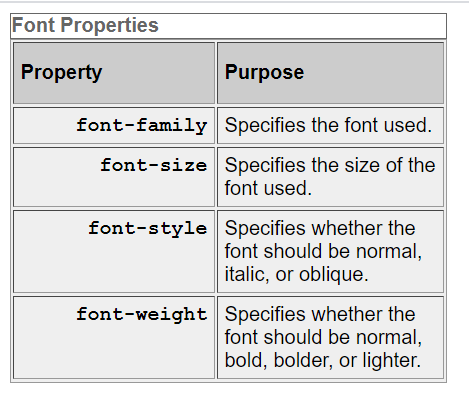
**Tables**

In the last chapter, you saw a couple of examples that use CSS with tables. Properties that are commonly used with the <table>, <td>, and <th> elements include the following:

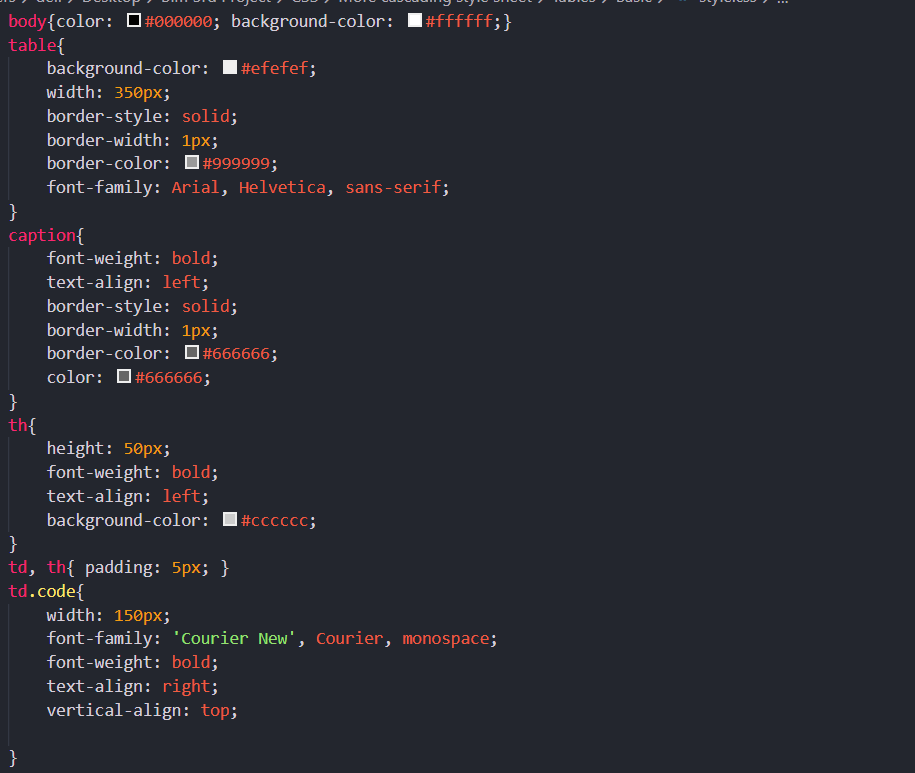
* border to set the properties of the border of a table.
* padding to set the amount of space between the border of a table cell and its content — this property is very important to make tables easier to read.
* Properties to change text and fonts.
* text-align to align writing to the left, right, or center of a cell.
* vertical-align to align writing to the top, middle, or bottom of a cell.
* width to set the width of a table or cell.
* height to set the height of a cell (often used on a row as well).
* background-color to change the background color of a table or cell.
* background-image to add an image to the background of a table or cell.

You should be aware that, apart from the background-color and height properties, it is best to avoid using these properties with <tr> elements, as browser support for these properties on rows is not as good as it is for individual cells.

To demonstrate how some of these properties are used with a table, take a look at the one shown in Figure; it might look familiar because you saw it at the beginning of the last chapter, but this time it has an added <caption> element:



Now take a look at the style sheet for this table:



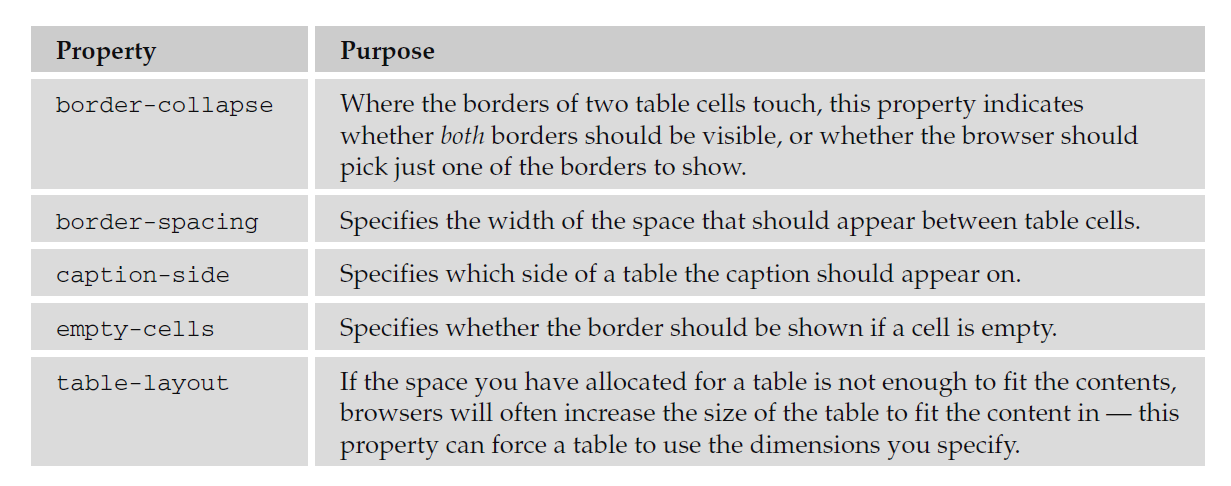
Here are some key points to note about this example. You will be altering settings of some of these properties using new properties that you will meet throughout this section.

* The rule for the < table > element uses a width property to fix the width of the table to 350 pixels; otherwise, it would take up as much of the screen as needed to show as much text as possible on one line.
* The rule for the < table > element also has a border property set, which creates a single-pixel border all around the table. Note, however, that none of the other cells in the table inherits this property.
* The rule that applies to the < caption > element has its font - weight, border, and text – align properties set. By default, the text is normal (not bold), aligned in the center, and without a border.
* The rule that applies to the <th> element sets the height of the headings to 50 pixels, and the text is aligned left (rather than centered, which is the default).
* There is a rule that applies to both the <th> and <td> elements, and this indicates that both should have a padding property set to 5px so that the content of the cells does not touch the border of those cells. Creating space around the cells is very important and makes the table more readable.
* The final rule states that the <td> elements whose class attribute has a value of code are given a width property whose value is 150px (150 pixels). This ensures that the content of this whole column remains on one line. Unfortunately, there is no way to assign a style to a column, but in the case of the width property, once it has been set on one element it does not need to be set on all the others in the column.

You may also have noticed in Figure that there is a white line around the two columns (which is particularly noticeable around table header cells). Browsers automatically add this to separate each cell from its neighbor. You can, however, remove this gap using a property called border-spacing, which you ’ll learn about in the next section.

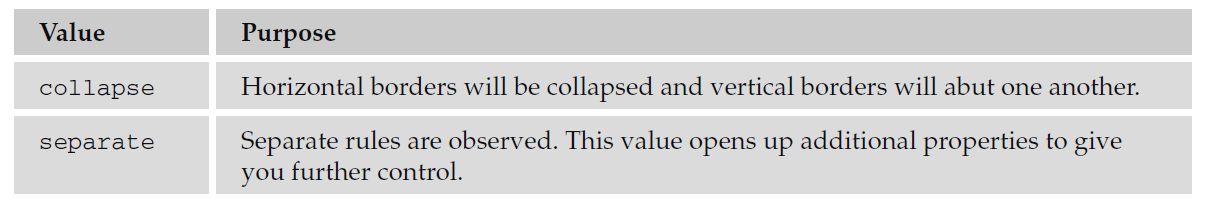
***Table-Specific Properties:***

In the following section you will meet five properties that can only be used with tables, and also some values for the border - style property that only apply to tables. Most of these properties were first supported in IE7 and FF2.



***The border-collapse Property:***

Where two table cells meet, you can tell the browser to show just one of the borders (rather than both — which is the default behavior). You can do this using the border-collapse property, which can take two values:

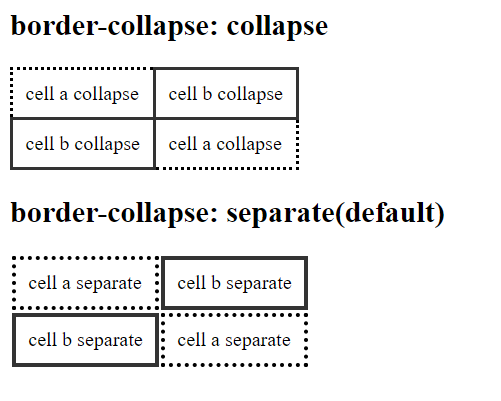


If two adjacent table cells have different border styles, and you have specified that borders should be collapsed, there is a complex set of rules to specify which border should be shown —rather than try to learn these rules it is quicker to simply try your table out in a browser.

To illustrate how the border-collapse property works, the following style rules apply to two tables: the first has a border-collapse property with a value of collapse, the second has a value of separate, and both tables contain adjacent cells with dotted and solid lines:



Figure below shows you how, with a value of collapse, the browser collapses borders into each other so that the solid border takes precedence over the dotted border. This wouldn’t’ t look as odd if the borders were both solid, but it does illustrate the point well.



If you do not specify that the borders should be collapsed, then two further properties control border presentation:

* border-spacing
* empty-cells

The following sections discuss these properties.

***The border-spacing Property:***

The border - spacing property specifies the distance that separates adjacent cells’ borders. If you provide one value, it will apply to both vertical and horizontal borders:

Eg:

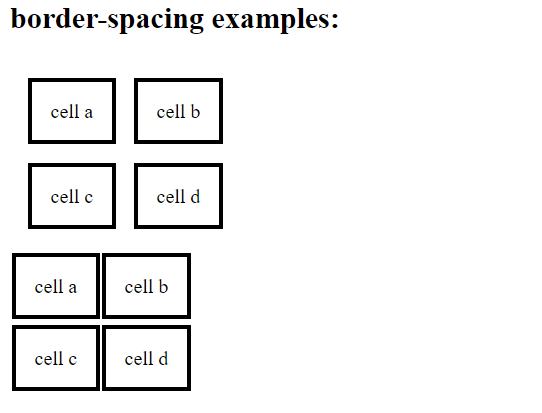
table.one {border-spacing:15px;}

Or you can specify two values, in which case the first refers to the horizontal spacing and the second to the vertical spacing:

Eg:

table.two {border-spacing:2px 4px;}

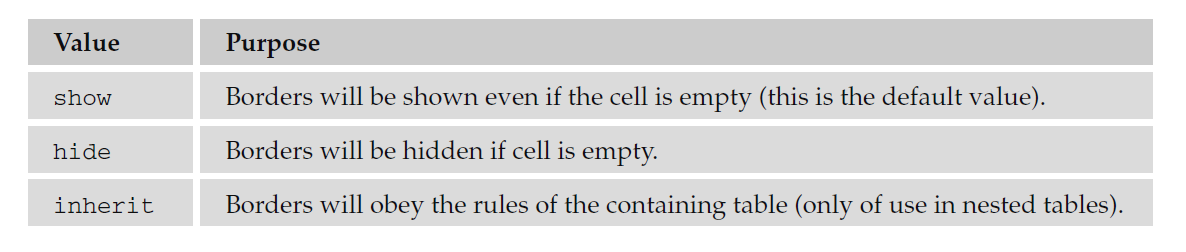
You can see what this looks like in Figure:



***The empty-cells Property:***

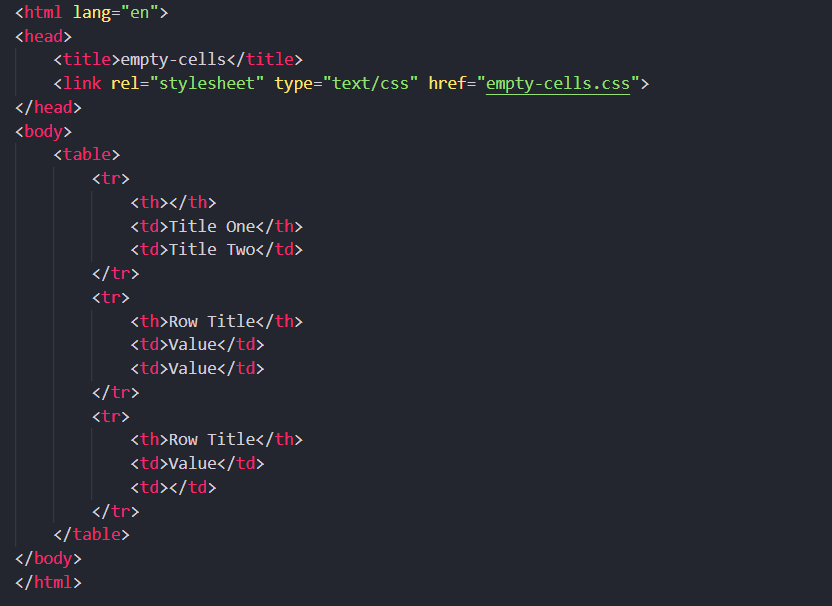
The empty-cells property indicates whether a cell without any content should have a border

displayed. It can take one of three values, as you can see in the table that follows.

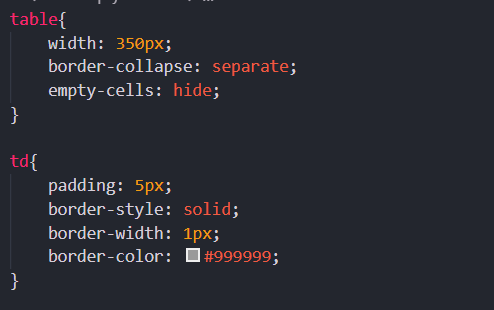


*If you want to explicitly hide or show borders of empty cells, you should use this property because some versions of IE and Firefox treat empty cells differently.*

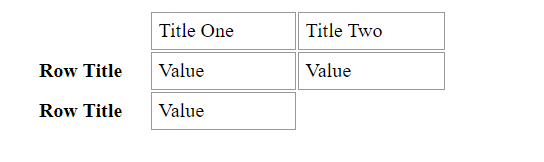
Here you can see a table with two empty cells: an empty <th> element and an empty <td> element:



The following code shows the empty-cells property used to hide borders of empty cells in the <table> element:

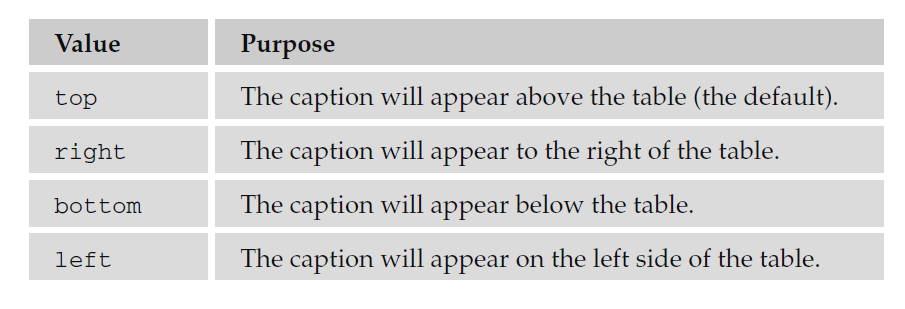


Output:



***The caption-side Property:***

The caption-side property is for use with the <caption> element to indicate on which side of the table the caption should go. The following table lists the possible values.



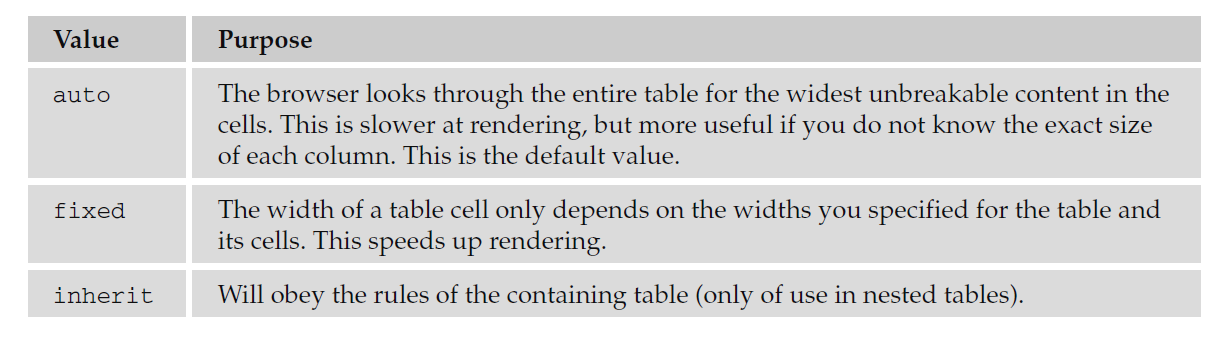
For example, here you can see the caption being set to the bottom of the table:

caption {caption-side: bottom}

***The table-layout Property:***

When you specify a width for a table or table cell, but the content does not fit into the space you have allowed, a browser can give the table more space to fit the content. The table–layout property allows you to force the browser to stick to the widths you specify, even if this makes the content unreadable.

See the table that follows for the three possible values this property can take.



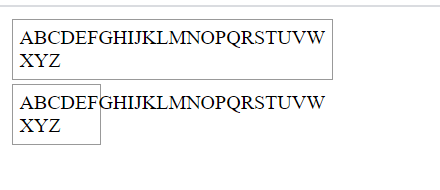
In the following example there are two tables, each with just one cell. The cells contain the letters of the alphabet, and there is a space before the last three letters. Normally, each table cell will be as wide as the longest unbroken set of characters in a cell — in this case, the letters A through W:



Now, if you look at the CSS for this example, you can see that the width of the table is set to 75 pixels — not enough for the letters A through W. One table has the table - layout property set to auto, the other to fixed:



You can see the results of this example in Figure below; by default, the table will make enough space for the letters A through W. However, when the second table is forced to stick to the width specified in the CSS, the letters spill out over the edge of the table.



To prevent the letters spilling out over the edge you could use the overflow property, which you will meet later in the chapter.

There are several other CSS properties that allow you to control groups of cells in one rule. They are not covered in this book because support for them is still patchy. Should you want to look them up on the Web, they are as follows:

* IE 5 and later supports table - header - group and table - footer - group.
* Firefox supports inline-table, table-row, table-column-group, table-column, table-row, and table-cell.