



Web Development

HTML Tables & CSS



HTML Table

An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.

Note: The **<td>** elements are the data containers of the table.

They can contain all sorts of HTML elements; text, images, lists, other tables, etc.



HTML Table

```
1 <table style="width:100%">
2   <tr>
3     <th>Firstname</th>
4     <th>Lastname</th>
5     <th>Age</th>
6   </tr>
7   <tr>
8     <td>Jill</td>
9     <td>Smith</td>
10    <td>50</td>
11  </tr>
12  <tr>
13    <td>Eve</td>
14    <td>Jackson</td>
15    <td>94</td>
16  </tr>
17 </table>
```



Colspan attribute

Cells that Span Many Columns. To make a cell span more than one column, use the **colspan** attribute:

```
1 <table style="width:100%">
2   <tr>
3     <th>Name</th>
4     <th colspan="2">Telephone</th>
5   </tr>
6   <tr>
7     <td>Bill Gates</td>
8     <td>55577854</td>
9     <td>55577855</td>
10  </tr>
11 </table>
```



Rowspan attribute

Cells that Span Many Rows. To make a cell span more than one row, use the **rowspan** attribute:

```
1 <table style="width:100%">
2   <tr>
3     <th>Name:</th>
4     <td>Bill Gates</td>
5   </tr>
6   <tr>
7     <th rowspan="2">Telephone:</th>
8     <td>55577854</td>
9   </tr>
10  <tr>
11    <td>55577855</td>
12  </tr>
13 </table>
```



HTML Table - Border

If you do not specify a border for the table, it will be displayed without borders.

```
1 table, th, td {  
2     border: 1px solid black;  
3 }
```



HTML Table - Collapsed border

If you want the borders to collapse into one border, add the CSS **border-collapse** property:

```
1 table, th, td {  
2     border: 1px solid black;  
3     border-collapse: collapse;  
4 }
```



HTML Table - Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

```
1 th, td {  
2   padding: 5px 15px;  
3 }
```




HTML Summary

- Use the HTML **<table>** element to define a table
- Use the HTML **<tr>** element to define a table row
- Use the HTML **<td>** element to define a table data
- Use the HTML **<th>** element to define a table heading
- Use the HTML **<caption>** element to define a table caption
- Use the CSS **border** property to define a border
- Use the CSS **border-collapse** property to collapse cell borders
- Use the CSS **padding** property to add padding to cells
- Use the CSS **text-align** property to align cell text
- Use the CSS **border-spacing** property to set the spacing between cells
- Use the colspan attribute to make a cell span many columns
- Use the rowspan attribute to make a cell span many rows
- Use the id attribute to uniquely define one table



HTML Table

Demo



HTML Table

Exercise Time!

Take a look at `table-exercise1.html`



HTML Table

Exercise 2

Create a table with several columns and rows. Make cells that contain numbers and texts.
Make your table stunning. Use padding, rowspan, colspan, background color, font color, everything and anything you want to make the table well built.



CSS Links

With CSS, links can be styled in different ways.

[Text Link](#)

[Text Link](#)

Link Button

Link Button

Links can be styled with any CSS property (e.g. color, font-family, background, etc.). In addition, links can be styled differently depending on what **state** they are in.

- **a:link** - a normal, unvisited link
- **a:visited** - a link the user has visited
- **a:hover** - a link when the user mouses over it
- **a:active** - a link the moment it is clicked



Exercise

Combine several CSS properties to display links as boxes/buttons.



Link Button

<normal>



Link Button

<hover>



CSS Layout - The display Property

The **display** property is the most important CSS property for controlling layout.

The **display** property specifies **if/how** an element is displayed.

Every HTML element has a default display value depending on what type of element it is.

The default display value for most elements is **block** or **inline**.



Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The `<div>` element is a block-level element.

Examples of block-level elements:

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`
- `<header>`
- `<footer>`
- `<section>`



Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline `` element inside a paragraph.

Examples of inline elements:

- ``
- `<a>`
- ``



Display Value

`display: none;` is commonly used with JavaScript to **hide** and **show** elements without deleting and recreating them.

Even though every element has a default display value, you can **override** this.

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

A common example is making inline `` elements for horizontal menus:



Hide an Element (`display:none` or `visibility:hidden`)

Hiding an element can be done by setting the `display` property to `none`. The element will be hidden, and the page will be displayed as if the element is not there.

`visibility:hidden;` also hides an element.

However, the element will still take up the same space as before. The element will be hidden, but still affect the layout.



CSS Layout - width and max-width

Difference demo

```
1 div.ex1 {  
2     width: 500px;  
3     margin: auto;  
4     border: 3px solid #73AD21;  
5 }  
6  
7 div.ex2 {  
8     max-width: 500px;  
9     margin: auto;  
10    border: 3px solid #73AD21;  
11 }
```



Resources

- [W3Schools](#)