**Creating Fixed Width Tables**

Fixed width tables require careful handling, and are dependent on very strict coding practices. The rigidity of these practices is due to the algorithms used by most browsers to draw tables. Remember that HTML has changed a very great deal over the years, and that most web browsers were built using code written in the early 1990's, when HTML was quite different than it is now. The browsers are changing slowly, but for the foreseeable future we must use modern, standards-compliant coding practices in combination with unusually restrictive syntax requirements in order to achieve consistent, reliable results.

Here is a very simple fixed width table:

<table width="250" border="1">

<tr>

<th>Pets</th>

<th>Colors</th>

</tr>

<tr>

<td>Cats</td>

<td>Red</td>

</tr>

<tr>

<td>Dogs</td>

<td>Blue</td>

</tr>

</table>

Displayed:

|  |  |
| --- | --- |
| **Pets** | **Colors** |
| Cats | Red |
| Dogs | Blue |

As you can see, this table is 250 pixels wide. The HEIGHT attribute is not usually set for real world tables, to allow the table to expand or contract to fit the available content.

Fixed width tables, when **BORDER="0"**, are demonstrably and actually the width they proclaim; they are the only constructions in HTML, besides images, which deliver precise sizes.

Individual table cells, sadly, can NOT be set to precise sizes using the WIDTH attribute of the TH or TD tags. Individual table cells may only be reliably sized using GIFs, not the WIDTH attribute. The WIDTH attribute of the TH and TD tags produces results which may or MAY NOT be approximately the size requested; they are not accurate.

To achieve accurate and consistent table cell widths, tables must have most of their built-in spacing taken out. BORDER, CELLPADDING, and CELLSPACING must all be set to "0", i.e. **<table width="500" border="0"cellpadding="0" cellspacing="0">**. Spacing between columns of cells, then, must be created with table cells between the main columns; these "spacer" columns of cells must be filled with some sort of "spacer" GIFs.

**Spacer GIFs**

A spacer GIF is a 3 pixel by 3 pixel GIF which is the same color as the background color of your web page. Realistically, only three colors of spacer GIFs are used: white, black, and transparent (the transparent spacer is used with all background colors and designs except white and black).

Because spacer GIFs are a flat color (or transparent), they may be resized safely in the IMG tag using the WIDTH and HEIGHT attributes.

<img src="spacer.gif" width="125" height="15" />

Note: Do NOT use 1 pixel by 1 pixel spacer GIFs, as these are not properly recognized by all web browsers; 3 pixel by 3 pixel GIFs are a safe spacer GIF size. Spacer gifs can also be easily created in Paint Shop Pro or Photoshop.  
  
An alternative to using spacer gifs is to use &nbsp; as the content of a table cell. For example:

<table width="250" border="1">

<tr>

<th>Pets</th>

<th width="30">&nbsp;</th>

<th>Colors</th>

</tr>

<tr>

<td>Cats</td>

<td width="30">&nbsp;</td>

<td>Red</td>

</tr>

<tr>

<td>Dogs</td>

<td width="30">&nbsp;</td>

<td>Blue</td>

</tr>

</table>

Displayed:

|  |  |  |
| --- | --- | --- |
| **Pets** |  | **Colors** |
| Cats |  | Red |
| Dogs |  | Blue |

**More About Fixed Width Tables**

Pages containing real-world tables often have BODY tags set to display content directly against the edge of the browser window (without the browser-generated white space around the margins of the web page, as this space is inconsistent from browser to browser). To remove the browser-generated space around the content of a web page, the following four attributes of the BODY tag must be set equal to "0": MARGINHEIGHT and MARGINWIDTH (for Firefox), and LEFTMARGIN and TOPMARGIN (for Internet Explorer).

<body marginheight="0" marginwidth="0" leftmargin="0" topmargin="0">

Removing the margin from a web page is useful if we have graphical content which we would like to have rest against the edge of the browser window, as in [this example](http://insight.ccsf.edu/file.php/4928/Module6/realw1.html).

If we take our text-based table from the previous section above, however, and place it immediately below the graphical table from the preceding example, the results are less than satisfactory; [this example](http://insight.ccsf.edu/file.php/4928/Module6/realw2.html) shows what I mean. Although the spacer GIF row at the top of the table prevents the columns of text from slamming into the top of the header table, notice how the text in the left column runs into the left side of the browser window.

This problem can be avoided by creating an extra column on the left side of the text-based table which generates a gutter, as demonstrated in [this example](http://insight.ccsf.edu/file.php/4928/Module6/realw3.html).

Placing multiple tables into a web page, one atop the other (as seen in the examples above), is a very common practice; it allows for tables with different numbers of columns (or different widths of columns) to appear easily on the same page.

Sometimes, complex elements within a column of a table are created using an additional TABLE or set of TABLEs nested within the cell; we'll look into nesting tables within table cells at greater length in a later section.

You may nest as many sibling tables as you like inside of a parent table cell. It is important to note, however, that you must NEVER nest tables more than one level deep (i.e. no grandchild or great-grandchild nested tables). In other words, don't try to nest a table inside of one of your nested table's cells; that is forbidden. Nesting tables more than one level deep creates printing and display difficulties that you will want to avoid.

**Comments**

Because real-world tables can become very complex and lengthy, make certain to use HTML comments to mark the beginning and ending of important areas where content is likely to be inserted or change regularly. This will help yourself and others find the correct insertion point for content quickly and easily in the hundreds or thousands of lines of your HTML code.

Example (abbreviated):

<td valign="top">

<!-- Begin Main Area -->

<p>Content here...</p>

<!-- End Main Area -->

</td>

Example (abbreviated):

<!-- Begin Article One -->

<table width="500" border="0" cellpadding="0" cellspacing="0">

<tr>

<td>Some content...</td>

</tr>

</table>

<!-- End Article One -->

**Printable Pages**

Never use tables to hold printable content, especially if they have been modified with CSS; you'll get very irregular printed results. Create printable pages with vanilla HTML, no TABLES! Better yet, if you have the time and inclination, create printable and downloadable Adobe Acrobat (PDF) files, which are the standard for web-based documentation, as well as being very consistent in appearance cross-platform (much more so than HTML pages).

**Miscellaneous**

Never put content in a cell which would make that cell BIGGER than your defined column width; you'll get very irregular results if you do.