

# /CSS/style an HTML data table



London-based web designer [Inayaili de León](#) ([yaili.com](#)) explains how to create an HTML data table using clean and semantic markup

Percentage of visitors per browser in the last quarter of 2009

	Firefox	Safari	Internet Explorer			Chrome	Opera
			IE 6	IE 7	IE 8		
SEPTEMBER	60.77%	12.14%	6.32%	4.34%	1.74%	7.59%	4.26%
OCTOBER	57.43%	11.71%	7.77%	5.33%	2.13%	8.05%	6.38%
NOVEMBER	56.66%	17.81%	4.52%	3.10%	1.24%	9.86%	3.97%
DECEMBER	57.37%	18.32%	3.34%	2.30%	0.92%	11.51%	3.17%
AVERAGE	58.06%	14.00%	5.49%	3.78%	1.51%	9.25%	4.45%

## Expert tip Multiple backgrounds

Instead of using the `border-image` property, you could use CSS3's multiple backgrounds to create the inner shadow effect. To do this you need four images, one for each corner of the cell. Now use the `background-image`, `background-position` and `background-repeat` properties, separating each value with a comma. For example, with the four images, you would use `background-image:url(top-right.png), url(bottom-right.png), url(bottom-left.png), url(top-left.png)` and `background-position:top right, bottom right, bottom left, top left`.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3 <head>
4
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <title>Percentage of visitors per browser in the last quarter of 2009</title>
7 <link type="text/css" rel="stylesheet" media="screen" href="style.css" />
8
9 <!--[if !IE 7]>
10 <link type="text/css" rel="stylesheet" media="screen" href="ie.css" />
11 <![endif]-->
12
13 </head>
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

**START Create the files** Create a new HTML file and link two stylesheets to it: `style.css` and `ie.css`, using a conditional comment for the latter, so that it only shows on IE7 and lower. Ensure you're using a valid DOCTYPE too. Here, we'll be using an HTML 4.01 Strict DOCTYPE.

```

1 <body>
2
3 <table>
4
5 <caption><h1>Percentage of visitors per browser in the last quarter of 2009</h1></caption>
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

**2 The markup** Create the `table` element and use the `caption` element, right after the opening `table` tag, to add a brief description of the table's contents. Because this is tricky to style across all browsers, enclose the definition text within heading tags (`h1`).

**What you'll learn** In this tutorial, we'll create a beautiful and original HTML data table using clean and semantic markup and by applying some of the latest CSS selectors and techniques. You'll also learn how to add in a few special details while still keeping the content accessible for older or less standards-compliant browsers

**Knowledge needed** Basic knowledge of HTML and CSS

**Requires** A text editor

**Project time** 30 minutes

**Styling CSS tables is one of those menial tasks that no web designer seems to particularly enjoy – after all, who wants to have to deal with loads of grey, amorphous information for a few hours?**

Thankfully, the days when we were using tables to lay out entire websites are long gone. Still, they are the right choice for styling certain types of data, so here we're going to try to make the act of styling a table a bit more exciting. With some advanced and uncommon CSS selectors, we'll style an HTML table to make it look less boxy.

Starting with a carefully laid out markup, we'll also ensure that everyone can access the

data within our table, even if they're using older browsers that can't render the finer details.

Finally, we'll explore the power of often under-used CSS selectors to avoid adding unnecessary classes or ids to our markup, keeping it as clean and simple as possible. And with a little help from some CSS3 properties – such as `text-shadow`, `border-radius` and `border-image` – we'll keep the number of images used to style the table to an absolute minimum. ●



### About the author

Name Inayaili de León

Site [yaili.com](#)

Areas of expertise CSS and user interface design

Clients Telmap, iSites

How do you like your coffee?

Skinny caramel macchiato, 1 shot

```

31 <tbody>
32
33 <caption>Percentage of visitors per browser in the last quarter of 2009</caption>
34
35 <thead>
36 <tr>
37 <th>
38 </th>
39 </tr>
40 </thead>
41
42 <tbody>
43 <tr>
44 <td>
45 </td>
46 </tr>
47 </tbody>
48
49 <tfoot>
50 <tr>
51 <td>
52 </td>
53 </tr>
54 </tfoot>
55
56 </table>
57
58 </body>
59 </html>

```

**3 Use thead, tbody and tfoot** The table's data is divided into three parts. Wrap the heading rows with the `thead` element. The `tfoot` element must come immediately after. The `tbody` element will wrap all the rest of the table's content. Each row is marked up with one `tr` element.

```

31 <thead>
32 <tr>
33 <th>
34 </th>
35 </tr>
36 </thead>
37
38 <tbody>
39 <tr>
40 <td>
41 </td>
42 </tr>
43 </tbody>
44
45 <tfoot>
46 <tr>
47 <td>
48 </td>
49 </tr>
50 </tfoot>
51
52 </table>
53
54 </body>
55 </html>

```

**4 Set scope** Each `th` has a scope, which can be either `row`, `col`, `rowgroup` or `colgroup`. The `th`s within `thead` have `scope=col`, except for Internet Explorer, which has `scope=colgroup`, since it's a header for three other columns (IE6, IE7 and IE8). `th`s in `tbody` and `tfoot` have `scope=row`.

```

31 <thead>
32 <tr>
33 <th>
34 </th>
35 </tr>
36 </thead>
37
38 <tbody>
39 <tr>
40 <td>
41 </td>
42 </tr>
43 </tbody>
44
45 <tfoot>
46 <tr>
47 <td>
48 </td>
49 </tr>
50 </tfoot>
51
52 </table>
53
54 </body>
55 </html>

```

**5 Specify colspan and rowspan** If a cell spans across two or more rows or columns, you must specify that with either the `rowspan` or `colspan` attribute. In this case, the Internet Explorer cell takes over three columns and all the other cells in `thead` span across two rows.

```

1 body {
2 font: 13px/1.5 "Helvetica Neue", Helvetica, Arial, sans-serif;
3 padding: 20px;
4 }
5
6 table, caption, thead, tfoot, tbody, th, td, h1 {
7 margin: 0;
8 padding: 0;
9 }
10
11 /* TABLE General */
12
13 table {
14 background: #A7DBD8;
15 border: 1px solid #79B2AF;
16 border-collapse: separate;
17 border-width: 0 1px 1px 0;
18 border-spacing: 4px;
19 color: #333333;
20 margin-top: -20px;
21 max-width: 900px;
22 min-width: 400px;
23 padding: 20px 5px 5px 5px;
24 width: 95%;
25 }

```

**6 Reset the styles** Reset the margins and paddings of the elements, then define a font size of 13px, a line height of 1.5 and the font stack of the body, starting with the ideal font (Helvetica Neue) through to the most generic font (sans-serif). Next, add 20px of padding to the `body`.

```

11 /* TABLE General */
12
13 table {
14 background: #A7DBD8;
15 border: 1px solid #79B2AF;
16 border-collapse: separate;
17 border-width: 0 1px 1px 0;
18 border-spacing: 4px;
19 color: #333333;
20 margin-top: -20px;
21 max-width: 900px;
22 min-width: 400px;
23 padding: 20px 5px 5px 5px;
24 width: 95%;
25 }

```

**7 Format the table** Add a background, text and a 1px border to the left and bottom. To ensure each cell is separate, add a `border-spacing` of 4px and set the borders to `separate`. Define a `max-` and `min-width` then pull the table -20px to the top, to override the 20px of top padding.

```

31 caption h1 {
32 background: #F38630;
33 border: 1px solid #B66A2E;
34 border-width: 0 1px 1px 0;
35 color: #38210E;
36 display: inline-block;
37 font-size: 16px;
38 font-weight: bold;
39 left: -10px;
40 padding: 3px 12px;
41 position: relative;
42 top: 14px;
43 }
44
45
46
47

```

**8 Sort the caption** Set a larger font-size for `caption h1`. Add a background colour, right and bottom border, and a text colour. Use `display:inline-block` so the caption's less than 100 per cent wide. To offset it, add `position:relative`, `left:-10px` and `top:14px`. Or, in the `ie.css` file, use `top:2px`.

```

31 caption h1 {
32 background: #F38630;
33 border: 1px solid #B66A2E;
34 border-width: 0 1px 1px 0;
35 color: #38210E;
36 display: inline-block;
37 font-size: 16px;
38 font-weight: bold;
39 left: -10px;
40 padding: 3px 12px;
41 position: relative;
42 text-shadow: 0 1px 1px #EECAA8;
43 top: 14px;
44 }
45
46
47

```

**9 Set text-shadow** Using the `text-shadow` property, add a 1px, light orange shadow to the text. The three numeric values for this property offset it 0px to right, 1px to the bottom and set a 1px blur radius. We will use this technique for other elements in the table.

```

66 /* THEAD */
67
68 thead th {
69 background: #D3EDEB;
70 background: rgba(255, 255, 255, 0.5);
71 }
72
73
74
75
76
77
78
79
80
81
82
83
84
85

```

```

46 th,
47 td {
48 border-width: 5px;
49 padding: 1px 6px;
50 }
51
52 th {
53 color: #31605D;
54 }
55
56 thead,
57 td {
58 text-align: center;
59 }
60
61 td:empty {
62 background: #A7DBD8;
63 border: 0;
64 }
65
66
67
68
69
70

```

**10 th and td cells** Add `border-width:5px` and `padding:1px 6px` to the `th` and `td` cells. Change `th` cells' colour to `#31605D`. Now align the `thead` and `td` text to the centre. Using `:empty`, make empty `td` cells invisible by giving them the same colour as the table and no border.

**11 RGBA colours** Add a white transparent background to the `thead` headers using `RGBA` colours (remember the last value is the alpha channel). Also provide a fall-back background colour before the `RGBA` one (in this case, `#D3EDEB`, a light blue). This way, if a browser doesn't support `RGBA` colours, it will ignore the second property.



```

66 /* THEAD */
67
68 thead th {
69     background: #D3EDEB;
70     background: rgba(255, 255, 255, 0.5);
71 }
72
73 thead tr:nth-of-type(n+2) th {
74     background: #C1E5E3;
75     background: rgba(255, 255, 255, 0.3);
76 }
77
78
79
80
81
82

```

**12 The :nth-of-type pseudo-class** To style the IE6, IE7 and IE8 headers, use the `:nth-of-type` selector. The `n+2` argument targets all `tr` elements, starting from the second. So we'll style all `th` elements that are within any table row (`tr`) that's not first inside `thead`.

```

78 /* TFOOT */
79
80 tfoot td {
81     background: #95C4C1;
82     background: rgba(0, 0, 0, 0.1);
83     font-style: italic;
84     font-weight: bold;
85     font-family: Georgia, "Times New Roman", Times, serif;
86     padding: 5px 6px;
87     text-shadow: 0 1px 1px #FFFFFF;
88 }
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106

```

**13 Style tfoot** Add a solid background colour to the `tfoot` data cells, followed by a transparent RGBA one. Make the text bold italic and use a serif font to make the content stand out. Add some padding to compensate for the lack of border and a white 1px text shadow.

```

90 /* TBODY */
91
92 th[scope=row] {
93     font-variant: small-caps;
94     letter-spacing: 1px;
95 }
96
97
98
99
100
101
102
103
104
105
106

```

**14 Attribute selectors** To make the headers for the rows (the months) look a bit different, target them with an attribute selector. Using `th[scope=row]`, you'll target all `th` elements with a `scope` of row. Add `font-variant:small-caps` and `letter-spacing:1px` to make them look nicer.

```

97 tbody td {
98     font-style: italic;
99     font-family: Georgia, "Times New Roman", Times, serif;
100     text-shadow: 0 1px 1px #FFFFFF;
101     background: #E5F2F1;
102 }
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

```

**15 Format data cell text** Make all the data cells within `tbody` italic and use a serif font, such as Georgia. Add a 1px text shadow. Then provide browsers that won't support the `border-image` property with a fall-back background colour (explained in the next step).

```

104 tbody tr:nth-child(odd) td {
105     ...
106 }
107 tbody tr:nth-child(even) td {
108     ...
109 }
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

```

**16 The :nth-child pseudo-class** To create a zebra-striped effect on the table body rows, use the `:nth-child` pseudo-class. You can target every other row by using `even` and `odd` arguments. These are the equivalent of `:nth-child(2n)` and `:nth-child(2n+1)`, respectively.

```

104 tbody tr:nth-child(odd) td {
105     -moz-border-image: url(shadow-odd.png) 5 repeat;
106     -webkit-border-image: url(shadow-odd.png) 5 repeat;
107     border-image: url(shadow-odd.png) 5 repeat;
108 }
109 tbody tr:nth-child(even) td {
110     -moz-border-image: url(shadow-even.png) 5 repeat;
111     -webkit-border-image: url(shadow-even.png) 5 repeat;
112     border-image: url(shadow-even.png) 5 repeat;
113 }
114
115 tbody tr:nth-child(odd):hover td,
116 tbody tr:nth-child(even):hover td {
117     -moz-border-image: url(shadow-tr-hover.png) 5 5 5 5 repeat;
118     -webkit-border-image: url(shadow-tr-hover.png) 5 5 5 5 repeat;
119     border-image: url(shadow-tr-hover.png) 5 5 5 5 repeat;
120 }
121
122 tbody tr:nth-child(odd) td:hover,
123 tbody tr:nth-child(even) td:hover {
124     -moz-border-image: url(shadow-td-hover.png) 5 repeat;
125     -webkit-border-image: url(shadow-td-hover.png) 5 repeat;
126     border-image: url(shadow-td-hover.png) 5 repeat;
127 }
128
129
130
131

```

**18 Use border-image** Define the image's location. Set each slice to be cut five pixels inwards from the edges (we previously defined that `td` elements have a 5px `border-width`). Set the `repeat` keyword so that the image is centred horizontally. Repeat this for the `tr` and `td`'s hover effect.



**17 Set data cells' background** Use the `border-image` property to add a pleasant shadowed border to the `tbody` data cells. After creating the image (you'll find all the images you need on this issue's CD), decide how much of it you'll need for the border. In this case, we need 5px for each side and corner.

```

129 /* BORDER RADIUS */
130
131 table,
132 caption h1,
133 thead th,
134 tfoot td,
135 tbody td {
136     -moz-border-radius: 2px;
137     -webkit-border-radius: 2px;
138     border-radius: 2px;
139 }
140
141
142
143
144
145

```

**FINISH Border-radius** Because the vendor-specific properties for `border-radius` take a lot of space and time to write, let's put them at the end of the stylesheet. Add `border-radius` to `table`, `caption h1`, `thead th`, `tfoot td` and `tbody td`. The end result is a great-looking table.

**Expert tip** **RGBA colours**

RGBA colours include an alpha value, which enables you to alter a colour's opacity. This is useful, because it means you can set the colour for an element independent of what's below it (there's no need to create 'fake' transparency effects) and of what it contains (unlike the `opacity` property, which makes all enclosed elements transparent too). You should, however, ensure you provide a suitable fall-back colour for browsers that don't support RGBA – such as IE – and make sure the second declaration has a higher specificity, to override the fall-back.