

An abstract graphic featuring three sets of concentric orange circles. One large set is in the top right, a medium set is in the middle right, and another large set is in the bottom right. Thin orange lines extend from the top left towards the middle set of circles.

# FED-130 Accessibility: Session 2

## **BOOK EXTRACT**

Web Standards Solutions: The Markup and Style  
Handbook, ISBN: 9781430219200  
*Chapter 3 Are Tables Evil?*

Dan Cederholm

17 February 2010  
©2010 Central Institute of Technology  
Version 1.0

## Book extract

### Web Standards Solutions: The Markup and Style Handbook Chapter 3: Are Tables Evil?

## A table that everyone can sit at

One of the reasons that tables get a bad rap is due to the accessibility problems they can cause if not carefully used. For instance, screen readers can have difficulty reading them properly, and small-screened devices are often hindered by tables when they are used for layout. But there are a few simple things we can do to increase the accessibility of a *data* table, while at the same time creating a lean structure that will be easy to style later on with CSS.

Let's take a look at the simple table example found in Figure 3-1, illustrating one of American baseball's longest droughts. (A drought no more, with the Red Sox victories of 2004 and 2007. I'm convinced this chapter had everything to do with it.)

Boston Red Sox World Series Championships

Year	Opponent	Season Record (W-L)
1918	Chicago Cubs	75-51
1916	Brooklyn Robins	91-63
1915	Philadelphia Phillies	101-50
1912	New York Giants	105-47

**Figure 3-1.** Example of a typical data table

Although at one time an extremely depressing set of statistics for a Red Sox fan to look at, Figure 3-1 is a perfect example of tabular data. There are three table headers (Year, Opponent, and Season Record (W-L)) followed by the data for each of the four years presented.

Above the table is a caption, defining what is contained below.

Marking up this data table is relatively straightforward, and we might do something like the following:

```
<p align="center">Boston Red Sox World Series Championships</p>
<table>
<tr>
<td align="center"><b>Year</b></td>
<td align="center"><b>Opponent</b></td>
<td align="center"><b>Season Record (
W-L)</b></td>
</tr>
<tr>
<td>1918</td>
<td>Chicago Cubs</td>
<td>75-51</td>
</tr>
<tr>
<td>1916</td>
<td>Brooklyn Robins</td>
<td>91-63</td>
</tr>
<tr>
<td>1915</td>
<td>Philadelphia Phillies</td>
```

```

<
td>101-50</
td>
</tr>
<tr>
<td>1912</td>
<td>New York Giants</td>
<
td>105-47</
td>
</tr>
</table>

```

That should render close to what we see in Figure 3- 1; however, there are a few improvements we can make here.

First off, we can treat the title of the table, “Boston Red Sox World Series Championships,” a little more semantically correct by using the `<caption>` element. The `<caption>` is required to immediately follow the opening `<table>` element and usually holds the title and/or nature of what’s contained within the table.

Visually, it will be easy for sighted people to understand the table’s purpose, while assisting those browsing by nonvisual means as well.

Let’s replace the opening paragraph and add in a proper `<caption>`:

```

<table>
<caption>Boston Red Sox World Series Championships</caption>
<tr>
<td align="center"><b>Year</b></td>
<td align="center"><b>Opponent</b></td>
<td align="center"><b>Season Record (
W-L)</
b></td>
</tr>
<tr>
<td>1918</td>
<td>Chicago Cubs</td>
<
td>75-51</
td>
</tr>
<tr>
<td>1916</td>
<td>Brooklyn Robins</td>
<
td>91-63</
td>
</tr>
<tr>
<td>1915</td>
<td>Philadelphia Phillies</td>
<
td>101-50</
td>
</tr>
<tr>
<td>1912</td>
<td>New York Giants</td>
<
td>105-47</

```

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

It's important for captions to quickly convey what the data is that follows. By default, most visual browsers will place text that's contained within `<caption>` elements centered and just above the very top of the table. We could, of course, alter the default styling of the caption after the fact using CSS if we wished—and we'll do just that later in the “Extra credit” section of this chapter. The fact that it's now in its own unique element makes this nice and easy.

## Adding a summary

Additionally, we could add the `summary` attribute to the `<table>` element, further explaining the purpose and contents of what is contained in our table. The `summary` is especially helpful for those using nonvisual means to read the information.

The following shows the `summary` attribute and value added to our table example:

```
<table summary="This table is a chart of all Boston Red Sox World
Series wins.">
<caption>Boston Red Sox World Series Championships</caption>
<tr>
<td align="center"><b>Year</b></td>
<td align="center"><b>Opponent</b></td>
<td align="center"><b>Season Record (
W-L)</b></td>
</tr>
<tr>
<td>1918</td>
<td>Chicago Cubs</td>
<td>75-51</td>
</tr>
<tr>
<td>1916</td>
<td>Brooklyn Robins</td>
<td>91-63</td>
</tr>
<tr>
<td>1915</td>
<td>Philadelphia Phillies</td>
<td>101-50</td>
</tr>
<tr>
<td>1912</td>
<td>New York Giants</td>
<td>105-47</td>
</tr>
</table>
```

## The head(s) of the table

Table headers are important to make use of when building data tables. Instead of using a presentational element like `<b>` to visually cue the user that the cell is of importance in grouping the data that follows, we can take advantage of the `<th>` element, much like we used proper heading elements for section page content in Chapter 2.

Visual browsers might render information contained in `<th>` elements as bold and centered by default, but again we can use the uniqueness of the `<th>` element to style these important cells differently from the rest of the table data that's contained in a `<td>`.

In addition to their presentational advantages, using `<th>` elements can be beneficial to nonvisual browsers as well—as we'll dive into further on.

The headers in our example table are found in the top row: Year, Opponent, and Season Record (W-L).

Let's replace our previous, presentational markup with proper headers:

```
<table summary="This table is a chart of all Boston Red Sox World
Series wins.">
<caption>Boston Red Sox World Series Championships</caption>
<tr>
<th>Year</th>
<th>Opponent</th>
<th>Season Record (
W-L)</
th>
</tr>
<tr>
<td>1918</td>
<td>Chicago Cubs</td>
<
td>75-51</
td>
</tr>
<tr>
<td>1916</td>
<td>Brooklyn Robins</td>
<
td>91-63</
td>
</tr>
<tr>
<td>1915</td>
<td>Philadelphia Phillies</td>
<
td>101-50</
td>
</tr>
<tr>
<td>1912</td>
<td>New York Giants</td>
<
td>105-47</
td>
</tr>
```

```
</table>
```

Using `<th>` elements to mark up the header cells will give us the same visual results shown in Figure 3-1. Let's review why this is a preferred way:

- We eliminate the need for extra presentational markup to differentiate the header cells from normal ones.
- By default, most visual browsers will render text within `<th>` elements bold and centered—making it easier to see the difference between headers and data.
- Because of their uniqueness from normal `<td>` elements, we can later style table headers differently from other cells in the table.

There is also an additional reason for using table headers that we'll discuss next.

## Header and data relationships

To make things a bit more organized for people using a screen reader to read the information from our table, we can utilize the `headers` attribute to associate header cells with the corresponding data found in `<td>` elements. Doing this will allow the screen reader to read the header and data information in a more logical order, rather than strictly reading each row left to right as it normally might.

Let's again use our Red Sox table as an example on how to achieve this. First, we'll need to add a unique id to each `<th>` in our table. We can then add the `headers` attribute to each data cell to match the two up accordingly.

Adding the id to each header is as simple as this:

```
<table summary="This table is a chart of all Boston Red Sox World
Series wins.">
<caption>Boston Red Sox World Series Championships</caption>
<tr>
<th id="year">Year</th>
<th id="opponent">Opponent</th>
<th id="record">Season Record (
W-L)</
th>
</tr>
<tr>
<td>1918</td>
<td>Chicago Cubs</td>
<
td>75-51</
td>
</tr>
<tr>
<td>1916</td>
<td>Brooklyn Robins</td>
<
td>91-63</
td>
</tr>
<tr>
<td>1915</td>
<td>Philadelphia Phillies</td>
<td>101-50</td>
</tr>
```

```

<tr>
<td>1912</td>
<td>New York Giants</td>
<
td>105-47</
td>
</tr>
</table>

```

We've used short, descriptive names for each header id. Now we can add the appropriate headers attribute to each data cell—with its value corresponding to the id that it's associated with.

```

<table summary="This table is a chart of all Boston Red Sox World
Series wins.">
<caption>Boston Red Sox World Series Championships</caption>
<tr>
<th id="year">Year</th>
<th id="opponent">Opponent</th>
<th id="record">Season Record (
W-L)</
th>
</tr>
<tr>
<td headers="year">1918</td>
<td headers="opponent">Chicago Cubs</td>
<td headers="
record">75-51</
td>
</tr>
<tr>
<td headers="year">1916</td>
<td headers="opponent">Brooklyn Robins</td>
<td headers="
record">91-63</
td>
</tr>
<tr>
<td headers="year">1915</td>
<td headers="opponent">Philadelphia Phillies</td>
<td headers="
record">101-50</
td>
</tr>
<tr>
<td headers="year">1912</td>
<td headers="opponent">New York Giants</td>
<td headers="
record">105-47</
td>
</tr>
</table>

```

When we create relationships between our header and data information, a screen reader might read this table as follows: “Year: 1918, Opponent: Chicago Cubs, Season Record (W-L): 75- 51,” and so on for each table row. This makes a little more sense than hearing each row read left to right. It also doesn't hurt for us to have those unique ids for each <th> in our table. We could later take advantage of that identification with exclusive CSS rules.