

Accessible Tables

Whoa, Tables!

Tabular data presents some unique difficulties for the accessibility-minded designer. Consider the user accessing your table via a screen reader. It is voiced linearly, one cell after another. Unless you add markers to help them parse the information, it can get confusing fast.

Layout Tables

- Layout tables are not evil, but they are deprecated.
- Layout tables are easily parsed by most screen readers and present few accessibility challenges. However, use only `<table>`, `<tr>`, and `<td>` elements, plus an empty summary. That's it!
- Don't use accessibility-related table markup for layout tables! "If a table is used for layout, do not use any structural markup for the purpose of visual formatting", says the W3C.
- Use "Empty summary" for validating layout tables, like "empty alt" for images.
- Use relative rather than absolute values.
- Table borders have no access issues. Color use, however, does!
- Consider using CSS or other tools to separate content from presentation instead of using layout tables.

Data Tables

- Data tables, as opposed to layout tables, present information in a matrix, and have column and rows that show the meaning of the information in the grid.
- You don't need to be blind to get thoroughly lost in a complex data table.
- In order for a data table to be accessible, it must have proper markup.
- You must designate row and/or column headers. The `<td>` tag is used for table data cells and the `<th>` tag is used for table header cells.
- Use relative as opposed to absolute values throughout.
- Now that you've created headers, associate the cells with the appropriate headers. There are two ways - `scope` and `id`.

Associating Data Cells with the Appropriate Headers: scope

The `scope` attribute should be used on simple data tables. Here is the markup for a table using the `scope` attribute:

```
<table border="1" align="center">
<caption>Shelly's Daughters</caption>
<tr>
<th scope="col">Name</th>
<th scope="col">Age</th>
<th scope="col">Birthday</th>
</tr><tr>
<th scope="row">Jackie</th>
<td>5</td>
<td>April 5</td>
</tr><tr>
<th scope="row">Beth</th>
<td>8</td>
<td>January 14</td>
</tr></table>
```

The scope tag tells the browser and screen reader that everything under the column is related to the header at the top, and everything to the right of the row header is related to that header.

Associating Data Cells with the Appropriate Headers: id

The "headers" and "id" method should only be used when there is more than one logical level in a table, and when it is necessary to link more than two headers with a data cell. If we extend our original example, we can create a table that fits this criterion. In the table below, data have three headers each, so it is appropriate to use a more complex technique.

Shelly's Daughters			
	Name	Age	Birthday
by birth	Jackie	5	April 5
	Beth	8	January 14
by marriage	Jenny	12	Feb 12

The markup looks like this:

```
<table border="1">
<caption>Shelly's Daughters</caption>
<tr><td>&nbsp;</td>
<th id="name">Name</th>
<th id="age">Age</th>
<th id="birthday">Birthday</th>
</tr><tr>
<th rowspan="2" id="birth">by birth</th>
<th id="jackie">Jackie</th>
<td headers="birth jackie age">5</td>
<td headers="birth jackie birthday">April 5</td>
</tr><tr>
<th id="beth">Beth</th>
<td headers="birth beth age">8</td>
<td headers="birth beth birthday">January 14</td>
</tr><tr>
<th id="step">by marriage</th>
<th id="jenny">Jenny</th>
<td headers="step jenny age">12</td>
<td headers="step jenny birthday">Feb 12</td>
</tr></table>
```

Don't use this method in simple tables where the scope attribute will work. Beware: spanned rows and columns are not handled well by the JAWS screen reader. If at all possible, avoid complex data tables, or represent the data in a way that is less complex, preferably with no more than two headings applying to a single data cell.

The <caption> tag

Use the caption tag, right after the opening <table> tag, like this:

```
<table border="1" align="center">
<caption>Shelly's Daughters</caption>
```

It is not absolutely necessary to have caption tags on every data table for the sake of accessibility, but it is still a good practice.

Using CSS, you can also assign alignment to left or right along with the top and bottom alignments built into the caption element.

- `caption { font-size: smaller; text-align: center }`

A counter-example of the grey area between layout and data: the use of a single-cell table to hold a photograph. In that case, the use of the caption element is warranted; the caption genuinely is a caption in that instance.

The summary Element

`summary` is an attribute of the table element. Add it along with any other attributes, like `border`. For example:

```
<table align="center" border="0" cellpadding="5" cellspacing="3"
summary="Unemployment figures in Queensland in 1994">
```

`summary` should be succinct. The W3C says “When an appropriate markup language exists, use markup rather than images to convey information,” ... “Mark up documents with the proper structural elements.... [U]sing presentation markup rather than structural markup to convey structure (e.g., constructing what looks like a table of data with an HTML pre element) makes it difficult to render a page intelligibly to other devices.”

Summaries are easy to write. Just ask yourself “What is this table for?” and write that down in a concise sentence. If you can't, add a link to explanatory text or include it in the body of your page.

Use an "empty summary" for layout tables, `summary=" "`, to placate accessibility validators.

Handy Links

Contact me at jason_morningstar@unc.edu

Check out UNC's fabulous accessibility site: <http://www.unc.edu/webaccess>

WebAIM's table tutorial, from which, along with Joe Clark's [Building Accessible Websites](#), this document borrows: <http://www.webaim.org/techniques/tables/>