

HTML tables

From Web Education Community Group

Contents

- 1 Introduction
- 2 The most basic table
- 3 Adding some more features
- 4 Structuring the table further
- 5 CSS to the rescue: a better looking table
- 6 Summary
- 7 Further reading
- 8 Exercise questions

Introduction

“Ack!” — how do you use web standards to organize reams of data? The very idea of tons of nested elements needed to get all the data into nice little rows and boxes ought to put your brain into “Ack!” mode, but there is a solution — tables to the rescue!

In web design tables are a good way to organize data into a tabular form. In other words, think of tables, charts, and other information graphics that help you to see and process a great deal of information in a summary that allows you to compare and contrast different pieces of data. You see them all the time on websites, whether they are giving a summary or comparison of political election results, sports statistics, price comparisons, size charts, or other data.

Back in the Jurassic Age of the Internet before CSS was popularised as a method of separating the presentation from structure of the HTML, tables were used as a way to lay out web pages — to create columns, boxes, and generally arrange the content. This is the wrong way to go about it; tables for layout resulted in bloated, messy pages with tons of nested tables — larger file sizes, hard to maintain, hard to modify after the fact. You’ll still see sites like this on the Web, but rest assured that these days you should just use tables for what they are designed for — tabular data — and use CSS to control layout.

In this article we will cover how to use HTML tables properly.

The most basic table

Let's start with the semantic HTML code required to render a basic table — this particular example compares recent volcanic eruptions in the Pacific region of North America:

```
<table>
  <tr>
    <td>Volcano Name</td>
    <td>Location</td>
    <td>Last Major Eruption</td>
    <td>Type of Eruption</td>
  </tr>
  <tr>
    <td>Mt. Lassen</td>
    <td>California</td>
    <td>1914-17</td>
    <td>Explosive Eruption</td>
  </tr>
  <tr>
    <td>Mt. Hood</td>
    <td>Oregon</td>
    <td>1790s</td>
    <td>Pyroclastic flows and Mudflows</td>
  </tr>
  <tr>
    <td>Mt .St. Helens</td>
    <td>Washington</td>
    <td>1980</td>
    <td>Explosive Eruption</td>
  </tr>
</table>
```

This code renders roughly like so:

Volcano Name	Location	Last Major Eruption	Type of Eruption
Mt. Lassen	California	1914-17	Explosive Eruption
Mt. Hood	Oregon	1790s	Pyroclastic flows and Mudflows
Mt .St. Helens	Washington	1980	Explosive Eruption

Let's start by breaking down the HTML markup used in the above code:

- `<table></table>`: The `<table>` wrapper element is necessary to indicate to the browser that you wish to arrange the content in a tabular fashion.
- `<tr></tr>`: The `<tr>` element establishes a table row. This allows the browser to organize any content between the `<tr>` and `</tr>` tags in a horizontal fashion, or all in a row.
- `<td></td>`: The `<td>` element defines the table cell or each individual space for content within the row. Only use as many `<td>` table cells as needed for actual data. Don't use empty `<td>` cells for white space or padding — you use CSS to create any white space or padding needed, as this is not only a good way to separate presentation from structure; keeping the HTML clean and simple also makes the table easier to understand for people with visual impairments who are using screen readers to read the table contents out to them.

Note that the basic elements must be nested as follows:

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

To order them in another fashion will cause the browser to spit up the equivalent of an internet hair ball and render the table in a very odd fashion, if at all.

Adding some more features

Now the basic table is in place, we can add some slightly more complex table features—first, we'll add a caption and Table headers to help improve the data both in terms of semantics and legibility for screen readers. The updated table markup looks like so:

```
<table>
  <caption>Recent Major Volcanic Eruptions in the Pacific Northwest</caption>
  <tr>
    <th>Volcano Name</th>
    <th>Location</th>
    <th>Last Major Eruption</th>
    <th>Type of Eruption</th>
  </tr>
  <tr>
    <td>Mt. Lassen</td>
    <td>California</td>
    <td>1914-17</td>
    <td>Explosive Eruption</td>
  </tr>
  <tr>
    <td>Mt. Hood</td>
    <td>Oregon</td>
    <td>1790s</td>
    <td>Pyroclastic flows and Mudflows</td>
  </tr>
  <tr>
    <td>Mt. St. Helens</td>
    <td>Washington</td>
    <td>1980</td>
    <td>Explosive Eruption</td>
  </tr>
</table>
```

This code is rendered as:

Recent Major Volcanic Eruptions in the Pacific Northwest			
Volcano Name	Location	Last Major Eruption	Type of Eruption
Mt. Lassen	California	1914-17	Explosive Eruption

Mt. Hood	Oregon	1790s	Pyroclastic flows and Mudflows
Mt. St. Helens	Washington	1980	Explosive Eruption

The new elements used here are:

- `<caption></caption>`: The `<caption>` element allows you to give the table data a caption. Most browsers will center the caption and render it the same width as the table by default.
- `<th></th>`: The `<th>` element delineates the content between the tag as the table head titles for each column. This is useful not just to help semantically describe what the function of this content is, but it also helps render it more accurately in a variety of browsers and devices.

Structuring the table further

As a final step in structuring our table, we will define header and body table sections, add a footer and define the scope of row and column headings. We will also add a `summary` attribute to describe the table contents. The final markup looks like so:

```
<table summary="a summary of recent major volcanic eruptions in the Pacific Northwest">
  <caption>Recent Major Volcanic Eruptions in the Pacific Northwest</caption>
  <thead>
    <tr>
      <th scope="col">Volcano Name</th>
      <th scope="col">Location</th>
      <th scope="col">Last Major Eruption</th>
      <th scope="col">Type of Eruption</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td colspan="4">Compiled in 2008 by Ms Jen</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <th scope="row">Mt. Lassen</th>
      <td>California</td>
      <td>1914-17</td>
      <td>Explosive Eruption</td>
    </tr>
    <tr>
      <th scope="row">Mt. Hood</th>
      <td>Oregon</td>
      <td>1790s</td>
      <td>Pyroclastic flows and Mudflows</td>
    </tr>
    <tr>
      <th scope="row">Mt. St. Helens</th>
      <td>Washington</td>
      <td>1980</td>
      <td>Explosive Eruption</td>
    </tr>
  </tbody>
</table>
```

This table code looks like so in a browser:

Recent Major Volcanic Eruptions in the Pacific Northwest			
Volcano Name	Location	Last Major Eruption	Type of Eruption
Mt. Lassen	California	1914-17	Explosive Eruption
Mt. Hood	Oregon	1790s	Pyroclastic flows and Mudflows
Mt. St. Helens	Washington	1980	Explosive Eruption
Compiled in 2008 by Ms Jen			

The new elements/attributes are as follows:

- The `<thead>`, `<tbody>` and `<tfoot>` elements: These define the table's header, body and footer respectively. Unless you are coding a really complex table with many columns and rows of data, using these may be overkill. In a complex table however, using them can add useful structure for the developers, and also for browsers and other devices.
- The `colspan` and `rowspan` attributes: The `colspan` attribute creates a table cell that will span more than one column. In the case of the above footer, we wanted the one table cell to span the whole width of the table, thus we told it that it was to span four columns. Alternately, you can add a table cell `rowspan` attribute that will allow the table cell to span x amount of rows, for example `<td rowspan="3">`.
- The `summary` attribute: This attribute is used to define a summary of the table contents, for use by screenreaders (you won't

see it on the rendered version of the table above.) Note that in the older W3C recommendations, WCAG 1.0 and HTML 4.0, it says you can use the `summary` attribute as detailed above. However, the W3C Working draft now states "The `summary` attribute on the table element is **obsolete**. Consider describing the structure of the table in a caption element or in a figure element containing the table element; or, simplify the structure of the table so that no description is needed" (<http://www.w3.org/TR/html-markup/table.html>). Users of HTML5 will thus want to avoid this attribute.

- The `scope` attribute: You may also have noticed the `scope` attributes in the `th` tags, and the fact that we have defined the volcano names as headings too, inside the data rows! The `scope` attribute can be used in the `th` element to tell screen readers that the `th` content is the title for a column or a row.

CSS to the rescue: a better looking table

The above listed elements and attributes are all that is necessary to code a good data table. Now the HTML structure is in place, let's look at some simple CSS to make the table look a bit nicer:

```
body {
    background: #ffffff;
    margin: 0;
    padding: 20px;
    line-height: 1.4em;
    font-family: tahoma, arial, sans-serif;
    font-size: 62.5%;
}

table {
    width: 80%;
    margin: 0;
    background: #FFFFFF;
    border: 1px solid #333333;
    border-collapse: collapse;
}

td, th {
    border-bottom: 1px solid #333333;
    padding: 6px 16px;
    text-align: left;
}

th {
    background: #EEEEEE;
}

caption {
    background: #E0E0E0;
    margin: 0;
    border: 1px solid #333333;
    border-bottom: none;
    padding: 6px 16px;
    font-weight: bold;
}
```

When applied to our final table markup, the table looks as seen in Figure 1. You can also check out the example live here (<http://dev.opera.com/articles/view/19-html-tables/table3.html>) .

Recent Major Volcanic Eruptions in the Pacific Northwest			
Volcano Name	Location	Last Major Eruption	Type of Eruption
Mt. Lassen	California	1914-17	Explosive Eruption
Mt. Hood	Oregon	1790s	Pyroclastic flows and Mudflows
Mt. St. Helens	Washington	1980	Explosive Eruption
Compiled in 2008 by Ms Jen			

Figure 1: The table now looks a lot more visually appealing.

Oooh... Look, much better. You can choose to style the table anyway you want, but the above provides a baseline to work with. You'll learn a lot more about styling tables with CSS later in the course, but for now I'll just briefly break down what each section of this CSS is doing:

- `<body>`: In the above CSS, we have added properties to set the margin (to zero in this case), padding to give a little room, background colour (white), font size and family, as well as the line height to also give a little breathing room. You can download the code for this example here (<http://dev.opera.com/articles/view/19-html-tables/csstable.zip>) — try altering the property values in the CSS file to see how things change.
- `<table>`: borders have been added using the CSS `border` property. To make this work correctly, we also had to set the

`border-collapse` property to `collapse`, to reset the border values in the table and allow the `border-bottom` to be a straight rule line across the whole row rather than being broken up at the end of each table cell. A `width` of 80% was chosen for this example (this makes the table extend across 80% of the screen width).

- `<th>` and `<td>`: In the above example CSS, we have set `text-align` to be `left`, but you could set it to `center` or other values if you like. we also gave both the `<th>` and `<td>` a bit of padding to open up the rows and allow for greater readability. In the case of the `<th>` selector above, we set another color as to differentiate the headings from the rest of the table.
- `<caption>`: In the above example we have given the caption a border (with no bottom border, as the border in the table already provided it), a different background colour and bold type to set the caption apart from the table header row.

Summary

In this article we have gone through all you need to know to create effective HTML data tables. That's a wrap! Let's leave this with some pertinent thoughts:

- It is important that tables are correctly coded to be readable by a variety web browsers, mobile, accessible, and other devices. Table HTML is best kept to a minimum, and you should use CSS to style the tables. You'll learn a lot more about CSS later on in the course.
- Tables can be accessible to mobile devices and users that use screen reading software by keeping the code clean, using attributes such as `scope` and `summary` as well as the `<caption>` element to help announce clearly and semantically what the respective sections are for. (Note: the `summary` attribute is obsolete in HTML5.) Also important for accessibility is to not use empty table cells for spacing (use CSS for this instead).

Further reading

- W3C HTML 4 Tables Recommendation (<http://www.w3.org/TR/html401/struct/tables.html>)
- W3C CSS 2 tables recommendation (<http://www.w3.org/TR/CSS21/tables.html>)
- Roger Johansson's "Bring on the Tables" (http://www.456bereastreet.com/archive/200410/bring_on_the_tables/)

Exercise questions

- Start by coding a simple table with only the 3 main table elements: `table`, `tr`, and `td`. Save it and view it in a browser.
- Much like the second example above, add a caption, header, and footer to your table. How does that change what you see in the browser?
- What can you do to make your table more accessible to screen readers and mobile devices?
- Now create a CSS file. Try styling the borders, padding, and cell spacing of your table with only CSS and no attributes in your HTML markup. Add background colour and style the fonts.

Have fun.

Note: This material was originally published as part of the Opera Web Standards Curriculum, available as 19: HTML tables (<http://dev.opera.com/articles/view/19-html-tables/>), written by Jenifer Hanen. Like the original, it is published under the Creative Commons Attribution, Non Commercial - Share Alike 2.5 (<http://creativecommons.org/licenses/by-nc-sa/2.5/>) license.

Retrieved from "http://www.w3.org/community/webed/wiki/HTML_tables"

Categories: Tutorials | WSC | HTML

- This page was last modified on 23 May 2013, at 23:26.