head, when I had the scene set in my head, I would then try to find the words to describe them on paper. To my everlasting joy writing HTML turned out to be exactly the same, I would envision the page in my head, when I knew what the page ought look like, I would then write the code to describe the page to the web browser.

*“E. H., I'm a guitarist, that's not how I work.”* OK, here it is. If I am teaching, you, my students, are going to learn to do things the right way. That means you will learn to write by hand, even if you later move on to a **WYSIWYGITA** (What You See Is What You Get, In Theory Anyway) editor, you will know enough HTML to edit the resulting source code, and you will learn to write it correctly. That last is my way of telling you that, unlike other tutorials, I will not teach you HTML tags that the World Wide Web Consortium (W3C) has targeted for extinction, referred to as deprecated tags. As these tags are soon to disappear I do not want you to get in the habit of using them. OK, have you finished downloading your HTML editor? Then I suppose it is time for the first lesson, *The Basics of HTML*.

## The Basics of HTML

We shall start with a brief explanation of what **HTML** actually is. HTML is a markup language, not a programming language. You probably wonder about the difference, eh? Too bad, I am not here to teach you to program (not yet anyway, check back in a few months when we get to perl ) for now we are going to limit the discussion to HTML and web browsers. Web browsers are, by nature, quite stupid, they have no capacity for independent thought, you can instruct it to open up a document, but it needs explicit instructions upon how to display the data contained within the document. Web browsers grow rather confused when they are not told explicitly what to do; much like a politician does when he has no public opinion polls to guide him. HTML tags are those instructions, they tell the browser how to display the text, where to place the images, etc. Without proper instructions, the browser makes mistakes, that is its job, and in the case of Netscape Navigator/Communicator, a job that it excels at. From the outset, get used to the fact that Netscape is the cranky, old, government bureaucrat of web browsers. It will make you pay for every mistake, no matter how miniscule, much like an IRS auditor. Take two aspirin and get over it. You can try sending AOL hate mail, but as it hasn't worked for me, I doubt you shall have better luck, but don't let me dissuade you. When you have finished your complaint come back and see me and we will continue.

Ready? Excellent. As I indicated above, HTML is nothing more than the instructions that tell the web browser how to display the data contained in a document, however, the tags need to be hidden so that your readers don't see them. To hide the instructions from the end user, we bracket them out like this `<html>`. Anything contained within brackets like that will not be displayed, the brackets tell the browser that instructions are contained within that should not be displayed.

“Umm, E. H., correct me if I am wrong, but I can clearly see that `<html>` in the sentence above, did the web browser screw up? Or were you lying to us? How did you do it?” Don't get ahead of yourself, you can't even write html yet, and you want me to instruct you on the fine art of using special characters already? You bloody artist types are all alike. Now time for the first lesson.

Lesson number one; HTML documents are text documents saved with a .html extension. Now, here comes your part, open your text editor and start a new document. Use the save as function and name the document **testpage.html** and save it as a text file. Done? Excellent, you have created your first web page. Open the page in Netscape.

“E. H., Netscape is giving me an error message, it tells me that the document contains no data. What happened?” Well, we haven't put any data in, of course. That is lesson two.

Every HTML document has the same basic structure and you are about to learn it. Are you ready for the arcane formula by which we warlocks of the web make the pretty pages appear? Here it is,

```
<html>
```

```
<head>
```

```
<title> My Very First Web Page </title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Now, type that in to your text editor and save it. Go back to your browser window and hit refresh.

“E. H., I still can't see anything, though at least the error message is gone.” Really? Take a look at window's title bar. See what it says? The text that sits in between the `<title>` `</title>` is displayed there. The text between the title

tags appears in the title bar, and, on a side note, if someone happens to bookmark the page, that is the name that displays for the bookmark.

“E.H., what’s with all those `</whatever>` tags?” In HTML, we open and close tags (with very few exceptions), so to open an HTML document, we use the `<html>` tag, to close the document we use the `</html>` tag, the `</>` literally means “end” so `</html>` would be “end HTML”. This becomes even more important inside of the document’s body, as I will explain when we reach the appropriate section. Now, as you are quite clever, you noticed that I opened and closed the tags in order. This one of those things you will ALWAYS do, close the tags in the order in which they are opened. Though not noticeable here, once we get to *tables* it will be vital, because there closing tags in the wrong order will cause some truly bizarre errors.

Now, the content of the page will be what you enter between the `<body>` `</body>` tags. For a start, let us type something in to that section of the document, how about “Mashed potatoes with horse radish sauce is my favourite Thanksgiving treat.” Now, save the document and switch back to your browser window and hit refresh. See?

“Hey, that seems easy enough, it looks a little dull, but even I can do this, thanks for all your help, E. H.” OK, I see you will need learn the hard way, come back in a few moments when you are ready.

“E.H., I just did what you taught me, I entered this really great poem of mine between the body tags, when I typed it in it looked like this,

```
Sometimes I feel my life is a meteor falling  
Destined to burn brightly a moment or two  
Ever feeling the cold earth calling  
Knowing I perish with the flame
```

But when I opened it in my browser window it looked like this,

```
Sometimes I feel my life is a meteor falling  
Destined to burn brightly a moment or two  
Ever feeling the cold earth  
calling  
Knowing I perish with the flame
```

What happened?”

Well, simple, you entered the text, but gave the browser no instructions on how to display it. It did its job and displayed the text absent any formatting. We need to give the browser explicit instructions, and with that, it is time for part two, formatting text.

## Formatting Text

OK, you want to present to the world your magnum opus (the poem from the previous section), but when you viewed it in your browser it appeared absent formatting, so now you need to know how to format text. There are two types of tags that we will deal with, in this first section; we will deal with general modifiers, positioning tags and style modifiers. Now, first let us have a look at your masterpiece again;

Sometimes I feel my life is a meteor falling  
Destined to burn brightly a moment or two  
Ever feeling the cold earth calling  
Knowing I perish with the flame

What we need here is a tag that tells the browser where to break the lines, and for that we have the `<br>` or *Line Break* tag, this tells the browser to place everything after the tag on another line, so, let us give it a try, shall we? Apply the `<br>` tags to the line endings like this;

Sometimes I feel my life is a meteor falling`<br>`  
Destined to burn brightly a moment or two`<br>`  
Ever feeling the cold earth calling`<br>`  
Knowing I perish with the flame

Now, hit refresh in the browser window, look better? “OK, E. H., that looks simple enough, but, I have a few two stanza pieces, and I need to double space between stanzas, do I need to put in an extra `<br>` tag? And, umm, not to question you or anything, but didn’t you say something earlier about always closing your tags?”

One question at a time. You artists are all so impatient. The `<br>` tag does not need to be closed, yet. Shortly it will need to be written `<br/>`, but that is part of the **XHTML** tutorial, so we shall revisit the matter then. For now it is enough to know that the `<br>` tag is not one that is applied to text, it just breaks everything after it down a line, so it need not be closed.

As for the question about multiple line breaks, you can do that, or you can use the *paragraph or* `<p>` tag. The paragraph tag puts in an extra line break for you, so let us turn back to that poem of yours, now, as you chose one a single stanza long, we shall need add some more text below, so let’s try it like this;

`<p>`Sometimes I feel my life is a meteor falling`<br>`  
Destined to burn brightly a moment or two`<br>`  
Ever feeling the cold earth calling`<br>`  
Knowing I perish with the flame`</p>`

`<p>`This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd rejection notice. Those bastards, I should beat them up.`</p>`

Now, hit refresh in the browser window and have a look. See? We have now double-spaced between the two paragraphs. Let’s try some more formatting, admittedly, what we have looks a touch awkward with everything aligned to the left hand margin of the page, so let’s center the poem above the text below it. How you ask? Easy, we use the `<center>` tag. Now, let us return to our page and add the `<center>` tag;

`<center>`  
`<p>`Sometimes I feel my life is a meteor falling`<br>`  
Destined to burn brightly a moment or two`<br>`  
Ever feeling the cold earth calling`<br>`  
Knowing I perish with the flame`</p>`  
`</center>`

`<p>`This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd rejection notice. Those bastards, I should beat them up.`</p>`

You know the drill, hit refresh. Now as you can see we have centred the poem above the text beneath it. For those of you aspiring to be the next e. e. Cummings, you have by now noted with some distress that no matter how many spaces you put between your words, that only one space appears between words. You are no doubt raging and cursing the limitations that the web is placing upon your artistic genius, well, you may calm down, some of you will no doubt begin to understand that the electronic medium will open up ways of presentation that simply do not take on paper. As for the spacing problem, there is an easy solution, the *non-breaking space*. What is that you ask? It is the **&nbsp;**; (the semicolon is vital) tag, it is one of many special characters that HTML contains (more on other special characters later), this one inserts an extra space, with that in mind let us go back to your poem, shall we? Let us say that for some inconceivable reason you wished to set the word meteor apart from the rest of the text and wanted three spaces on each side of it, wishing to wade the waters of Cummings but still fearful of drowning, you make this one experiment. Back to the poem, shall we?

```
<center>
<p>Sometimes I feel my life is a &nbsp;; meteor &nbsp;; falling<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p>
</center>
```

<p>This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd rejection notice. Those bastards, I should beat them up.</p>

“Hey, E. H., you only put one of those non breaking thingies, I thought we were putting in three spaces, don’t I need two on each side?”

No. HTML recognises only a single space between characters, but the **&nbsp;**; is a character, so, the browser will recognise the spaces on either side of it. OK, you know the drill, it is refresh time. No, I meant in the browser window. OK, now we have meteor set off by three spaces on either side. The **&nbsp;**; is a handy tool for positioning items horizontally, nudging them a few spaces to the left or right.

Now, for a few items of font style, let us say that you are an aspiring essayist, and more, aspire to be a literary reviewer. More, being a person of astounding wit you decide to review my novel, Saki’s Bastard Son. OK, so my novel won’t be released for several months yet, and if I am taking the time to teach you HTML, you had best repay the favour by purchasing it. As you are still a person of discriminating taste in literature you decide to write an essay on Dawn Powell’s *The Golden Spur* (and may I commend you on a most excellent choice). No doubt wish to know how to italicise text for book titles, that is simple, we use the **<em>** or *emphasis* tag. Let us have a look at it in action, shall we? Put in your opening sentence;

<p>Recently while looking through the literature section of my local bookstore I came across a thirty five year <br> old book that recently achieved best seller status, Dawn Powell’s **<em>**The Golden Spur**</em>**.</p>

Now, hit the browser refresh and have a look. See how the **<em>** tags have changed the text? Next, we will tackle bold-faced type. To give type a bold face effect we use the **<strong>** tag. Let us try it, shall we?

<p>Recently while looking through the literature section of my local bookstore I came across a sixty year <br> old book that recently achieved best seller status, **<strong>**Dawn Powell’s**</strong>** **<em>**The Golden Spur**</em>**.</p>

Now you will see that the author’s name appears in bold. Here I shall give you a few other stupid text tricks and let you try them out on your own. To create subscripted text (as in H<sub>2</sub>O) we use the **<sub>** or *subscript* tag, in this case place it about the character you will be subscripting (H**<sub>**2**</sub>**O). To superscript a character (as in 3<sup>rd</sup>) we use the **<sup>** tag, and the same rules apply. To create underlined text, we use the **<u>** or *underline* tag. Strictly speaking the **<u>** tag is deprecated, that is it has been targeted for extinction, however, as its replacement will be handled with **Cascading Style Sheets** (CSS), I will permit you to use the **<u>** tag until I have taught you better. Mind you, once I have posted the CSS tutorial, I shall expect you all to do it properly, and not resort to using the older and soon to be archaic tags.

“OK, that’s easy enough, I want to put the title of my poem/essay/short story on my page, is there a simple way to differentiate between the title and the rest of the story?” Why, yes, there is, the <H> or *headline* tag. A number, from <H1> for the largest to <H6> for the smallest will always modify the headline tag. In an ironic aside, when you learn to use the font tag, you will learn that 6 is the largest typeface and 1 the smallest.

“That makes no sense, E. H., why do they have to confuse us like that?” Well, my theory is that HTML was designed by pasty-faced individuals with large hands and propellers on their hats with no sense of style. Or perhaps after a lifetime of being beaten and ridiculed by the rest of us, the people who designed HTML are taking their vengeance upon us. It’s just a theory. Either way it does no good to complain, you simply must learn to live with it.

Now, there are several tags that I have not dealt with till now, I am only broaching them and showing you their use so that you **NEVER EVER EVER** use them. There are two of them, the <marquee> and <blink> tags, they do precisely the same thing, annoy the person viewing your web page. The <marquee> tag annoys the IE users and <blink> annoys the Netscape users. The <blink> causes the text in between the tags to blink on and off, while the <marquee> causes the text to scroll across the screen. OK, I can see you are determined to try it out, so let’s have a look at them in action, shall we? We shall place the <blink> about the word meteor in you magnum opus, and set off the paragraph below with a <marquee> so that you can see why you will **NEVER** use them again;

```
<center>
<p>Sometimes I feel my life is a &nbsp;<blink> meteor</blink> &nbsp;  falling<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p>
</center>
```

```
<p><marquee>This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd
rejection notice. Those bastards, I should beat them up.</marquee></p>
```

Hit refresh and cringe. I shall presume that you won’t use them again. The <blink> and <marquee> tags are for philistines, and if you are an official eVille genius, you are no philistine. Now, what next, any suggestions?

“E. H., I am looking at your page here, and your fonts are colourful and stylish and mine are so...Courier. How do I get rid of this ugly typeface?”

Well, simple, for that we use the <font> tag. The <font> tag allows us to define for the browser how we want the text between the tags to appear. There are three modifiers to the tag that must be defined for the browser, face, color, and size. One note about using modifiers, when you add the modifier, always put the value in quotes (we will see that in action presently). Now, here is the tricky part, before you go wild and start using something like Lila Funk as a font type for your page, keep in mind that you are going to be at the mercy of the fonts that the end user has actually installed on their machine. You are probably wondering what happens if the font you choose is not installed on the end user’s PC. Simple, it reverts to the default typeface. Stop cringing, there is an answer for the dilemma.

Let us have a look at the <font> tag in action, shall we? Now, let us turn back to that poem of yours, now, as we wish to ditch the Courier, we will change the typeface to a san serif to start with like this, and please note the quotes around the values I set for the modifiers;

```
<center><font face= “Arial, Helvetica, San Serif” size= “3” color= “black”>
<p>Sometimes I feel my life is a &nbsp;  meteor &nbsp;  falling<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p></font>
</center>
```

```
<p>This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd rejection notice.
Those bastards, I should beat them up.</p>
```

OK, hit the refresh and have a look.. Strictly speaking, in the <font> tag above I entered default values for the size and color attributes, in other words, if I leave out the size attribute, the browser will presume that I want the display size set to 3. Likewise will presume that the color attribute, if left undefined, is black. So, let us try an experiment, shall we? Let us be bold and daring and go where e. e. Cummings fears to tread. Go back to the poem and remove the size and color attributes from the first <font> tag, so that everything will default to size 3 black type, and let us add a SECOND font tag, leaving the face attribute blank but defining the color as red and the size as 4 and place it around the word meteor like this;

```
<center><font face= "Arial, Helvetica, San Serif">
<p>Sometimes I feel my life is a &nbsp;<font size= "4" color= "red"> meteor</font> &nbsp;<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p></font>
</center>
```

```
<p>This is my magnum opus, I wrote it on the eve of my 30th birthday, after receiving my 72nd rejection notice.
Those bastards, I should beat them up.</p>
```

Now hit refresh and have a look. Feel better now? OK, now, you are probably wondering about that face attribute, looking at how I defined it and wondering why. Well, by using font families we create many defaults for the browser to use. In the example above, the browser looks first for Arial, if it does not find Arial it moves on to Helvetica, etc... The idea here is to provide the browser with multiple fonts to default to, so that it does not give your viewers that awful Courier to look at. As noted in the previous section, the font sizes are the opposite of the headline sizes.

“I still say that it makes no sense, you would think we could rely on computer programmers to be logical.” My theory is that years of learning to develop in strange arcane languages have left the people who invented HTML with an alien form of logic that we normal humans simply can not "grok". It's just a theory.

Some of you, no doubt, bought HTML primers before turning to this tutorial, and are probably itching to ask me a question about the <font> tag, so go ahead, fire away. “E. H., you promised us that you would teach us to write the code properly, none of those tags destined for the dustbin of code history, what did you call them? Oh, yeah, deprecated tags. Well, according to my O'Reilly's HTML book, the <font> tag **IS** deprecated. What gives? Were you lying to us again?”

Well, I did say that there would be exceptions, and this is one of them. The way we are supposed to control font usage in the future is through Cascading Style Sheets and the use of <div> and <span> tags. However, those are topics for a more advanced tutorial than this one. I needed to teach you to modify font faces quickly and simply, and that meant the font tag. Once we get to CSS, you shall never want to go back to the old way, because you will find that with CSS you will be able to define font usage far more precisely. Now, as we are done with text, time to move on to the next lesson, *Colours & Backgrounds*.



we are in to the <body> tag modifiers, time for some new ones then. Let us say you are now feeling quite daring, you are ready to integrate your various artistic pursuits in to comprehensive works. In addition to being a wonderful poet, you are an aspiring illustrator as well, and you want to use an illustration of a meteor as a background image for the page. Well, simple enough, we have a modifier called bgimage, I shall give you three guesses what the bg stands for.

OK, now that we have come to the bgimage modifier, we are skirting dangerously close to site architecture, but, we shall successfully avoid it for now by placing all work in the same folder. The value for the background image must be the location, either relative or absolute, of the image file you are using. As relative is an easier designation to use, we shall start there. Place the image file (any image file will do, so long as it is either a .gif or a .jpg file) in the same folder as the HTML document and turn back to your poem and the head of the document, and this time we will add the bgimage modifier;

```
<html>

<head>
<title>My Magnum Opus</title>
</head>
<body bgcolor= "papayawhip" bgimage= "imagename.jpg">(whatever your image file may be named)
```

OK, time to have a look. I know what you are about to say, you want to know why the image keeps repeating, well, that is how HTML behaves. It starts in the upper left corner and starts repeating the image, both vertically and horizontally. It is useful, despite what you are thinking, there are times when you want a pattern to repeat across an entire page, and it is easier to design a 20pixel square than an 800x600 one. As you are still incredulous, allow me to calm your fears and tell you that, yes, we can control the repetition, eliminating it altogether, or limiting the repeat to the horizontal or vertical axis.

Now, I shall teach you the magic formula, and be prepared, because you are about to dip your toes in to the icy waters of **DHTML**. We use the **style** modifier to achieve this. The style modifier is used to apply dynamic styles to the HTML tag it is modifying. Clear as mud, eh? Let us have a look at it in action then, shall we? Now, as we are going to use the style modifier, we will remove the bgcolor & bgimage modifiers and replace them with this syntax;

```
<html>

<head>
<title>My Magnum Opus</title>
</head>
<body style= "background: papayawhip url(imagename.gif) no-repeat">
```

No, when you hit refresh you will see that your image has been placed in the upper left corner of the browser window, to repeat the image horizontally we would use **repeat-x** and for repeat on the vertical we would use **repeat-y**. You, no doubt, wonder why we would use repeat-x or repeat-y, no? Well, hopefully you all have an image editing program, even if it is naught more than Paint Shop Pro (some of you digital artists will be using ProCreate Painter or Adobe PhotoShop). Let us create a new image, one pixel in height and 120 pixels width and fill it with that color we used earlier, #330033. Now, save the file as border.gif, and let us go back to the head of your document and change the image file to border.gif and set the repeat to repeat-y as such;

```
<html>

<head>
<title>My Magnum Opus</title>
</head>
<body style= "background: papayawhip url(border.gif) repeat-y">
```

Now, when we hit the refresh, we will see that we have created a border down the left-hand side of the page. There are two more modifiers to discuss, actually, one with a lot of different names. Some of you with a keen eye have, by now, picked up on the fact that there is a slight margin on the top of the page and down the left side, and are wondering if there is an easy way to remove it. Yes and no. Yes there is an easy way to remove it, but it needs to be

done separately for each browser, that is, there is one set of modifiers for Netscape and another for IE.

“E. H., why can’t programmers all get along? Do they really need to make all this extra work for us?”

Have you not grasped by now that this is a conspiracy? The programmers hate you, that is why they do it. It is because you refused to date them in high school, barred them from attending the prom and humiliated them graduation night, this is their revenge, get used to it. Show them that you, for one, have moved on and grin and write the code. Now, back to that <body> tag, shall we? Now, in Netscape, the modifiers are **marginheight** and **marginwidth**, in this case we will set them to zero as such;

```
<html>
```

```
<head>
```

```
<title>My Magnum Opus</title>
```

```
</head>
```

```
<body style= "background: papayawhip url(border.gif) repeat-y" marginheight= "0" marginwidth= "0">
```

Now, open the page in Netscape and have a look, the margin has vanished, alas, if we look at it in IE we see the margin is still there, so, now we need to add the IE modifiers to eliminate the margin there. In IE the modifiers are **topmargin** & **leftmargin** (there are also **rightmargin** & **bottommargin** modifiers but you will rarely if ever use them) and we will now go back to our ever expanding <body> tag and add these new modifiers to take care of the IE issues as such;

```
<html>
```

```
<head>
```

```
<title>My Magnum Opus</title>
```

```
</head>
```

```
<body style= "background: papayawhip url(border.gif) repeat-y" marginheight= "0" marginwidth= "0" topmargin= "0" leftmargin= "0">
```

Now when we gaze at the page in IE we see that the margin is gone, as it is in Netscape. Congratulations, you are near finished with your basic layout lesson. We shall move on to the next lesson, one you digital artists are champing at the bit for, *Images & Links*.

## Images & Links

Nearly every web page you view has images built in to it and links to other web pages. As every other web page has these things, it is high time you learned to add them to your own. We will also get a quick lesson in site architecture, as we can no longer avoid it (the topic will be covered in full during the advanced HTML tutorial). As links are the simpler of the two, we will start with them. We add links to our web page with the `<a>` or *anchor* tag. We will place the anchor tags around the bit of text or the image that we want to use for the link. We must identify the location we want the user sent to using the *Hypertext reference* or `href` property, let's have a look, shall we? Now, back to that poem of yours, let us say that you know for certain that the reason it has never been published is that Disney, in league with General Motors and the CIA, is conspiring to prevent it from reaching the masses, for fear that people shall read it and awake from their self induced coma and realise that bowdlerised versions of classic fairy tales are not suitable family viewing. In light of this you decide to make the words "Those bastards" in that explanatory note a link to the Disney website, so that everyone knows how you feel about them. Here is how we do it;

```
<center><font face= "Arial, Helvetica, San Serif">
<p>Sometimes I feel my life is a &nbsp;<font size= "4" color= "red"> meteor</font> &nbsp;<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p></font>
</center>
```

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after
receiving my 72nd rejection notice. <a href= "http://www.disney.com">Those bastards</a>, I should beat them
up.</p>
```

Now, after we have modified the code, hit the browser refresh button and have a look, you will see that the text "those bastards" is now underlined (the default setting for links) and coloured blue. If you click the text, you will see that you have indeed been taken to the Disney website. If you hit the back button and look at the link, you will note that the text has now changed colour. This is the browser's friendly way of letting you know that you have visited it.

"E.H., is there anyway we can change that? The colour is awful and that underlining thing is tacky. Your links are only underlined when I move the cursor over them, how do I do that?"

Well, this should be part of a more advanced tutorial, but because I have taken a liking to you, I shall give you some instruction in the use of `<style>` tags. No, not the style modifier for other tags, there is a separate `<style>` tag, and here you will begin to learn to use it. Let us turn back to the head of the document, and we are going to add the `<style>` tag. The tag is placed inside the `<head>` tags, so, place your cursor at the end of the `</title>` tag and hit return. Now, add the `<style>` tags as such;

```
<html>
<head>
<title>My Magnum Opus</title>
<style type="text/css">
<!--
-->
</style>
</head>
<body style= "background: papayawhip url(border.gif) repeat-y" marginheight= "0" marginwidth= "0" topmargin=
"0" leftmargin= "0">
```

The `<!-- -->` is the comment tag, everything placed between those tags will be hidden from the browser, yes, they can be used anywhere in your document, and you can leave little comments for people who are viewing your source code to see. In this case, when used between the `<style>` tags, it is our way of letting the browser know that this is where the actual style instructions are to be found. The `type= "text/css"` is our way of letting the browser know what

type of style we are applying. Now that we have added the <style> tag, let us define a style, shall we? Now, inside the comment tags, we will add a style definition for the <a> tag using this syntax;

```
<style type="text/css">
<!--
A:active {color: #330033; text-decoration:underline;}
A:link {color: #000000; text-decoration:none;}
A:visited {color: #000000; text-decoration:none;}
A:hover {color: #330033; text-decoration:underline;}
-->
</style>
```

Note that we have used the curly brace when defining the style for the <a> tag. Yes, that is important. Now, when you hit refresh, we see that the link text looks like all the text around it, except when you run the cursor over it. For now (I will teach you more when we get to the CSS tutorial) limit your definitions to color and text-decoration when defining <a> tags and limit the values for text-decoration to underline and none. Of course, this only works in Internet Explorer, in Netscape Navigator the cool color change does not occur. You would think with all that money AOL/Time Warner has they could afford to hire better programmers, you would be wrong, but you would most certainly expect it.

Now, let us say that, though you want everyone to know what you think of Disney, you don't want him or her leaving your page. After all, they might become distracted by all the pretty pictures and miss out on the chance to partake of your wit and erudition, so, what we need is a way to open the link in a new window so that your page stays there on their desktop. Quite simple, we use the **target** modifier for the <a> tag.

To have your link open in a new window we add a target= "new" like this;

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice, <a href= "http://www.disney.com" target= "new">Those bastards</a>, I should beat them up.</p>
```

Now, hit refresh and click on the link. See? We have opened the Disney page inside a new window and your audience will not forsake you for the pretty pictures. Now, time for another official *cool HTML trick*<sup>TM</sup>. You have seen those email links on webpages, right? The ones that call up your email program with an email addressed to whomever? Here is how we do it, let us add your email address to the bottom of the page, complete with anchor link, thusly:

```
<a href= "mailto:whatever@wherever.com">email me, you bastard</a>
```

When you hit refresh we see that the text is a clickable link, when we click the link, it brings us a blank email message addressed to the sender. Yes, you can put anything you want in there, the address, an insult, anything.

OK, I can see you digital artists out there are itching to ask me something, go ahead, fire away, "E. H., I see these pages on the web, virtual art galleries as it were, where the designers have a bunch of thumbnail images of their work, and the thumbnails linked to full size versions of the artwork. How do I make links out of my images?"

You haven't even learned how to place an image on the page, and already you want to use them as links. You artists are all so impatient, how about we learn to place them before we make them in to links, OK? Now, to place an image on the page, we use the <img> or *image tag*. The <img> tag must contain an **src** or *source* value, and that value will tell the browser precisely where to find the file in question. Now, as we can no longer avoid it, time for that site architecture lesson. Inside the folder that you created to hold the html file create a new folder (and, if you have been working from your desktop, we will start by creating a folder to hold the HTML file in the first place) named images. This folder will act to store any graphics we create for this project. Inside this folder, place your image file (which I shall call whatever.jpg), and let us return to your webpage. Beneath your poem and explanatory note we are going to place an image, to let everyone in on your sick, pathological obsession with Disney it is an image of Mickey Mouse with a red X through his head. We will add the image like this;

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after
```

receiving my 72nd rejection notice. [Those bastards](http://www.disney.com), I should beat them up.

```
<img src= "graphics/whatever.jpg">
```

Now, when we reload the page, we see that your sick twisted graphic image appears beneath the explanatory note, however, you notice that your graphic is (besides twisted) too large, it overpowers the text, you need to dial it down a bit. Well, for that we have the **height** and **width** attributes. Do I need to explain these to you? Good. We set the values by pixel, so, let us say that we decide we want our image to be 160pixels in height and 140 in width;

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice. <a href= "http://www.disney.com" target= "_blank">Those bastards</a>, I should beat them up.</p>
```

```
<img src= "graphics/whatever.jpg" height= "160" width= "140">
```

Pretty self-explanatory? You can use these attributes in a more sinister way. Going back to the code above, change the width to 350 and hit reload. Finished chuckling? Suitably evil thoughts racing through your mind? OK, now, that image seems to unbalance the page, does it not? So, now we need to align it in such a way that it compliments the rest of the page. For that, we have the handy dandy **align** attribute. Inside of the `<img>` tag the align attribute will accept two values, **left** and **right**. Now, here we get in to some more advanced technique, because there is a second use for the align attribute. First, let us add an align= "right" to the code above like this;

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice, <a href= "http://www.disney.com" target= "_blank">the bastards</a>.</p>
```

```
<img src= "graphics/whatever.jpg" height= "160" width= "140" align= "right">
```

Now, when you reload the page you will see that the image has been aligned to the right hand side of the page, but it still doesn't look right, does it? You want EVERYTHING to the left of the graphic, but you want the graphic flush with the top of your poem, no? Here is the second use for the align attribute and an official "Cool HTML Trick". Take the image tag from above and place it in the FIRST line of your poem, in between the `<p>` tag and the word "Sometimes". That's right, you read that correctly, just like this;

```
<center><font face= "Arial, Helvetica, San Serif">
```

```
<p><img src= "graphics/whatever.jpg" height= "160" width= "140" align= "right">Sometimes I feel my life is a &nbsp;<font size= "4" color= "red"> meteor</font> &nbsp;<br>
```

```
Destined to burn brightly a moment or two<br>
```

```
Ever feeling the cold earth calling<br>
```

```
Knowing I perish with the flame</p></font>
```

```
</center>
```

```
<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice. <a href= "http://www.disney.com" target= "_blank">Those bastards</a>, I should beat them up.</p>
```

Hit reload. See? The image is aligned right, the text wraps to the left. Time to slap a big gold star on your forehead so that everyone knows that you did something cool. OK, so maybe we shan't go that far. Now, we got on to this because some of you impatient artist types wanted to use images as links, correct? Well, it works the same way that I taught you above, we merely surround the image with `<a>` tags. Shall we have a go at it? Splendid. Now, as you are truly a sick and sinister sort, you wish to link your anti Mickey Mouse graphic to the Death to Disney website (you really ought seek help with this obsession of yours). So, we will add anchor tags to the above code like this;

```
<center><font face= "Arial, Helvetica, San Serif">
```

```
<p><a href= "http://www.deathtodisney.com" target= "new"> <img src= "graphics/whatever.jpg" height= "160" width= "140" align= "right"></a>Sometimes I feel my life is a &nbsp;<font size= "4" color= "red">
```

```
meteor</font> &nbsp;<br>
```

```
Destined to burn brightly a moment or two<br>
```

```
Ever feeling the cold earth calling<br>
```

```
Knowing I perish with the flame</p></font>
```

</center>

<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice. <a href= "http://www.disney.com" target= "\_blank">Those bastards</a>, I should beat them up.</p>

Hit reload and have a look. You aren't happy, are you?

"E. H., my brilliant image is surrounded by this tacky looking blue box, what gives?"

Remember how our links looked before we added the <style> tags?

"Yes."

Same thing. The tacky blue border is the browser's way of underlining the image. What we need, is a way to take that tacky border away, and thankfully, we have an attribute to do just that, the **border** attribute which we will now apply to the <img> tag.

"E. H., I thought that style thingie we did was supposed to take care of this problem? Were you lying to us again?"

Nope. As the type= "text/css" modifier indicated, it worked with the text, not with the images. Now, borders are measured in pixels (there will be times you want one, so feel free to experiment), for now, we will set the border to zero like this;

```
<center><font face= "Arial, Helvetica, San Serif">
<p><a href= "http://www.deathtodisney.com" target= "_blank"> <img src= "graphics/whatever.jpg" height= "160"
width= "140" align= "right" border= "0"></a>Sometimes I feel my life is a &nbsp;<font size= "4" color= "red">
meteor</font> &nbsp;<br>
falling<br>
Destined to burn brightly a moment or two<br>
Ever feeling the cold earth calling<br>
Knowing I perish with the flame</p></font>
</center>
```

<p>This is my magnum opus, I wrote it on the eve of my<font color= "#330033"> 30th </font> birthday, after receiving my 72nd rejection notice. <a href= "http://www.disney.com" target= "\_blank">Those bastards</a>, I should beat them up.</p>

Now, when you hit the reload button, you will see your masterpiece, complete with background image and graphics. Congratulations, you are nearly done. All that remains of this first tutorial is our last section, *Lists, Tables & Page Layout Techniques*.

## Lists, Tables and Page Layout Techniques

No doubt you have all read my web page The Wit & Wisdom of E. H. Munro (and if you haven't then you had best do so immediately) and are wondering how I produced those bullet lists. Well, fortunately for you we have arrived at the appropriate section of the tutorial where we will teach you to use lists and tables and how these can be used to achieve page layouts far more precise than we have till now achieved.

First, we shall open a new HTML document, and may I say I am glad to finally be rid of that awful bit of doggerel you have been using till now. I hope you don't mind my observation that it was quite morbid. Now, let us say you fancy yourself the next E. H. Munro, and are looking to produce a page of your original wit. OK, so obviously you are not ever likely to achieve such lofty status, but you are ambitious and aim high. So, you want your own list of your greatest quotes, or perhaps you wish to merely produce a list of your favourite books. Well, in any event, you are looking for a bullet list, well, they are quite easy, to create a bullet list we use the `<ul>` or *Unordered List* tag. The `<ul>` tag is used to create the list, what lies in between the `<ul>` tags will be bulleted. We designate the items to be bulleted with the `<li>` or *Line Item* tag. Let us try it shall we? Let us start with a list of your favourite books. Open a new HTML document, and somewhere in the body of the document place the following code;

These are my favourite books, the ones that I simply can not live without.

```
<ul>
<li>The Compleat Saki- H. H. Munro
<li> The Heart of the Matter- Graham Greene
<li> The Plays of Oscar Wilde- Oscar Wilde
<li> Four Faultless Felons- G. K. Chesterton
<li> Till We Have Faces- C. S. Lewis
<li> The Wicked Pavillion- Dawn Powell
<li> Possession- A. S. Byatt
<li> Everything that Rises Must Converge- Flannery O'Connor
<li> The Jeeves Omnibus- P. G. Wodehouse
<li> The Devil in a Forest- Gene Wolfe
</ul>
```

Now, open the page inside a browser and have a look. You will see that we have created a list of your favourite books, and that the list is indented from the text above it (and may I compliment you on your exceptionally good taste in literature). Now, you will probably want to know what types of bullets you have available to you, so, we will now discuss the **type** attribute for the `<ul>` tag and its possible values; **disc**, **circle** and **square**. The disc value is the default so you will not use it often, mostly you would set the value to square or circle (which produces a hollow circle bullet). Let us try it out in the code above;

```
<ul type= "circle">
<li>The Compleat Saki- H. H. Munro
<li> The Heart of the Matter- Graham Greene
<li> The Plays of Oscar Wilde- Oscar Wilde
<li> Four Faultless Felons- G. K. Chesterton
<li> Till We Have Faces- C. S. Lewis
<li> The Wicked Pavillion- Dawn Powell
<li> Possession- A. S. Byatt
<li> Everything that Rises Must Converge- Flannery O'Connor
<li> The Jeeves Omnibus- P. G. Wodehouse
<li> The Devil in a Forest- Gene Wolfe
</ul>
```

Hit refresh and have a look, now we see that the novels are bulleted by hollow circles rather than the previous solid ones. OK, I can hear you, you think this is all well and good, but it produces only very basic lists. You want to know how to include letters, numbers and combinations thereof for producing content tables for your website. Well, that is a different tag entirely. It is the `<ol>` or *Ordered List* tag. We use this tag to produce sequential lists (numbers, letters, roman numerals). As with the `<ul>` tag, each item in the list is designated with an `<li>` tag. OK, ready to try it for yourself? Open your HTML editor and create a new document, inside our new webpage add the following code:

Top Reasons To Learn HTML

```
<ol>
<li>No more Micro$loth Frontrage
<li>Text editors won't crash your machine
<li>When you hit view source, you'll know what you are looking at
<li>No more Macromedia Screamerweaver
<li>You won't have to worry about any WYSIWYGITA editors inserting glitchy code
<li>You will appreciate the good designs of other people
<li>No more Adobe GoEvil
<li>No more Micro$loth FrontRage
</ol>
```

OK, so we listed FrontRage twice, but it is twice as awful as any other WYSIWYGITA editor, so it deserves the extra mention. Now, have a look at it in a browser window. "OK, E. H., that is all well and good, but what if I want to upload a research paper, I need to put an outline at the front, and I need to have something that looks like a real outline, complete with links."

My, my, aren't we impatient? You haven't even mastered the `<ol>` tag and you want links in your lists. Bloody artists, you're all alike. OK, do you recall the `type` attribute that we added to the `<ul>` tag to produce different types of bulleted lists? Well, we will use the `type` attribute again to change the look of our sequential lists, to go through numbers, letters and roman numerals. The default is numbers, so if you do not assign a type, your list will be sequentially numbered. Below I will give you the values and brief explanations:

```
type= "1" produces numbered lists (1,2,3)
type= "A" produces uppercase letters (A,B,C)
type= "a" produces lowercase letters (a,b,c)
type= "I" produces uppercase roman numerals (I, II, III, IV)
type= "i" produces lowercase roman numerals
```

Now, ready to try it yourself? Let us go back to that page we just wrote add `type= "I"` to the `<ol>` tag above. Voila. Now we see roman numerals in place of the arabic numbers. "Well, E. H., that is all well and good, but I need to mix and match the headings. I did ask about an outline, remember? So, how do I do it?"

OK, now we are going to get into advanced layout concepts here, so you perhaps may wish to brew yourself a strong mug of tea. Liquor is no good as it will dull the wits and lead to typing errors. There is naught worse than a stray `<fpnt>` tag to wreck a perfectly good webpage. Ready? Tea in hand? As I always like to say, I like my tea like I like women, cold and bitter with a splash of bourbon, though in this case we shall forego the bourbon. Now for *nested lists*, the means by which we will create an HTML outline.

Now, we will make a new page, we need a slightly more complex outline than the mere list above. What shall we choose for a topic? The works of Oscar Wilde? Excellent choice. May I commend you again on your tastes in literature. Now, ready for a new document?

The works of Oscar Wilde

```
<ol type= "I">
<li>Short Stories
<ol type= "A">
<li> The Nightengale and the Rose
<li> The Fisherman's Soul
```

```

<li> The Selfish Giant
<li> The Happy Prince
<li> The House of Pomegranates
</ol>
<li> Novels
<ol type= "A">
<li> The Picture of Dorian Gray
</ol>
<li> Plays
<ol type= "A">
<li> A Woman of no Importance
<li> An Ideal Husband
<li> The Importance of Being Earnest
</ol>
</ol>

```

Now, when we look at this in the browser, we see an outline containing selected works of Wilde, with the categories designated with roman numerals and the actual stories being designated by letter. "E. H., what about the links? I told you, I wanted links to each section from the table of contents, how do I do that?"

Bloody ingrates. Here I show you a handy trick which we will reuse in a moment with tables and all you can do is complain, Alright, remember the <a> tag? Of course you do. We used them not twenty minutes ago. OK, what we are going to do is add anchor tags to our list above to link out to other pages. Ready? Excellent, let us go back to the document we just created:

```

The works of Oscar Wilde
<ol type= "I">
<li>Short Stories
<ol type= "A">
<li> <a href= "page1.html">The Nightengale and the Rose</a>
<li><a href= "page2.html">The Fisherman's Soul</a>
<li><a href= "page3.html">The Selfish Giant</a>
<li><a href= "page4.html">The Happy Prince</a>
<li><a href= "page5.html">The House of Pomegranates</a>
</ol>
<li> Novels
<ol type= "A">
<li><a href= "page6.html">The Picture of Dorian Gray</a>
</ol>
<li> Plays
<ol type= "A">
<li><a href= "page7.html">A Woman of no Importance</a>
<li><a href= "page8.html">An Ideal Husband</a>
<li><a href= "page9.html">The Importance of Being Earnest</a>
</ol>
</ol>

```

Now, hit refresh. See? We now have active links for your readers to click. Using the style tags that we created earlier, you can make this Table of Contents look even better. There is one other thing that I feel compelled to point out to you. Up until now I have heard you complain about how much easier layouts are in desktop publishing programs, and even word processing programs such as MS Word, well, here is one place where HTML beats out Word. You will note that HTML lists are far easier to bullet than those in Word, because here we can insert line breaks where ever we please without the software bulleting the new line just because we hit return. So there. Now, I know what is troubling you still, it is that inability to get your text to stay in one place, It just fills the page, even if that page is a 24" monitor set to 1920x1600 resolution, and the text stuck in one long line. You want control, don't you?

Well, we have come to the proper section for that, because now we are about to tackle the subject of tables. They are not quite so ominous as they sound, those of you who spend considerable time doing presentations in Microsoft Word, or other word processing programs will have familiarity with the concept of using tables for layout purposes, we do the same thing in HTML. To insert a table in to a document we use the `<table>` tag. Pretty simple, eh? Well, not quite so simple, but easy enough to grasp. Those of you Windows users who are using NoteTab (one of my highly recommended text editors) no doubt have the auto close feature activated, so that the program will close tags as you enter them (missing `</font>` or `</h2>` tags wreck havoc on a webpage) will now notice an annoying side effect, the moment you insert a `<table>` tag a dialog box will come up asking you all manner of questions about the table's properties, and you haven't a clue as to how to answer them. Turn auto complete off until you understand the `<table>` tag, it will only confuse you otherwise.

Tables have two basic elements, *rows* and *cells*. When constructing tables, it is done row by row, and to start a row we use the `<tr>` or table row tag. Once we have opened a row, we insert the individual table cells, this is done with a `<td>` or *table datum* (the singular of data) tag. Simple enough? Good, let's give it a try. We will build a basic tic tac toe board, that is a table of three rows of three cells each. Open a new HTML document and insert the following code:

```
<table>
```

```
<tr>
<td>x</td>
<td>o</td>
<td>o</td>
</tr>
```

```
<tr>
<td>o</td>
<td>x</td>
<td>o</td>
</tr>
```

```
<tr>
<td>x</td>
<td>x</td>
<td>o</td>
</tr>
```

```
</table>
```

Now, open the document in your browser window, and you see a game that no one has won. "E. H.? I see the Xs and Os, but I don't see the lines." Well, it is time for some table attributes. First, we will use the border modifier, the default value is 0, we will set the value to 2 pixels to start, go back to the document and change the table tag as follows:

```
<table border= "2">
```

Now, when we reload the browser window, we see that our tic tac toe board has a 2 pixel border. "OK, E. H., I see the border, but it looks rather funny, how do I make it square?" I'm glad you asked, we will now use height and width attributes. These attributes can be expressed either absolutely as pixels or relatively as percentages. Let us first square off the tic tac toe board and use absolute (pixel) measurements. Go back to the document and modify the table tag as follows:

```
<table border= "2" width= "150" height= "150">
```

Now, hit refresh and have a look. "Cool, E. H., now, what about that percentage bit? What happens when I set the attributes by percent?" Well, I am glad you asked, if you set the table's height and width by percentage, it will take up exactly that percentage of available page space, let's give it a try, shall we? Let's set the attributes to 25% as follows:

```
<table border= "2" height= "25%" width= "25%">
```

Now when we hit refresh we will see that the table takes up the upper left hand quadrant of the screen. Now, before you pester me with more questions, it is time to learn about the **cellspacing** and **cellpadding** attributes. Cellspacing refers to the distance between individual cells in a table, while cellpadding refers to the space between a cell's content and its border. You probably want to see this in action, eh? Let us start with the cell padding, let us revert our table to an absolute size of 150x150 and add a cellpadding value of 5 as follows:

```
<table border= "2" height= "150" width= "150" cellpadding= "5">
```

When we hit refresh we will notice that our Xs and Os no longer rest against the border, they are now 5 pixels off the edge, that is the cellpadding. Next we will add a cellspacing attribute and set the value to 5 as follows below:

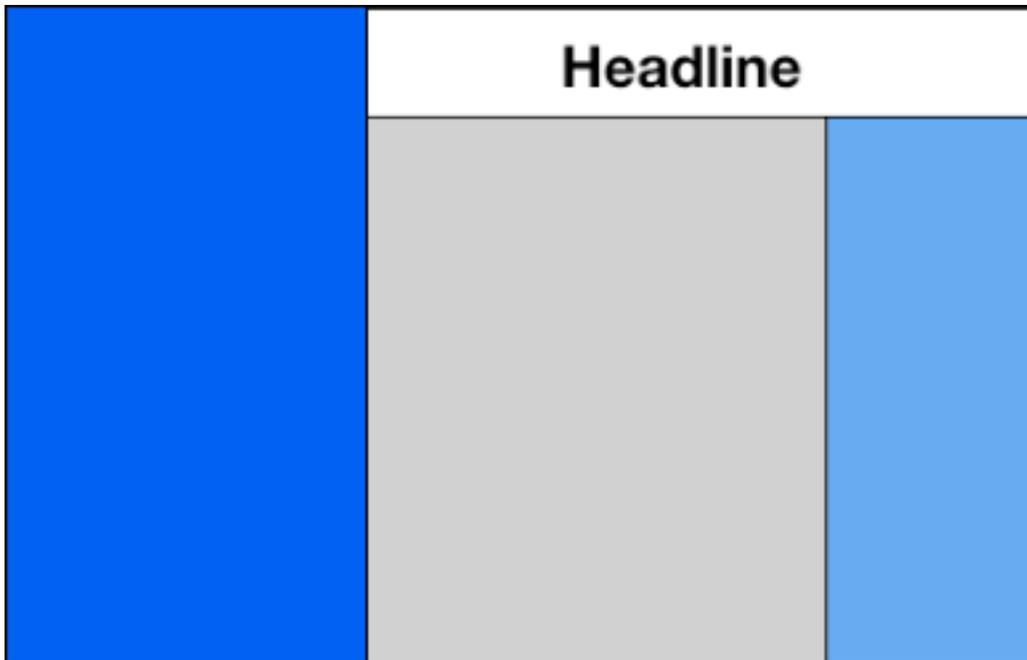
```
<table border= "2" height= "150" width= "150" cellpadding= "5" cellspacing= "5">
```

Now, hit refresh....OK, I can see you want to say something, go ahead. "E. H., I did what you told me to, and the border expanded, but that's it." Well, you see, the borders are on, so when we expand the spacing between cells the border expands to fill the space. When we set the borders to zero, however, you will notice that the Xs Os move apart. Give it a try, set the border to 0 and hit refresh. Satisfied? Excellent. Now, I can see you have some other questions, but we have some more attributes to learn, so you will hold these for a little longer. Now, you might perhaps want a little more control over where things appear inside of table cells, and we have ways of aligning such elements. There are two alignment attributes, **align** and **valign**. The align attribute is for horizontal positioning and has three possible values: **right**, **left**, and **center**. The valign attribute is used for, that's right, you guessed it, vertical positioning and also has three possible values: **top**, **bottom**, and **middle**. Ready to give it a shot? Let us go back to our tic tac toe board and put the Xs and Os in the middle where they belong as follows:

```
<table border= "2" height= "150" width= "150" cellpadding= "5" cellspacing= "5" align= "center" valign= "middle">
```

Now, hit refresh and have a look. See? We have put everything in to the middle of the table cells, where we wanted them to be. When placed in the <table> tag, the align and valign tags will affect all table cells, now, before you ask it, yes, I know that you may wish to have different alignments for different cells, well, thankfully, align and valign can be applied to the <tr> and <td> tags, as well. When you are using a table for layout purposes, you will implement the attributes in the <td> tags. If you are using it for delivering charts, then you will implement them in the <table> tag.

"E. H., I have a problem, this tic tac toe board is neat and all, but how would I lay out a page like that? It seems awfully awkward. Are you sure there aren't any other ways to get better layouts?" Well, we have just started, don't be so impatient. Let us say that you are designing an HTML newsletter, and are looking for a specific layout modeled on the lines of the image below:



The area on the left will be for the picture and we want a headline above with the main story below, the column to the right of the story serving as a spot for our table of contents, quite a challenge, no? Well, for you it is, for myself it is a snap. Let us open a new document, and start with a brand new table, here we will set the cellpadding to 5 to keep the text from running up against the border, and while we're at it, set the border to zero. Let us also use an 800x600 page so that our opening tag looks like this:

```
<table cellpadding= "5" cellspacing= "0" border= "0" height= "600" width= "800">
```

Now, that we have set the page size let us look at the diagram above closely. We can see that there are two rows, one for the headline box and one for what comes beneath, we can also see three columns, one for the graphic image, one for the feature box and headline and one for the Table of Contents. The problem is that the graphic takes up two rows. and the headline two columns. So, how do we manage the trick? Easy, we use the **rowspan** and **colspan** attributes. These attributes are applied to individual table cells, so let us return to our above webpage and start the first row of our new table and add the cells as follows:

```
<table cellpadding= "5" cellspacing= "0" border= "0" height= "600" width= "800">
```

```
<tr>
<td rowspan= "2" width= "25%"><img src= ""></td>
<td colspan= "2" height= "20%" align= "center">headline</td>
</tr>
```

```
<tr>
<td width= "50%" align= "center" valign= "middle">story text</td>
<td width= "25%" align= "left" valign= "top">
<ul>
<li><a href= "story1.html">story 1</a>
<li><a href= "story2.html">story 2</a>
<li><a href= "story3.html">story 3</a>
</ul>
</td>
</tr>
```

```
</table>
```

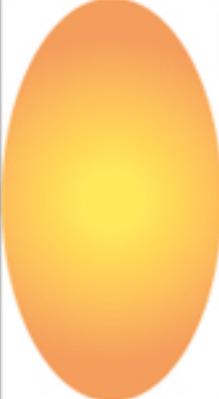
Now, you will notice that we have applied the height and width attributes to the relevant table cells, and did it via percentage, we could have done it just as easily with pixel sizes (as the table's size is fixed at 800x600), the inverse will not work, however, that is if the table's height and width are set by percentage, you should set the cell sizes as such as well. You will note in the second that alignment elements were set in the table cells, as opposed to the <table> tag as we implemented it originally. You will also notice that we have implemented an unordered list in our table of contents cell, illustrating one means of combining lists and tables to achieve better layouts, and that we have added anchor tags to the individual list items in order to achieve easy navigation. OK, with those points noted, open the page in your web browser and have a look.

"E. H., ummm, that image section has a little red X in it." Well, silly, that is probably because you copied the above code without adding an actual image file. Find an image of the approximate dimensions and reference it in the <img> tag and that will solve that problem. Now, we have learned about lists, we have learned how to nest lists for more complex layout, we have seen the use of tables to produce better page layout, now we will take the next step and talk about nesting tables .

Tables produce more accurate layouts, on occasion, when you get in to quite advanced layouts, you will find the need for nesting one table inside of another. "E. H., that sounds awfully tough, is there really a good reason for doing it?" Yes. Our above email newsletter was set to the dimensions of 800x600, but not everyone uses such dimensions for their desktop. It is common enough on laptops, but with desktop systems people use 17", 19" and even 21" monitors on a regular basis, these people will find 800x600 pages lost on their desktop. WebTV users won't be very happy either (though, in reality, someone with a philistine streak that wide deserves every ounce of misery they get) as the dimensions won't jibe with what they can see. So, how do we create a layout that will work across screen sizes? Well, we can start by creating a master table to hold the entire page, regardless of window size. So, let us start a new document, and in between the <body> tags insert the following code:

```
<table cellpadding= "0" cellspacing "0" border= "2" height= "100%" width= "100%">  
  
<tr>  
<td>  
  
</td>  
</tr>  
  
</table>
```

Now, we have turned the borders on only so that we can see the table on the page until we add extra elements, and for the same reason put in only a single cell for now. If we view this in a browser window, you will see that the table fills the window size, even when we resize the window. This will be our master table, this will ensure that our layout adheres to page size, no matter what that page size might be. Now, let us envision a page layout complex enough to pose a design challenge for you, eh? Let us refer to the figure below:

	<h2>Rotating Banner</h2>	
	<h3>Feature Story</h3> <p>Here is where your feature story will reside A place for the stuff you want everyone to see first and foremost</p>	<p>Feature 1 Feature 2 Feature 3</p>
<p><b>Story 1</b> Here is the first story about dogs and cheese. Is there a connection?</p> <p><b>Story 2</b> Here is the second story. Should we stuff kids in to lockers</p>	<p><b>Story 3</b> Here is the text for the third story. Should we drive solar powered bicycles.</p> <p><b>Story 4</b> Text goes here, or else.</p>	<p><b>Story 5</b> Put the text here, because I said so, that's why</p> <p><b>Story 6</b> Here is where, OK, I will presume by now that you get the point</p>

Now, here we will (in theory) have a headline banner running across the top row, we have two picture boxes, the first starts with the top cell row one and has a rowspan of two, but we also have a complex navigational element below, a three column set of story links with captions, the other two elements, the primary feature box and the regular feature links can be coded with what we have learned to date. So, really, what we need is a way to create the lower section layout, and for that we will use a nested table, and, as we are working within a master table anyway, it is a table, nested in a table that is nested in a table. Ready for the challenge? Excellent. First, let us go back to our master table and shut the border off as follows:

```
<table cellpadding= "0" cellspacing "0" border= "0" height= "100%" width= "100%">
```

Now that our master table is invisible, we will place the new table inside the master table's one cell as follows:

```
<table cellpadding= "0" cellspacing "0" border= "0" height= "100%" width= "100%">
```

```
<tr>
<td>
```

```
<table border= "0" cellpadding= "5" cellspacing= "0" height= "80%" width= "80%">
```

```
</table>
```

```
</td>
</tr>
```

```
</table>
```

Now, you will notice that I have closed the table tag immediately, and you will always follow my example when using nested tables, lists, etc... Missing end tags will ruin a page, and nested tags are a nightmare to check. So, to avoid the problem, when nesting, you will close your tags directly after you insert them. Now, the nested table has been given height and width attributes of 80% because I felt like it, I am the designer so it is my choice...so there. Now, let us turn back to our first nested table and start coding, shall we? Now, as we can plainly see, our first row contains two cells with the second cell (the rotating banner) spanning two columns, and the first cell spanning two rows. The second row has two normal cells, though the width on the feature story cell should be set to 50%, the third row is

going to be another nested table, so it will have a single table cell spanning three columns. Now, let us go back to the document and code the table as follows:

```
<table border= "0" cellpadding= "5" cellspacing= "0" height= "80%" width= "80%">

<tr>

<td width= "25%" rowspan= "2" align= "left" valign= "middle">

</td>

<td height= "20%" colspan= "2" align= "center" valign= "middle">
</td>

</tr>

<tr>

<td width= "50%" height= "40%" align= "center" valign= "middle"
<h1> Feature Story</h1><br>Text Goes Here
</td>

<td width= "25%" align= "left" valign= "top">
<h3>Feature 1</h3><br>
<h3>Feature 2</h3><br>
<h3>Feature 3</h3><br>

</tr>

<tr>

<td height= "40%" colspan= "3">

<table cellpadding= "5" cellspacing= "5" border= "0" align= "left" valign= "top">

</table>

</td>

</tr>

</table>
```

Now, you will please note that I opened and closed the tags for our next nested table in the third row for convenience. When we view this in our browser we will see everything except the story blurbs reserved for the bottom cell. Now, time to tackle the last nested table, we could do it two different ways, two rows of three cells each is the most straightforward, and though it will involve the writing of extra code, that is how we will do it. I hope that you noted above that when I inserted the code that I placed the alignment elements within the <table> tag rather than the <td> tags, because as you can see, I want the same alignment for every cell, why write extra code? Let us back to our HTML document and code the third table as follows:

```
<table cellpadding= "5" cellspacing= "5" border= "0" align= "left" valign= "top">

<tr>
```

```
<td width= "33%">
<a href= "story1.html"><h2>Story 1</h2></a><br>
This is our first story, <em>Dogs and Cheese: What is the Hidden Connection</em>?
</td>

<td width= "33%">
<a href= "story3.html"><h2>Story 3</h2></a><br>
This is where we place the text for story three, <em>Should we drive solar powered bicycles</em>?
</td>

<td width= "33%">
<a href= "story5.html"><h2>Story 5</h2></a><br>
Put the text here. Because I said so, that's why.
</td>

</tr>

<tr>

<td width= "33%">
<a href= "story2.html"><h2>Story 2</h2></a><br>
Here is where we put the text for the second story, <em>Should we stuff kids in to lockers</em>?
</td>

<td width= "33%">
<a href= "story4.html"><h2>Story 4</h2></a><br>
Put the text here, or else...
</td>

<td width= "33%">
<a href= "story6.html"><h2>Story 6</h2></a><br>
Here is where...OK, I shall presume that by now you have taken the point.
</td>

</tr>

</table>
```

Now, when we open the newly saved document in the browser window we see that our page looks like the picture above (more or less) and you have successfully nested tables within one another, and have learned a valuable technique for composing advanced page layouts. Now, for those of you graphic artists reading this, what I say next will make sense, the rest of you will not understand the utility until you get in the habit of doing it yourself. Before actually coding your pages, draw them out on paper. Sketch your rough idea for what you want the page to look like on paper, when it is on paper in front of you, it is far easier to write the code that will comprise the webpage. If you intend to do this seriously, as either a vocation or a hobby, buy yourself a drawing tablet and a mechanical pencil, because you will use them both.

As a last point, I must restate the importance of neat code. Put in extra line breaks between elements, note that I always double space between tags, it makes it easier for others (and yourself) to read the code. The former principle was that HTML code should always be written in uppercase letters ( <BR> as opposed to <br> ) to make the code easier to read, but you, my students, will not do that, you will include extra spaces to achieve the same effect (easier to read code) because in the future, uppercase code will be a no no (see part four XHTML and Beyond). Secondly, as I stated earlier, always open and close tags in the proper order, referencing the nested table directly above, you will note that the anchor tags open before the headline tags and close after the closing of the headline tags. First opened, last closed. Also, the closing anchor tags precede (as they ought) the line break tags. It is the sloppy lines of code that cause Netscape...errrr....Netscape to misdisplay your pages. So, we will write clean code. Always. Got it? Well, why don't you try designing a couple of original webpages with what you have learned here, and when you are ready,

we will move on to the next section, *Advanced HTML*.

© 2001 by E. H. Munro. All rights reserved. No portion of this may be reprinted without expressed consent of the author.