Tables

**Why Tables?**

There are many websites on the Internet that display information like stock prices, sports scores, invoice data, and more. This data is naturally tabular in nature, meaning that a table is often the best way of presenting the days.

In this part of the course, you'll learn how to use HTML to present tabular data to users.

Let's begin!

**Create a Table**

Before displaying data, you must first create the table that will contain the data by using the <table> element.

<table>

</table>

The <table> element will contain all of the tabular data you plan on displaying.

# Table Rows

In many programs that use tables, the table is already predefined for you, meaning that it contains the [rows, columns, and cells that will hold data](https://en.wikipedia.org/wiki/Table_%28information%29). In HTML, all of these components must be created.

The first step in entering data into the table is to add rows using the table row element: <tr>.

<table>

<tr>

</tr>

<tr>

</tr>

</table>

In the example above, two rows have been added to the table.

# Table Data

Rows aren't sufficient to add data to a table. Each cell element must also be defined. In HTML, you can add data using the table data element: <td>.

<table>

<tr>

<td>73</td>

<td>81</td>

</tr>

</table>

In the example above, two data points (73 and 81) were entered in the one row that exists. By adding two data points, we created two cells of data.

If the table were displayed in the browser, it would show a table with one row and two columns.

**Table Headings**

Table data doesn't make much sense without titles to describe what the data represents.

To add titles to rows and columns, you can use the *table heading* element: <th>.

The table heading element is used just like a table data element, except with a relevant title. Just like table data, a table heading must be placed within a table row.

<table>

<tr>

<th></th>

<th scope="col">Saturday</th>

<th scope="col">Sunday</th>

</tr>

<tr>

<th scope="row">Temperature</th>

<td>73</td>

<td>81</td>

</tr>

</table>

What happened in the code above?

First, a new row was added to hold the three headings: a blank heading, a Saturday heading, and a Sunday heading. The blank heading creates the extra table cell necessary to align the table headings correctly over the data they correspond to.

In the second row, one table heading was added as a row title: Temperature.

Note, also, the use of the scope attribute, which can take one of two values:

1. row - this value makes it clear that the heading is for a row.
2. col - this value makes it clear that the heading is for a column.

HTML code for tables may look a little strange at first, but analyzing it piece by piece helps make the code more understandable.

# Table Borders

So far, the tables you've created have been a little difficult to read because they have no borders.

In older versions of HTML, a border could be added to a table using the border attribute and setting it equal to an integer. This integer would represent the thickness of the border.

<table border="1">

<tr>

<td>73</td>

<td>81</td>

</tr>

</table>

The code in the example above is following is [deprecated](https://en.wikipedia.org/wiki/Deprecation), so please don't use it. It's meant to illustrate older conventions you may come across when reading other developers' code.

The browser will likely still interpret your code correct if you use the border attribute, but that doesn't mean the attribute should be used. Instead, you can achieve the same effect using CSS.

table, td {

border: 1px solid black;

}

The code in the example above uses CSS instead of HTML to show table borders.

# Spanning Columns

What if the table contains data that spans multiple columns?

For example, a personal calendar could have events that span across multiple hours, or even multiple days.

Data can span columns using the colspan attribute. The attributes accepts an integer (greater than or equal to 1) to denote the number of columns it spans across.

<table>

<tr>

<th>Monday</th>

<th>Tuesday</th>

<th>Wednesday</th>

</tr>

<tr>

<td colspan="2">Out of Town</td>

<td>Back in Town</td>

</tr>

</table>

In the example above, the data Out of Town spans the Monday and Tuesday table headings using the value 2 (two columns). The data Back in Town appear only under the Wednesday heading.

**Spanning Rows**

Data can also span multiple rows using the rowspan attribute.

The rowspan attribute is used for data that spans multiple rows (perhaps an event goes on for multiple hours on a certain day). It accepts an integer (greater than or equal to 1) to denote the number of rows it spans across.

<table>

<tr> <!-- Row 1 -->

<th></th>

<th>Saturday</th>

<th>Sunday</th>

</tr>

<tr> <!-- Row 2 -->

<th>Morning</th>

<td rowspan="2">Work</td>

<td rowspan="3">Relax</td>

</tr>

<tr> <!-- Row 3 -->

<th>Afternoon</th>

</tr>

<tr> <!-- Row 4 -->

<th>Evening</th>

<td>Dinner</td>

</tr>

</table>

In the example above, there are four rows:

1. The first row contains an empty cell and the two column headings.
2. The second row contains the Morning row heading, along with Work, which spans two rows under the Saturday column. The "Relax" entry spans three rows under the Sunday column.
3. The third row only contains the Afternoon row heading.
4. The fourth row only contains the Dinner entry, since "Relax" spans into the cell next to it.

If you'd like to see how the browser interprets the code above, feel free to copy and paste it into the code editor to understand it a little better.

# Table Body

Over time, a table can grow to contain a lot of data and become very long. When this happens, the table can be sectioned off so that it is easier to manage.

Long tables can be sectioned off using the table body element: <tbody>.

The <tbody> element should contain the all of the table's data, excluding the table headings (more on this in a later exercise).

<table>

<tbody>

<tr>

<th></th>

<th>Saturday</th>

<th>Sunday</th>

</tr>

<tr>

<th>Morning</th>

<td rowspan="2">Work</td>

<td rowspan="3">Relax</td>

</tr>

<tr>

<th>Afternoon</th>

</tr>

<tr>

<th>Evening</th>

<td>Dinner</td>

</tr>

</tbody>

</table>

In the example above, all of the table data is contained within a table body element. Note, however, that the headings were also kept in the table's body — we'll change this in the next exercise.

# Table Head

In the last exercise, the table's headings were kept inside of the table's body. When a table's body is sectioned off, however, it also makes sense to section off the table's headings using the <thead> element.

<table>

<thead>

<tr>

<th></th>

<th>Saturday</th>

<th>Sunday</th>

</tr>

</thead>

<tbody>

<tr>

<th>Morning</th>

<td rowspan="2">Work</td>

<td rowspan="3">Relax</td>

</tr>

<tr>

<th>Afternoon</th>

</tr>

<tr>

<th>Evening</th>

<td>Dinner</td>

</tr>

</tbody>

</table>

In the example above, the only new element is <thead>. The table headings are contained inside of this element. Note that the table's head still requires a row in order to contain the table headings.

# Table Footer

The bottom part of a long table can also be sectioned off using the <tfoot> element.

<table>

<thead>

<tr>

<th>Quarter</th>

<th>Revenue</th>

<th>Costs</th>

</tr>

</thead>

<tbody>

<tr>

<th>Q1</th>

<td>$10M</td>

<td>$7.5M</td>

</tr>

<tr>

<th>Q2</th>

<td>$12M</td>

<td>$5M</td>

</tr>

</tbody>

<tfoot>

<tr>

<th>Total</th>

<td>$22M</td>

<td>$12.5M</td>

</tr>

</tfoot>

</table>

In the example above, the footer contains the totals of the data in the table. Footers are often used to contain sums, differences, and other data results.

# Styling with CSS

Tables, by default, are very bland. They have no borders, the font color is black, and the typeface is the same type used for other HTML elements.

You can use CSS to style tables just like you have done in the past. Specifically, you can change style the various aspects mentioned above.

table, th, td {

border: 1px solid black;

font-family: Arial, sans-serif;

text-align: center;

}

The code in the example above demonstrates just some of the various table aspects you can style using the CSS properties you learned about earlier.