Tables and HTML Validation

Table of Contents

[What are HTML tables?](#_gy0m75d5qmg2)

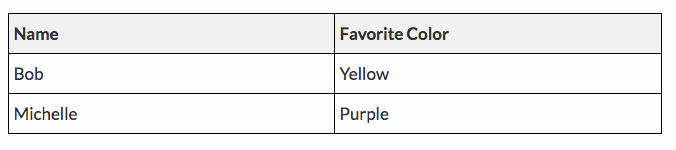
[Sectioning Tables](#_ankgj1romd0w)

[Table Styles, Zebra Striping, and Row Hovers](#_vyxmylayz6f4)

[HTML Validation](#_gwus2shwocga)

## What are HTML tables?

Tables are used in HTML documents (web pages) to present tabular data (rows and columns format).



Tables used to be used to layout entire webpages, but with the advancements of CSS, this is never (at least, *should* never be) done anymore.

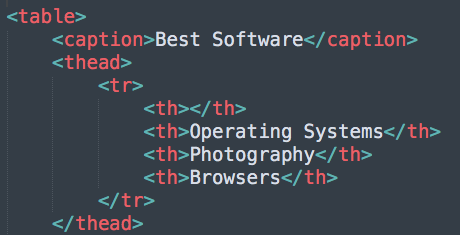
When creating tables, a good idea is to draw out the table first. Having a good idea of what the table is supposed to look like is a big help when you’re writing out your table markup.

## Sectioning Tables

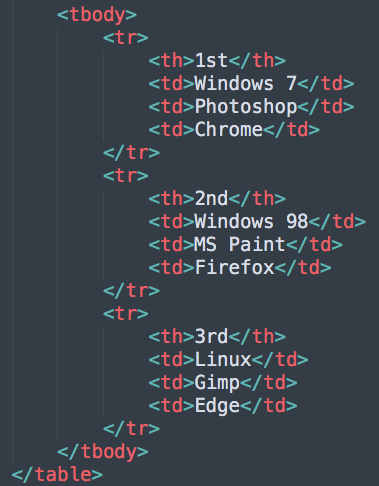
The **<table>** tag defines a table. Within it, there are three tags used to section a table into organized parts: **<thead>, <tfoot>, and <tbody>.** These three tags are not required, but they do make the table more semantic, they help with CSS selectors, and as we’ll see later on, they help in Javascript as well. **<caption>** tags are used in tables as headings that appear above the table.

Tables in HTML are made one row at a time. To create a new row, the **<tr>** tag is used. Within the table row cells are defined where content can be placed. To define a cell, the **<td>** (table data)tag is commonly used. For heading row cells, the **<th>** (table heading) tag can be used.

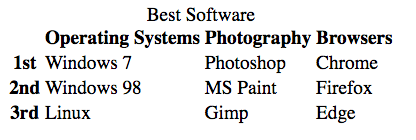
Here’s an example of creating a heading area in a table:



For the body of a table, the **<tbody>** tag is used. This is where the main table data will go.

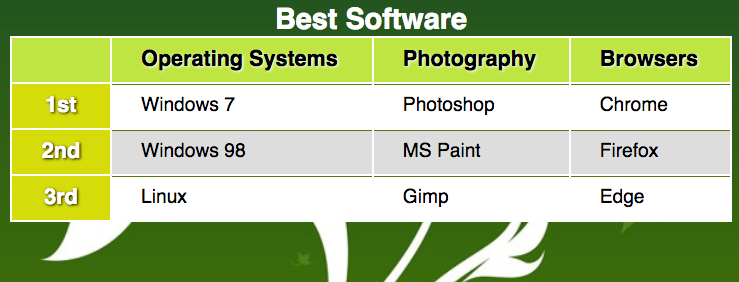


With no styles applied, this is what the table will look like in a browser:



## Table Styles, Zebra Striping, and Row Hovers

The table above only has three rows, but most tables have many rows. Making a table visually appealing is important. With well designed styles, users can easily see what row they’re focusing on, how many rows there are, and so on. Here’s what our table could look like:



Let’s see how these styles were created.



By adding two fonts, with a comma delimiter, a backup has been added. If a font isn’t located on a client computer, the browser will go to the next one. In this case the browser will use any sans-serif font, if Helvetica isn’t located.

A background image has been used as well. These can greatly improve the look of your webpages. But remember that text should *always* be on a flat background colour, like it is above. Text on top of an image is always hard to read.



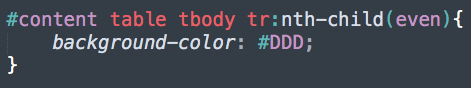
First off, notice the #content. This example has a table included in another tag that has an id value of “content”.

The styles are fairly straightforward, but one thing to always remember is that *selectors should be read right to left*. Take the last selector here, for example. We’re accessing all <th> tags, within a <thead> tag, inside a <table> that’s inside a tag with an id of “content”. These <th> styles should only be for <th> tags within a <thead>. There are <th> tags in the <tbody> tag, and we don’t want those styles to conflict. This is the advantage of using the <thead> and <tbody> tags.

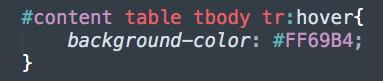


Now we have styles done for our <tbody> tag, with the <th> and <td> tags within it.

A common visual technique for tables with a lot of rows is to highlight every 2nd row a slightly different colour. This is referred to as *zebra-striping*. To do this, the nth-child pseudo-class can be used, with a passed in parameter of “even” or “odd”, depending on what rows you want to target.



Another technique that makes it easier for users to read tabular data is to highlight the row they are hovering over with their mouse. The :hover pseudo-class was used with the <a> tags that we’ve seen before. We can use :hover for any HTML tag, including <tr>.



## HTML Validation

There are many browsers and as such, webpages can look different in each one of them. One way to minimize this effect is to validate our HTML by a set of rules defined by the [W3C](https://www.w3.org/). Validating your markup is a sign of professionalism, it future proofs your markup, and controls the way your markup is interpreted by any browser.

The easiest way to validate your markup is to use the [W3C Validation Service](https://validator.w3.org/#validate_by_input). Simply copy your markup and paste it into the textarea on the W3C site.

If your webpage has errors (unclosed tags, missing required attributes, etc..) they will be documented. You can read explanations to why your code has errors and what you can do to fix them.