

McCartney's Journeys in Genealogy

# HTML Tables

---

A Basic Understanding

*George McCartney*

1 October 2010

# Table of Contents

<a href="#">Preface</a>	2
<a href="#">Laying Out the Table</a>	3
<a href="#">Coding the Table</a>	4
<a href="#">Two Important Attributes</a>	5
<a href="#">The Quiz</a>	7
<a href="#">The Answer</a>	8
<a href="#">Table Attributes</a>	9
<a href="#">background</a>	10
<a href="#">bgcolor</a>	12
<a href="#">border</a>	13
<a href="#">bordercolor</a>	14
<a href="#">cellpadding</a>	15
<a href="#">cellspacing</a>	15
<a href="#">colspan</a>	16
<a href="#">rowspan</a>	18
<a href="#">height</a>	21
<a href="#">width</a>	22
<a href="#">Conclusion</a>	23
<a href="#">Problem Solutions</a>	
<a href="#">Practice Table 1</a>	24
<a href="#">Practice Table 2</a>	25
<a href="#">Practice Table 3</a>	26

# Preface

This document is intended to be a very basic explanation of certain aspects of coding HTML tables. It is not intended to be a complete treatise. There are many excellent books which cover the entire HTML coding structure and all of its nuances.

## Recommended Books

Creating web pages encompasses many areas that are too involved to be included in this document. I highly recommend the book, *HTML & XHTML; The Definitive Guide*, by Chuck Musciano & Bill Kennedy. It is available from O'Reilly & Associates, Inc. Their website is <http://www.oreilly.com>. The book is somewhat technical, but it is very readable and understandable.

For a basic introduction to web design and coding, try "Web Site Design Made Easy: Learn HTML, XHTML, and Css" by Dennis Gaskill. Dennis has a very relaxed writing style but can explain things from a beginner's standpoint. If you don't know anything or know very little about web site creation and writing, this book is for you. The book is available through Amazon.com. Dennis has a website with many helpful articles and tips. Visit <http://boggiejack.com>.

No, I don't receive anything from recommending these books. They are books I have on my bookshelf, refer to constantly and have recommended to my friends.

## Legal Comments

© 1 October 2010, by George McCartney, Staten Island, NY, USA.

The content of this ebook was written for educational purposes and is based on experience and techniques the author used when developing the [McCartney's Journeys in Genealogy](#) website. Both the ebook and the website are protected by copyright. You may download the ebook for your personal use. You may also offer this ebook as a free download from your personal website provided no charges or special requirements are attached. It cannot be changed or modified in any way and it may only be distributed in the PDF format. It may not be distributed in any other format, whether electronic, printed or any other media format.

In using the information presented within, the reader assumes all responsibility, and acknowledges that the author and any and all distributors of this ebook shall be held blameless for losses of any kind, direct or indirect, real or imagined, tangible or intangible, at any time.

# Understanding HTML Tables

Some data is best presented in a tabular form: columns of numbers would be one example. Tables are the best way of doing this in HTML. Tables have also been used to lay out web pages, although this is better done with Cascading Style Sheets (CSS).

## Laying Out the Table

What confuses most people about tables is trying to visualize a column/row context in a vertical coding. Each element of a table is coded on a separate line, but the elements are side by side on the displayed web page. It might help to visualize the table in a diagram which shows the elements coded in a way representative of what you want for your page. Physically drawing the table on a piece of paper will allow you to place the data where you want it.

Let's say we want to show the letters A through I in a table, with A, B and C on one line, D, E and F on the next, and G, H and I on the third line. Our diagram would look like this:

<b>A</b>	<b>B</b>	<b>C</b>
<b>D</b>	<b>E</b>	<b>F</b>
<b>G</b>	<b>H</b>	<b>I</b>

In HTML the parts of the table have specific names. Each box is a cell. Vertical groups of cells are called columns and horizontal groups of cells are called rows. So, A, D and G are in column 1; B, E and H are in column 2, and C, F and I are in column 3. While we would say "Put A, B and C on one line", HTML would say "Put them in one row".

The entire table is coded between two tags which define it: `<table>` and `</table>`. The first, `<table>`, defines the start of the table. Like most other elements in HTML, the table must be closed. The `</table>` does this for us. So let's define the start and end of the table on our diagram like this:

`<table>`

<b>A</b>	<b>B</b>	<b>C</b>
<b>D</b>	<b>E</b>	<b>F</b>
<b>G</b>	<b>H</b>	<b>I</b>

`</table>`

To define a row, we use the table row command `<tr>` and its closing command `</tr>`. So, let's define the rows in our diagram:

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>D</td>
    <td>E</td>
    <td>F</td>
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
    <td>I</td>
  </tr>
</table>
```

A	B	C
D	E	F
G	H	I

The letters A through I are each in a cell of the table. The cells are where the data you want to present are placed. To define cells we use the `<td>` and `</td>` commands. We will put the `<td>` and `</td>` commands in each cell. Our diagram now looks like this:

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>D</td>
    <td>E</td>
    <td>F</td>
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
    <td>I</td>
  </tr>
</table>
```

<td>A</td>	<td>B</td>	<td>C</td>
<td>D</td>	<td>E</td>	<td>F</td>
<td>G</td>	<td>H</td>	<td>I</td>

## Coding the Table

Now let's get to the HTML coding of the table. It is possible to code all of the elements on one line, but it becomes very difficult to read later. Therefore, it is good practice to code each element of the table on a single line

Our table would be coded as:

It would appear on the web page as:

```
<table width="50%"
  border="3">
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>D</td>
    <td>E</td>
    <td>F</td>
  </tr>
  <tr>
    <td>G</td>
    <td>H</td>
    <td>I</td>
  </tr>
</table>
```

A	B	C
D	E	F
G	H	I

There are two attributes listed in the <table> command that need brief explanations. If no width is specified, the browser will make the table the width of the screen. I have added a width of 50% so the table will only take half of your screen. Widths can be stated as percentages or as pixels. If you only want the table to be 150 pixels wide, the attribute would be width="150". The second attribute is borders. Not having borders is the default. If you want borders, you must specify them. More detailed explanations are available in the article about table attributes.

That is a basic table design. There are two attributes available for designing tables that we will look at next.

## Two Important Attributes

As most commands in HTML have attributes, so too do tables, rows and cells. There are a number of different attributes available for these commands. We saw two, width and borders, in the previous discussion. For this discussion we are only concerned with two cell attributes: cell depth and cell width. First, we will deal with depth.

In the previous discussion, our table was uniform. It had the same number of cells in each row and the cells were all the same size. There will be times when this is disadvantageous. We may need one or two cells to take more space than others.

Here's a small problem. What if you want to put a picture in the first cell of the top row? If the picture is the same size as the cell, there is no problem, simply add an <img> command between the <td> and </td>. For this example, let's assume that the picture is the width of a cell, but two cells deep. Simply putting the picture in the first cell will change the height of all the cells in that row. We don't want to do that. Note, for this example we are not actually using a picture, but two words, Big and Picture separated by a line command <br />. Again, we diagram our table:

<b>Big</b>	<b>T</b>	<b>U</b>
<b>Picture</b>	<b>V</b>	<b>W</b>
<b>X</b>	<b>Y</b>	<b>Z</b>

This time we won't go through each step as we did before. We will go directly to the code, which is:

```
<table width="50%"
border="3">
  <tr>
    <td rowspan="2"
      width="33%">
      Big<br />Picture
    </td>
    <td width="34%">T</td>
    <td width="33%">U</td>
```

And it would appear on the web page as:

<b>Big</b>	<b>T</b>	<b>U</b>
<b>Picture</b>	<b>V</b>	<b>W</b>
<b>X</b>	<b>Y</b>	<b>Z</b>

```

</tr>
<tr>
  <!-- <td> </td> this cell is not
  defined. Its position will be taken by
  the depth of cell 1, row 1 -->
  <td>V</td>
  <td>W</td>
</tr>
<tr>
  <td>X</td>
  <td>Y</td>
  <td>Z</td>
</tr>
</table>

```

Notice that the <td> elements on row 1 have width attributes added. These are used to force each cell and, therefore all cells in the respective columns to be the same width.

Now let's explain of the differences in this code. For convenience they are in red.

In the first cell of the first row, we need to define the fact that its content will exceed the cell depth. Note the rowspan="2". This tells us that the picture will take up cell 1 in row 1 and cell 1 in row 2. The content could be anything. For this example we put Big on one line and Picture on another by using the <br /> tag. Notice also that we lose the first cell in row 2. It is no longer available because cell 1 in row 1 now occupies that space.

In row 2, cell one is commented out (placed between the HTML comment command <!-- and its close -->). It is not necessary to put this in the code, but it makes it easier to read when you have a lot of cell juggling in your table. By showing the cell commented out, every row will have the same number of <td> and </td> commands, even though some are not used.

Now, let's look at the second attribute: width. Just like depth, width is stated in the <td> command. We will assume the picture is the same depth as a cell, but is two cells wide. Our diagram is:

<b>Long Picture</b>		<b>T</b>
<b>U</b>	<b>V</b>	<b>W</b>
<b>X</b>	<b>Y</b>	<b>Z</b>

The code for this is:

```
<table width="50%"
border="3">
<tr>
<td colspan="2">
  Long Picture</td>
<!-- <td> </td> this cell is not
defined. Its position will be taken by
the width of cell 1, row 1 -->
<td>T</td>
</tr>
<tr>
<td width="33%">U</td>
<td width="34%">V</td>
<td width="33%">W</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</table>
```

And it would appear on the web page as:

Long	Picture	T
U	V	W
X	Y	Z

Notice the change in cell 1, row 1. Instead of a rowspan attribute, there is a colspan attribute and the second <td> is now commented out. In row 2 all <td> elements are coded. Notice also that the width attributes have been moved to the second row. Row 2 is the first row to have all three cells defined. Row 1 only defines two cells.

These two pages are a very basic explanation of tables. We have briefly touched on setting widths. There are ways to set the width of the entire table, change the borders of cells or remove them completely and many other features. We will cover the major attributes and features on page 10.

## The Quiz

Now it's your turn. Try to code a table that looks like this. Try coding it before you look at the answer.

<b>Really Big Long Picture</b>		V
		W
X	Y	Z

The answer is on the next page. Try to code this before you look at the answer.

## The Answer

Did you match this diagram?

<b>Really Big Long Picture</b>		<b>V</b>
		<b>W</b>
<b>X</b>	<b>Y</b>	<b>Z</b>

The code for this is:

```
<table width="50%"  
border="3">  
<tr>  
  <td colspan="2" rowspan="2">  
    Really Big<br />Long Picture  
  </td>  
  <!-- <td> </td>  
    this cell is not needed. Its  
    position will be taken by the  
    width of cell 1, row 1 -->  
  <td>V</td>  
</tr>  
<tr>  
  <!-- <td></td>  
    <td></td> these cells are not  
    needed. Their positions will be  
    taken by the width and depth of  
    cell 1, row 1 -->  
  <td>W</td>  
</tr>  
<tr>  
  <td width="33%">X</td>  
  <td width="33%">Y</td>  
  <td width="33%">Z</td>  
</tr>  
</table>
```

And it would appear on the web page as:

<b>Really</b>	<b>Big</b>	<b>V</b>
<b>Long</b>	<b>Picture</b>	<b>W</b>
<b>X</b>	<b>Y</b>	<b>Z</b>

Note that cell 1 of row 1 now has the colspan and rowspan attributes and cell 2 of row 1 is commented out. In row 2 both cells 1 and 2 are commented out since cell 1 of row 1 will occupy those cells. Also, notice that the only row that defines all three cells is row 3. That is where the width attributes are placed.

Were you successful? If not, don't feel bad. Everyone has problems with this in the beginning. Study the answer code and try again. After you get it right, try a few of the diagrams below. Practice coding them. Be careful, some of them are tricky.

**Practice Table 1**


**Practice Table 2**


**Practice Table 3**


**TIP:** to leave a cell empty, you must use the named entity `&nbsp;` in the cell definition for at least one cell in the row. If you just leave the cells empty, the browser will use the default height of 2 pixels, and it will look like this:



**TIP 2:** Numbering the rows in Practice Table 3 will help to know where you are when you code the table

The answers to the practice tables are in the [Conclusion](#)

# Table Attributes

As we saw in our previous discussion, there are four major elements in tables: the table itself, rows, columns and cells. We also saw that attributes could be used to format tables, rows and cells. Since columns are made up of stacks of cells (<td>'s), there is no column command, therefore no attributes apply directly to columns. Only tables, rows and cells have attributes.

In HTML, there are 35 attributes for tables, 25 for rows and 34 for cells. Many attributes are common between several elements. We will be covering the ten most used attributes as shown in the following table with the elements that use them.

	Attribute	Table <table>	Row <tr>	Cell <td>
1	<a href="#">background</a>	YES	no	YES
2	<a href="#">bgcolor</a>	YES	YES	YES
3	<a href="#">border</a>	YES	no	no
4	<a href="#">bordercolor</a>	YES	YES	YES
5	<a href="#">cellpadding</a>	YES	no	no
6	<a href="#">cellspacing</a>	YES	no	no
7	<a href="#">colspan</a>	no	no	YES
8	<a href="#">rowspan</a>	no	no	YES
9	<a href="#">height</a>	YES	No	YES
10	<a href="#">width</a>	YES	no	YES

Just click on the name of an attribute to see an explanation of how to use it.

## background

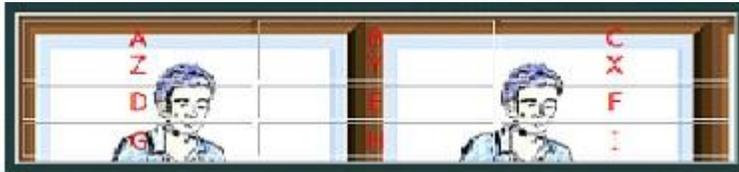
Availability: tables, cells.

By default the background will be the same as the page background. This attribute allows you to use an image for the background of the entire table or just a cell.

If the attribute is used with the <table> command, the image specified will appear behind the entire table. The size of the table will determine how much of the image will be visible. If the table is smaller than the image, the image will be truncated. If the table is larger than the image, the browser will attempt to tile the image. In the following examples the borders have been left in to show the positioning of the image.



The table is the same width as the image, but not as long. The bottom of the image is truncated.

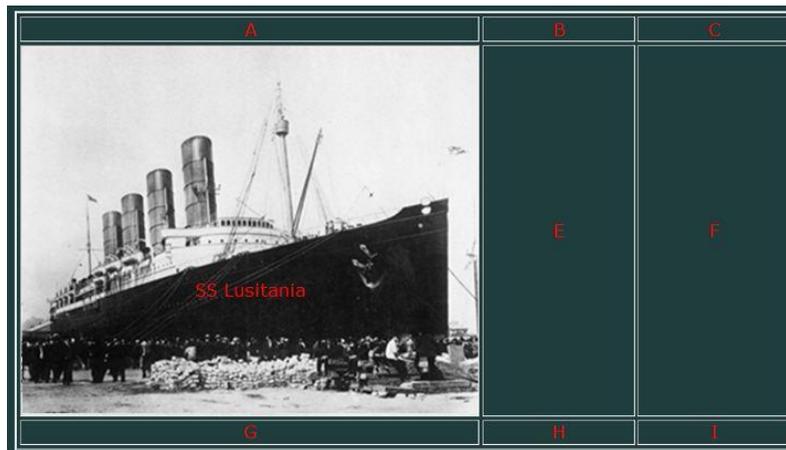


The image was tiled because the table is wider.

The coding would be: `<table background="URL of the image">`

Using images as a background for tables isn't always a good choice. Placing text over the image usually becomes too confused as can be seen in the two examples above.

Using an image in a cell can help to create an interesting table. There are two ways to do this. First, you can just add a `<img>` command to the cell, but this would not allow text content to appear over the image. Using the background attribute will allow both image and text to occupy the same space. There is a caveat, however; you must specify height and width attributes for the `<td>` element. If the textual content is smaller than the image, the image will be truncated.



In the table above, the image is rather large, so height and width attributes were added to the cell. Also, to get the title "SS Lusitania" into the dark area so it could be seen, several `<br />` commands were used. They have nothing to do with the table setup, but illustrate the need for some customizing in this type of situation. The code for the row is:

`<tr>`

```

<td background="URL of lusitania.jpg"
  height="267"
  width="335"><br /><br /><br />
  <br /><br /><br />SS Lusitania</td>
<td>E</td>
<td>F</td>
</tr>

```

Height and width will be discussed in more detail later. Notice what setting the height of the first cell did to the rest of the cells in the row. This may not be a problem if your content for each cell is extensive enough.

Background is not supported by most browsers. It is unique to Internet Explorer and NetScape/FireFox. The manner of presentation varies between these two browsers as well. This is another reason to consider other ways to present your data. I have not experimented with it, but I have read that NetScape will render the image in a totally unrecognizable way. Larisa Thomason gives an explanation in her article:

**“HTML Tip: Avoid Table Background Image Bugs”.**  
[http://www.netmechanic.com/news/vol3/html\\_no6.htm](http://www.netmechanic.com/news/vol3/html_no6.htm).

## bgcolor

Availability: tables, rows, cells.

Changing the background color of a table, row or cell is easy. Simply include this in the element code: `bgcolor="color of your choice"`. The value of the color can be the standard name or its hexadecimal value. For example, the color RED may be used. Or you could use its hex value #FF0000. There are tables of color names and hex values available on the WWW. One good one, showing over 500 colors is [cloford.com \(http://cloford.com/resources/colours/500col.htm\)](http://cloford.com/resources/colours/500col.htm).

The following examples show the tables and the coding;

A	B	C
D	E	F
G	H	I

The entire table:

```

<table
bgcolor="blue"...

```

A	B	C
D	E	F
G	H	I

A row:

```

<tr
bgcolor="#0000ff"
...

```

A	B	C
D	E	F
G	H	I

A single cell:

`<td bgcolor="blue">`

Notice the use of the hex value in the row example: the name 'blue' and the hex value '#0000ff' are the same color.

It is possible to use one color for the table background and several different colors for rows and/or cells. Of course, you have to be careful of the color choices to be sure they don't clash.

Have fun coloring your tables.

## border

Availability: tables.

Tables are usually displayed by the browsers without borders. That is the default. If you want borders they must be specified. When coded, the borders will be beveled as is seen in the examples below. The border can be made as thick as you want. The thicker they are the more the beveled affect can be seen.

To specify borders, the code is: `<table border="integer">`. Integer represents pixels and can be any number you decide. The top table in the example below has `border="2"` and the bottom has `border="8"`.

A	B	C
A	B	C

A	B	C
A	B	C

As usual there is a caveat: Internet Explorer will display the borders in white, but NetScape/FireFox will use the `color=` attribute which specifies the color of the text, so the text and borders will be the same color. You will notice in the examples below that there is a slight difference in size as well. This is part of the vagaries between the two major browsers.

A	B
D	E

The table as displayed in Internet Explorer.

A	B
D	E

The same table displayed in NetScape/FireFox.

## bordercolor

Availability: tables, rows, cells.

The border's color can be changed with the bordercolor attribute. It does, however, change the appearance as well; the beveled affect is lost. The code for the example below is:

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

A	B
D	E

Unfortunately, it only works part way in NetScape/FireFox. The table border gets changed completely, but the cell borders remain red, as shown below.

A	B
D	E

In the following example, bordercolor has been applied to the table (white), the top row (limegreen) and the second cell in row 2 (gold). Internet Explorer displays the table as:

A	B
D	E

NetScape/FireFox still displays it as:

A	B
D	E

Internet Explorer offers two non-standard variations of bordercolor: bordercolorlight and bordercolordark. These attributes will lighten or darken the border by approximately 25%. They will not be recognized by any other browser, so no change to the border color will occur in the others.

So that is bordercolor. Now you can have fun coloring your borders too.

## cellpadding

Availability: tables.

The space between the cell's border and its content is, by default, 1 pixel. Normally, this is adequate, but there are times when more space is desired. The attribute cellpadding does this. Its value is a number which the browsers automatically read as the number of pixels to set the padding. It is possible to set the attribute value to zero (0). This will remove the 1 pixel default.

When the attribute is set, every cell will receive the same padding amount.

The coding would be: `<table cellpadding="some integer">`. An example of cellpadding is shown under the cellspacing attribute below.

## cellspacing

Availability: tables.

While cellpadding takes place within the cell, cellspacing occurs between cells. The default here is normally 2 pixels. If borders are used it will be 2 pixels greater than the border thickness. For example, if you specify `border="3"` and `cellspacing="5"` the total space between borders will be 7 pixels.

Cellpadding and cellspacing can be combined in the same table. The following examples show a table with default padding and spacing and the same table with padding and spacing set to 8 pixels.

This is all cell data. This is all cell data. This is all cell data. This is all cell data.	More data.
More data.	More data.

This table element was coded:

```
<table border="5"
      style="width:
298px; ">
```

This is all cell data.	More Data.
More data.	More data.

This table element was coded:

```
<table border="5"
      cellpadding="8"
      cellspacing="8"
      style="width:
298px; ">
```

The overall size of the tables was controlled by the style attribute which is part of Cascading Style Sheets and will not be explained here. That is a totally different subject and one too involved for this article.

In the second table you will notice the added space between the cell borders (cellspacing="8"). The default spacing of one pixel has now been increased. Notice also that the padding around the text has also been increased (cellpadding="8"). This has caused the text lines to wrap in the first cell of row 1. The affects of padding can be seen more easily in row 2 where the data is small enough not to wrap.

### colspan

Availability: cells.

You have encountered this attribute when you read the article "A Basic Understanding of HTML Tables" where it was briefly explained.

There are times when you want the data to extend beyond its cell into the cell or cells adjacent to the right. This is done by specifying the number of columns you want the cell to expand. In the table below a long title is used needed to define the contents of two columns. The colspan attribute defines this.

Employee Name		Date Hired
John	Smith	June 3, 1980
Mary	Jones	May 5, 2007

The first row only contains two cells while the rest of the rows have three. This is accomplished by the following code:

```
<table>
  <tr>
```

```

        <td colspan="2">Employee Name</td>
        <td>Date Hired</td>
    </tr>
    <tr>
        <td>John</td>
        <td>Smith</td>
        <td>June 3, 1980</td>
    </tr>
    <tr>
        <td>Mary</td>
        <td>Jones</td>
        <td>May 5, 2007</td>
    </tr>
</table>

```

Notice that row 1 only defines two cells (tds). This is because the first cell is expanded to fill what would normally have been the second cell.

If the colspan integer is too large - more than the number of cells in the row - the browser will not add more columns. It will expand the cell over the columns that do exist. If, as in our example, another td follows the first one, it will be shown with no cells under it. In the example below the first row was defined as:

```

<tr>
    <td colspan="2">Employee Name</td>
    <td>Date Hired</td>
</tr>

```

This is the result:

Employee Name		Date Hired
John	Smith	June 3, 1980
Mary	Jones	May 5, 2007

When used properly (without any typos) colspan can be a very effective way to present data.

## rowspan

Availability: cells.

Rowspan works in a very similar way as colspan, except that it goes vertically instead of horizontally. Unlike colspan, extending beyond the existing rows has no affect on the table appearance. The over-extension is ignored by the browser.

In the table below, we have added a column at the beginning which extends down the rows and contains the company name. The row is coded as:

```

<tr>
<td rowspan="4">
  T<br />
  H<br />
  E<br />
  &nbsp;<br />
  S<br />
  T<br />
  O<br />
  R<br />
  E<br />
</td>
<td colspan="2"
  class="center">
  Employee Name
</td>
<td>Date Hired</td>
</tr>

```

The result is shown on the next page.

T H E  S T O R E	Employee Name		Date Hired
	John	Smith	June 3, 1980
	Mary	Jones	May 5, 2007

In the previous chapter it was mentioned that colspan and rowspan can be used in the same table and in the same cell. The entire code for the table above is:

```

<table border="3"
  style="width: 298px;
  margin-left: 35%;">
<tr>                                <!-- Row 1 -->
<td rowspan="4">
  T<br />
  H<br />
  E<br />
  &nbsp;<br />
  S<br />

```

```

        T<br />
        O<br />
        R<br />
        E<br />
    </td>
    <td colspan="2"
        class="center">
        Employee Name
    </td>
    <td>Date Hired</td>
</tr>
<tr>                                <!-- Row 2 -->
    <td>John</td>
    <td>Smith</td>
    <td>June 3, 1980</td>
</tr>
<tr>                                <!-- Row 3 -->
    <td>Mary</td>
    <td>Jones</td>
    <td>May 5, 2007</td>
</tr>
</table>

```

All of the rows define three cells. Yet, looking at the displayed table it is obvious that rows 2 and 3 have 4 cells each. Where are they defined in the code? This is why I recommend putting all of the cell definitions in the code and commenting out the ones which are replaced by the colspan and rowspan attributes. Putting notes as to why the cells are commented out also helps future understanding. It may seem like extra work now, but believe me, when you look at a big table that you coded months before, it will help you to understand what you did and make changes easier. The recommended way to code this table is shown on the next page. The changes are in red.

```

<table border="3"
    style="width: 298px;
    margin-left: 35%;">
    <tr>                                <!-- Row 1 -->
    <td rowspan="3">
        T<br />
        H<br />
        E<br />
        &nbsp;<br />
        S<br />
        T<br />
        O<br />
        R<br />
        E<br />

```

```

</td>
<td colspan="2"
    class="center">
    Employee Name
</td>
<!-- <td> </td> this cell is not needed. Its position is taken by the
        preceding cell. -->
<td>Date Hired</td>
</tr>
<tr>                                <!-- Row 2 -->
<!-- <td> </td> this cell is not needed. Its position is taken by cell 1 of
        row 1 -->
<td>John</td>
<td>Smith</td>
<td>June 3, 1980</td>
</tr>
<tr>                                <!-- Row 3 -->
<!-- <td> </td> this cell is not needed. Its position is taken by cell 1 of
        row 1 -->
<td>Mary</td>
<td>Jones</td>
<td>May 5, 2007</td>
</tr>
</table>

```

Diagramming your table on a piece of paper should simplify the coding because you will see exactly what you need to do in order to accomplish what you want.

## height

Availability: tables, cells.

This attribute was briefly mentioned in the discussion of background images. You can control the height of a cell, and therefore a row, or the entire table by using `height=` in your code. When used the browser will adjust the height of the element, either table or cell, to the specified amount. The amount is an integer representing pixels.

The table below had the first cell of row 2 defined with a height of 50 pixels. Notice the space above and below the text. Row 3 uses the default height.

Employee Name		Date Hired
John	Smith	June 3, 1980
Mary	Jones	May 5, 2007

The height adjustment will work the way you want it to, provided the content is shorter than the specified height. If the content is larger it will not be truncated, but the browser will set it to accommodate the content.

The next example has a bit more text in the first cell. It was coded to force each word onto a new line so that we could see how the browser affects the height.

Employee Name		Date Hired
John has been a valued employee for many years.	Smith	June 3, 1980
Mary	Jones	May 5, 2007

The height attribute is still set to 50 pixels, but the data required more so the browser ignored the attribute.

## width

Availability: tables, cells.

This attribute was also discussed briefly under background images. It allows you to set the width of a table or cell. Like height, the width is specified in pixels, but it can also be stated in percentages.

The following shows the use of the width attribute for both the table and cells.

```
<table width="75%">
  <tr>
    <td width="25%">A</td>
    <td width="50%">B</td>
    <td width="25%">C</td>
  </tr>
</table>
```

Some explanation is necessary. The width assigned to the table will make it 75% of the data display area. The widths applied to the first and third cells will make each of them 25% of the table or, if you do the math, 18.75% of the data area. Cell 2 will be 50% of the table or 37.5% of the data area.

Where the browser will adjust the height if the content is larger than the attribute specifies, it will not do so for the width. If you put an image in a cell and specify a width that is narrower than the width of the image, the image will be truncated.

As was mentioned earlier, there are many more attributes available for tables, rows and cells. Those discussed here are the most often used. They will allow you to generate a variety of table appearances that will add style to your pages.

# Conclusion

This has been a very basic discussion of some aspects of HTML/XHTML tables. There are many other attributes available for tables, rows and cells. They were not discussed here, not because they aren't important, but because they go beyond the scope of basic usage.

We have only touched on the intricacies of tables. In the preface, I recommended two books about HTML/XHTML coding. They are excellent treatise and explain things more fully than I could do in this document.

I hope this has been helpful to you. Tables are really not difficult once you understand their structure. Diagramming them on paper is a good way to develop a table that does what you want, not what the browser wants.

I didn't forget. The answers to the practice tables are on the next three pages.

Good luck with tables.

George McCartney

## Practice Table 1

```
<table border="3"
  style="width: 50%;
    margin-left: 25%;">
<tr>
  <!-- ROW 1 -->
  <td colspan="2"
    class="center">&nbsp;</td>
  <!-- <td> </td> cell not used - position take by preceding cell -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <!-- ROW 2 -->
  <td>&nbsp;</td>
  <td colspan="2"
    rowspan="2">&nbsp;</td>
  <!-- <td> </td> cell not used - position take by preceding cell -->
  <td>&nbsp;</td>
</tr>
<tr>
  <!-- ROW 3 -->
  <td>&nbsp;</td>
  <!-- <td> </td>
    <td> </td> cells not used - position take by cell 2, row 2 -->
  <td>&nbsp;</td>
</tr>
<tr>
  <!-- ROW 4 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
</table>
```

## Practice Table 2

```
<table border="3"
  style="width: 50%;
  margin-left: 25%;">
<tr>          <!-- ROW 1 -->
  <td>&nbsp;</td>
  <td rowspan="2">&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>          <!-- ROW 2 -->
  <td>&nbsp;</td>
  <!-- <td> </td> cell not used - position take by cell 2, row 1 -->
  <td>&nbsp;</td>
</tr>
<tr>          <!-- ROW 3 -->
  <td colspan="3">&nbsp;</td>
  <!-- <td> </td>
  <td> </td> cells not used - position take by cell 1, row 3 -->
</tr>
<tr>          <!-- ROW 4 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
</table>
```

### Practice Table 3

```
<table border="3"
  style="width: 298px;
  margin-left: 35%;">
<tr>          <!-- ROW 1 -->
  <td>&nbsp;</td>
  <td colspan="2"
    rowspan="2">&nbsp;</td>
  <!-- <td> </td> cell not used - position take by previous cell -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>          <!-- ROW 2 -->
  <td>&nbsp;</td>
  <!-- <td> </td>
    <td> </td> cells not used - position take by cell 2, row 1 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>          <!-- ROW 3 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td rowspan="2">&nbsp;</td>
  <td>&nbsp;</td>
  <td colspan="2"
    rowspan="4">&nbsp;</td>
  <!-- <td> </td> cell not used - position take by previous cell -->
</tr>
<tr>          <!-- ROW 4 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <!-- <td> </td> cell not used - position take by cell 3, row 3 -->
  <td>&nbsp;</td>
  <!-- <td> </td>
    <td> </td> cells not used - position take by cell 5, row 3 -->
</tr>
<tr>          <!-- ROW 5 -->
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
```

```

<td>&nbsp;</td>
<!-- <td> </td>
    <td> </td> cells not used - position take by cell 5, row 3 -->
</tr>
<tr>                                <!-- ROW 6 -->
<td colspan="2">&nbsp;</td>
<!-- <td> </td> cell not used - position take by previous cell -->
<td colspan="2"
    rowspan="3">&nbsp;</td>
<!-- <td> </td> cell not used - position take by previous cell -->
<!-- <td> </td>
    <td> </td> cells not used - position take by cell 5, row 3 -->
</tr>
<tr>                                <!-- ROW 7 -->
<td>&nbsp;</td>
<td>&nbsp;</td>
<!-- <td> </td>
    <td> </td> cells not used - position take by cell 3, row 6 -->
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>                                <!-- ROW 8 -->
<td>&nbsp;</td>
<td>&nbsp;</td>
<!-- <td> </td>
    <td> </td> cells not used - position take by cell 3, row 6 -->
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>                                <!-- ROW 9 -->
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>

```