**Tutorial: Tables**

In this short tutorial, I'll walk you through creating and styling basic HTML tables. We use tables to display data on a page in rows and columns. We could use tables for a calendar, a schedule, or just some statistical data.

Let's start out with a blank HTML page and create a table.

To do that we use the `table` tag.

**Add the below bolded HTML into the body.**

**<table>**

**</table>**

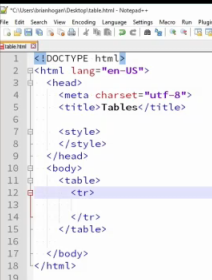
Tables are composed of rows and columns. To create a row, we use the `tr` tag.

**Inside the table tag add a tr tag.**

**<tr>**

**</tr>**

Notice that the `tr` tag is indented over, as it's a child of the `table.

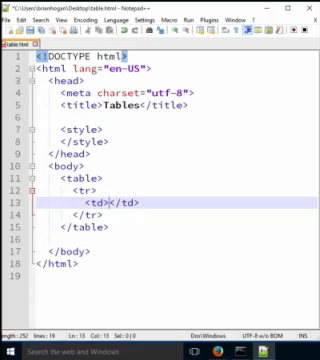


This creates a row, and now we need to create a "table cell" by using the `td` tag, which stands for `table data`.

**Inside the tr tag add a td tag.**

**<td>row one col one</td>**

Note that once again, this is indented because it's a child of the table row.



To add more columns to the row, we just add another table data.

**Inside the tr tag add another td tag.**

**<td>row one col two</td>**

The new td is not indented though. It's a sibling of the other table row.

**Inside the table tag add a second tr tag.**

To add another row, we simply create the same structure again.

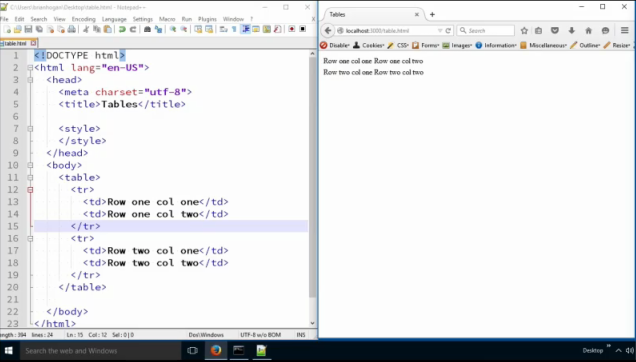
**<tr>**

**<td>row two col two</td>**

**<td>row two col two</td>**

**</tr>**

**Save index.html and refresh the page.**



Now, to improve accessibility, we should add column headers. We do that using another table row at the top of the table, but instead of using `td` tags, we use `th` tags, for `table header`.

**Inside the table tag add column headers at the top of the table tag.**

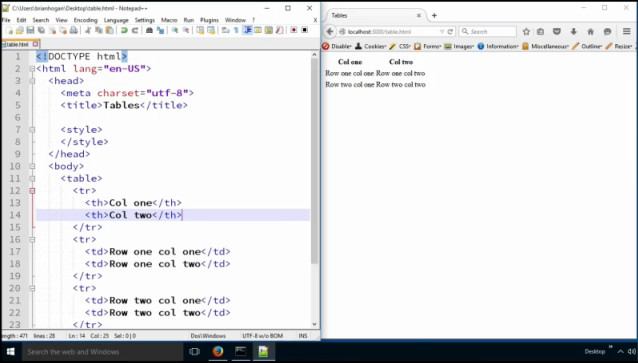
**<tr>**

**<th>col one</th>**

**<th>col two</th>**

**</tr>**

**Save index.html and refresh the page.**



We should use the `thead` and `tbody` elements too.

**Add the bolded tags inside the table tag. You will have to indent the child tags by selecting those new children rows and pressing the tab key.**

<table>

**<thead>**

<tr>

<th>col one</th>

<th>col two</th>

</tr>

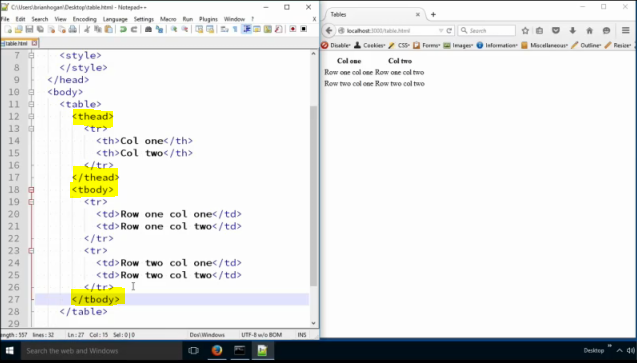
**</thead>**

**<tbody>**

….

**</tbody>**

</table>

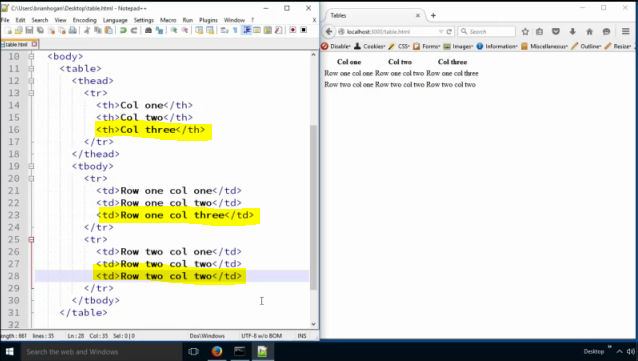


Our table is valid without these, but you'll find you'll need them eventually. Many libraries you'll use will require them. In addition, the browser puts them in for you if you don't, and we really don't want the browser adding code for us. So, we'll use them, and you should use them in this class.

So now, how do you add another row? Just add one on to the table. And how do you add a third column? you just insert another `<td>` to each row.

**Add the highlight th and td(s) to the table to produce the 3rd column.**

**Save the index.html file and refresh.**



We can also make a column stretch across the table. Let's add a new row.

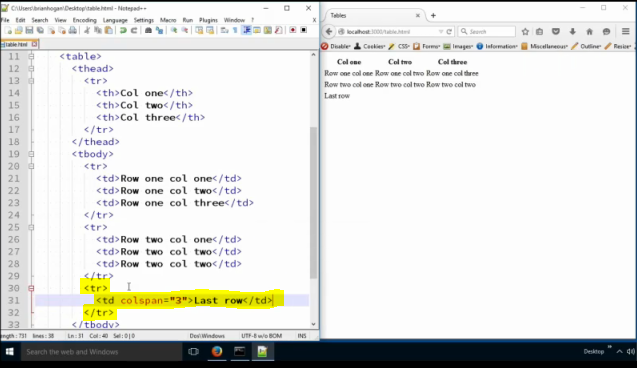
**Add the following to the table tag under the last tr.**

**<tr>**

**<td colspan="3">Last row</td>**

**</tr>**

**Save the index.html page and refresh.**



The `colspan` attribute makes the cell stretch across the number of columns we have.

**Styling**

Let's look at how we style tables by using some more advanced CSS.

First, let's set the width of the table and set a border.

**Add the following bolded CSS to a styles tag in the head of the document. Add the style tag first to the head.**

**table {**

**border: 1px solid #000;**

**border-collapse: collapse;**

**width: 600px;**

**}**

By default, the web browser styles tables so each cell has its own borders. But by setting the border-collapse property to "collapse" then adjacent cells will share borders.

Next, we can style the headers.

**th {**

**background-color: #000;**

**color: #FFF;**

**text-align: left;**

**}**

We'll apply a background color and a foreground color and align the text to the left.

Next, just for fun, let's right align the last column. we can use the pseudo class last-child to get the last element of a group. Kind of a goofy name though. It’s not looking for a child element, it’s just looking at the group and grabbing the last one.

**td:last-child, th:last-child {**

**text-align: right;**

**}**

We target both the `td` and the `th` elements because we want the header to be adjusted too.

Finally let's stripe the rows of the table. We'll make every other row a different color. The nth-child selector works great here. There’s a complex formula we can use to select specific child elements but we won’t cover that in this class. Look it up in the spec. We can however use keywords.

**tr:nth-child(even) {**

**background-color: #F3F3F3;**

**}**

And we can do the same for the odd rows:

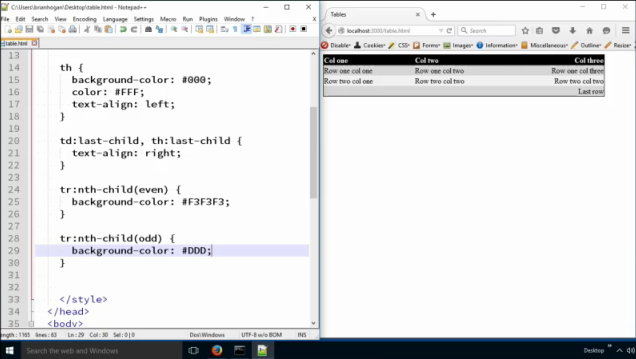
**tr:nth-child(odd) {**

**background-color: #DDD;**

**}**

And that does it for our tables.

**Save the index.html file and refresh the page.**



Tables are designed for tabular data. Years ago, they used to be used to lay out a web page. But those days are long gone because they make it very difficult for people who use screen readers to access the page content in a meaningful way. If you have some data you need to show, a table is a very good fit though. And now you know how to make them, and style them.

**Below is the index.html:**

<!DOCTYPE html>

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>Tables</title>

<style>

table {

border: 1px solid #000;

border-collapse: collapse;

width: 600px;

}

th {

background-color: #000;

color: #FFF;

text-align: left;

}

td:last-child, th:last-child {

text-align: right;

}

tr:nth-child(even) {

background-color: #F3F3F3;

}

tr:nth-child(odd) {

background-color: #DDD;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>col one</th>

<th>col two</th>

<th>col three</th>

</tr>

</thead>

<tbody>

<tr>

<td>row one col one</td>

<td>row one col two</td>

<td>row one col three</td>

</tr>

<tr>

<td>row two col one</td>

<td>row two col two</td>

<td>row two col three</td>

</tr>

<tr>

<td>row three col one</td>

<td>row three col two</td>

<td>row three col three</td>

</tr>

</tbody>

</table>

</body>

</html>