



CSS Layout - The display Property

[< Previous](#)[Next >](#)

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

The display Property

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**.

Click to show panel

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: none;

`display: none;` is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The `<script>` element uses `display: none;` as default.

Override The Default Display Value

As mentioned, every element has a default display value. However, 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:

Example

```
li {  
  display: inline;  
}
```

Try it Yourself »

Note: Setting the display property of an element only changes **how the element is displayed**, NOT what kind of element it is. So, an inline element with `display: block;` is not allowed to have other block elements inside it.

The following example displays `` elements as block elements:

Example

```
span {  
  display: block;  
}
```

Try it Yourself »


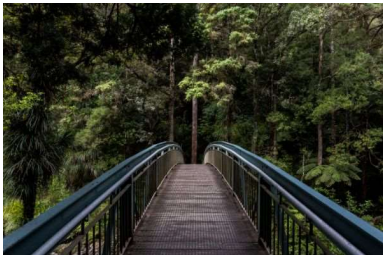

The following example displays `<a>` elements as block elements:

Example

```
a {  
  display: block;  
}
```

[Try it Yourself »](#)

Hide an Element - display:none or visibility:hidden?

| | | |
|--|---|--|
| <p>display:none</p>  <p>Remove</p> | <p>visibility:hidden</p>  <p>Hide</p> | <p>Reset</p>  <p>Reset All</p> |
|--|---|--|

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:

Example

```
h1.hidden {  
  display: none;  
}
```

[Try it Yourself »](#)

`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:

Example

```
h1.hidden {  
  visibility: hidden;  
}
```

[Try it Yourself »](#)

More Examples

[Differences between display: none; and visibility: hidden;](#)

This example demonstrates display: none; versus visibility: hidden;

[Using CSS together with JavaScript to show content](#)

This example demonstrates how to use CSS and JavaScript to show an element on click.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

CSS Display/Visibility Properties

| Property | Description |
|-----------------------------------|---|
| <u>display</u> | Specifies how an element should be displayed |
| <u>visibility</u> | Specifies whether or not an element should be visible |

[< Previous](#)

[Next >](#)



COLOR PICKER



SHOP

HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

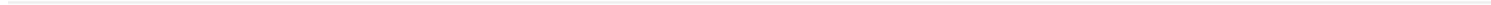
SHARE



CERTIFICATES

HTML
CSS
JavaScript
SQL
Python
PHP
jQuery
Bootstrap
XML

[Read More »](#)





[REPORT ERROR](#)

[FORUM](#)

[ABOUT](#)

[SHOP](#)

Top Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)
[JavaScript Tutorial](#)
[How To Tutorial](#)
[SQL Tutorial](#)
[Python Tutorial](#)
[W3.CSS Tutorial](#)
[Bootstrap Tutorial](#)
[PHP Tutorial](#)
[jQuery Tutorial](#)
[Java Tutorial](#)
[C++ Tutorial](#)

Top References

[HTML Reference](#)
[CSS Reference](#)
[JavaScript Reference](#)
[SQL Reference](#)
[Python Reference](#)
[W3.CSS Reference](#)
[Bootstrap Reference](#)
[PHP Reference](#)

[HTML Colors](#)[jQuery Reference](#)[Java Reference](#)[Angular Reference](#)

Top Examples

[HTML Examples](#)[CSS Examples](#)[JavaScript Examples](#)[How To Examples](#)[SQL Examples](#)[Python Examples](#)[W3.CSS Examples](#)[Bootstrap Examples](#)[PHP Examples](#)[jQuery Examples](#)[Java Examples](#)[XML Examples](#)

Web Certificates

[HTML Certificate](#)[CSS Certificate](#)[JavaScript Certificate](#)[SQL Certificate](#)[Python Certificate](#)[jQuery Certificate](#)[PHP Certificate](#)[Bootstrap Certificate](#)[XML Certificate](#)[Get Certified »](#)

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2020 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

