

CSS Tutorial

[W3Schools Home](#)

[Next Chapter](#)



Save a lot of work with CSS!

In our CSS tutorial you will learn how to use CSS to control the style and layout of multiple Web pages all at once.

Examples in Each Chapter

This CSS tutorial contains hundreds of CSS examples.

With our online editor, you can edit the CSS, and click on a button to view the result.

CSS Example

```
body {  
    background-color: #d0e4fe;  
}  
  
h1 {  
    color: orange;  
    text-align: center;  
}  
  
p {  
    font-family: "Times New Roman";  
    font-size: 20px;  
}
```

[Try it yourself »](#)

Click on the "Try it Yourself" button to see how it works.

CSS Examples

Learn from 200 examples! With our editor, you can edit the CSS, and click on a button to view the result.

[Try-It-Yourself!](#)

CSS Quiz Test

Test your CSS skills at W3Schools!

[Start CSS Quiz!](#)

CSS References

At W3Schools you will find complete CSS references of all properties and selectors with syntax, examples, browser support, and more.

[CSS Properties Reference](#)

[CSS Selectors Reference](#)

[CSS Aural Reference](#)

[CSS Units](#)

[CSS Color Reference](#)

Exam - Get Your Diploma!



W3Schools' Online Certification

The perfect solution for professionals who need to balance work, family, and career building.

More than 10 000 certificates already issued!

[**Get Your Certificate »**](#)

The [HTML Certificate](#) documents your knowledge of HTML.

The [HTML5 Certificate](#) documents your knowledge of advanced HTML5.

The [CSS Certificate](#) documents your knowledge of advanced CSS.

The [JavaScript Certificate](#) documents your knowledge of JavaScript and HTML DOM.

The [jQuery Certificate](#) documents your knowledge of jQuery.

The [PHP Certificate](#) documents your knowledge of PHP and SQL (MySQL).

The [XML Certificate](#) documents your knowledge of XML, XML DOM and XSLT.

 [W3Schools Home](#)

[Next Chapter](#) 

CSS Introduction

[◀ Previous](#)[Next Chapter ▶](#)

What You Should Already Know

Before you continue you should have a basic understanding of the following:

- HTML

If you want to study this subject first, find the tutorial on our [Home page](#).

CSS Demo - One Page - Multiple Styles!

One HTML page displayed with different style sheets: [See how it works!](#)

What is CSS?

- **CSS** stands for **Cascading Style Sheets**
- CSS defines **how HTML elements are to be displayed**
- Styles were added to HTML 4.0 **to solve a problem**
- CSS saves a lot of work
- External Style Sheets are stored in **CSS files**

CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a document.

HTML was intended to **define the content** of a document, like:

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

When tags like ``, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

In HTML 4.0, all formatting could (and should!) be removed from the HTML document, and stored in a separate CSS file.

CSS Saves a Lot of Work!

The style definitions are normally saved in external .css files.

With an external style sheet file, you can change the look of an entire Web site by changing just one file!

 Previous

Next Chapter 

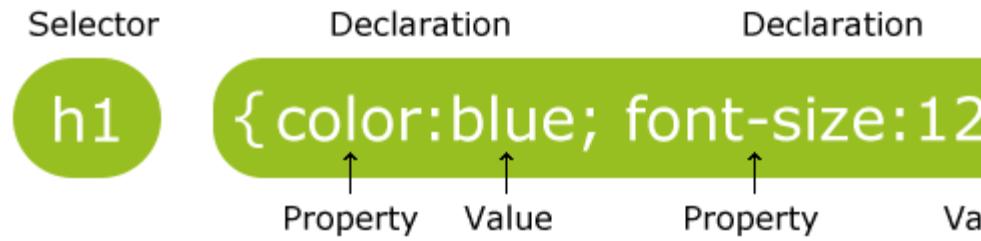
CSS Syntax

[Previous](#)

 Watch video of this tutorial 
[Next Chapter](#)

CSS Syntax

A CSS rule set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a property name and a value, separated by a colon.

CSS Example

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:

```
p {color:red;text-align:center;}
```

To make the CSS code more readable, you can put one declaration on each line.

In the following example all <p> elements will be center-aligned, with a red text color:

Example

```
p {
  color: red;
  text-align: center;
}
```

[Try it yourself »](#)

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

Example

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}  
  
/* This is  
a multi-line  
comment */
```

[Try it yourself »](#)[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

CSS Selectors

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

CSS Selectors

CSS selectors allow you to select and manipulate HTML elements.

CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

The element Selector

The element selector selects elements based on the element name.

You can select all `<p>` elements on a page like this: (all `<p>` elements will be center-aligned, with a red text color)

Example

```
p {  
    text-align: center;  
    color: red;  
}
```

[Try it yourself »](#)

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

An id should be unique within a page, so the id selector is used if you want to select a single, unique element.

To select an element with a specific id, write a hash character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

Example

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

[Try it yourself »](#)



Do **NOT** start an ID name with a number!

The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period character, followed by the name of the class:

In the example below, all HTML elements with class="center" will be center-aligned:

Example

```
.center {  
    text-align: center;  
    color: red;  
}
```

[Try it yourself »](#)

You can also specify that only specific HTML elements should be affected by a class.

In the example below, all <p> elements with class="center" will be center-aligned:

Example

```
p.center {  
    text-align: center;  
    color: red;  
}
```

[Try it yourself »](#)

Do **NOT** start a class name with a number!



Grouping Selectors

If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```

you can group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

[Try it yourself »](#)

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

[◀ Previous](#)

Watch video of this tutorial

[Next Chapter ▶](#)

CSS How To...

[Previous](#)Watch video of this tutorial [Next Chapter](#) 

When a browser reads a style sheet, it will format the document according to the information in the style sheet.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
 - Internal style sheet
 - Inline style
-

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the `<link>` tag. The `<link>` tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file called "myStyle.css", is shown below:

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
```

```
    margin-left: 20px;
}
```



Do not add a space between the property value and the unit (such as margin-left: 20 px;). The correct way is: margin-left: 20px;

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the `<style>` tag, like this:

Example

```
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

[Try it yourself »](#)

Inline Styles

An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!

To use inline styles, add the `style` attribute to the relevant tag. The `style` attribute can contain any CSS property. The example shows how to change the color and the left margin of a `h1` element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

[Try it yourself »](#)

Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, assume that an external style sheet has the following properties for the <h1> element:

```
h1 {
    color: navy;
    margin-left: 20px;
}
```

then, assume that an internal style sheet also has the following property for the <h1> element:

```
h1 {
    color: orange;
}
```

If the page with the internal style sheet also links to the external style sheet the properties for the <h1> element will be:

```
color: orange;
margin-left: 20px;
```

Try it yourself »

The left margin is inherited from the external style sheet and the color is replaced by the internal style sheet.

Multiple Styles Will Cascade into One

Styles can be specified:

- inside an HTML element
- inside the <head> section of an HTML page
- in an external CSS file

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

Cascading order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

Try it yourself »

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).



Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

Test Yourself with Exercises!

[Exercise 1 »](#)[Exercise 2 »](#)[Exercise 3 »](#)[Exercise 4 »](#)[« Previous](#)[Watch video of this tutorial](#)[Next Chapter »](#)

CSS Background

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position



Background Color

The background-color property specifies the background color of an element.

The background color of a page is set like this:

Example

```
body {  
    background-color: #b0c4de;  
}
```

[Try it yourself »](#)

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

Look at [CSS Color Values](#) for a complete list of possible color values.

In the example below, the `<h1>`, `<p>`, and `<div>` elements have different background colors:

Example

```
h1 {  
    background-color: #6495ed;  
}  
  
p {  
    background-color: #e0ffff;  
}  
  
div {  
    background-color: #b0c4de;  
}
```

[Try it yourself »](#)

Background Image

The `background-image` property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

Example

```
body {  
    background-image: url("paper.gif");  
}
```

[Try it yourself »](#)

Below is an example of a bad combination of text and background image. The text is almost not readable:

Example

```
body {  
    background-image: url("bgdesert.jpg");  
}
```

[Try it yourself »](#)

Background Image - Repeat Horizontally or Vertically

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Example

```
body {  
    background-image: url("gradient_bg.png");  
}
```

[Try it yourself »](#)

If the image is repeated only horizontally (repeat-x), the background will look better:

Example

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

[Try it yourself »](#)



Note: To repeat an image vertically set background-repeat: repeat-y;

Background Image - Set position and no-repeat



Note: When using a background image, use an image that does not disturb the text.

Showing the image only once is specified by the background-repeat property:

Example

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
}
```

[Try it yourself »](#)

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

Example

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

[Try it yourself »](#)

Background - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.

The shorthand property for background is simply "background":

Example

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

[Try it yourself »](#)

When using the shorthand property the order of the property values is:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

It does not matter if one of the property values is missing, as long as the ones that are present are in this order.

This example uses more advanced CSS. Take a look: [Advanced example](#)



More Examples

How to set a fixed background image

This example demonstrates how to set a fixed background image. The image will not scroll with the rest of the page.

Test Yourself with Exercises!

[Exercise 1 »](#)
[Exercise 2 »](#)
[Exercise 3 »](#)
[Exercise 4 »](#)
[Exercise 5 »](#)

All CSS Background Properties

Property	Description
<u>background</u>	Sets all the background properties in one declaration
<u>background-attachment</u>	Sets whether a background image is fixed or scrolls with the rest of the page
<u>background-color</u>	Sets the background color of an element
<u>background-image</u>	Sets the background image for an element
<u>background-position</u>	Sets the starting position of a background image
<u>background-repeat</u>	Sets how a background image will be repeated

[◀ Previous](#)
[Watch video](#) of this tutorial

[Next Chapter ▶](#)
[WEB HOSTING](#)
[UK Reseller Hosting](#)
[WEB BUILDING](#)
[FREE Website BUILDER](#)
[Free HTML5 Templates](#)

W3SCHOOLS EXAMS

HTML, CSS, JavaScript, PHP, jQuery, and XML Certifications

SHARE THIS PAGE



COLOR PICKER



CSS Text

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

TEXT FORMATTING

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from the "Try it yourself" link.

Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

Look at [CSS Color Values](#) for a complete list of possible color values.

The default color for a page is defined in the body selector.

Example

```
body {  
    color: blue;  
}  
  
h1 {  
    color: #00ffff;  
}  
  
h2 {
```

```
    color: rgb(255,0,0);
}
```

[Try it yourself »](#)

 **Note:** For W3C compliant CSS: If you define the color property, you must also define the background-color property.

Text Alignment

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

Example

```
h1 {
    text-align: center;
}

p.date {
    text-align: right;
}

p.main {
    text-align: justify;
}
```

[Try it yourself »](#)

Text Decoration

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

Example

```
a {
    text-decoration: none;
}
```

[Try it yourself »](#)

It can also be used to decorate text:

Example

```
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}
```

[Try it yourself »](#)



Note: It is not recommended to underline text that is not a link, as this often confuses users.

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Example

```
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}
```

[Try it yourself »](#)

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text.

Example

```
p {  
    text-indent: 50px;  
}
```

[Try it yourself »](#)



More Examples

Specify the space between characters

This example demonstrates how to increase or decrease the space between characters.

Specify the space between lines

This example demonstrates how to specify the space between the lines in a paragraph.

Set the text direction of an element

This example demonstrates how to change the text direction of an element.

Increase the white space between words

This example demonstrates how to increase the white space between words in a paragraph.

Disable text wrapping inside an element

This example demonstrates how to disable text wrapping inside an element.

Vertical alignment of an image

This example demonstrates how to set the vertical align of an image in a text.

Add shadow to text

This example demonstrates how to add shadow to text.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

[Exercise 5 »](#)

All CSS Text Properties

Property	Description
<u>color</u>	Sets the color of text
<u>direction</u>	Specifies the text direction/writing direction

<u>letter-spacing</u>	Increases or decreases the space between characters in a text
<u>line-height</u>	Sets the line height
<u>text-align</u>	Specifies the horizontal alignment of text
<u>text-decoration</u>	Specifies the decoration added to text
<u>text-indent</u>	Specifies the indentation of the first line in a text-block
<u>text-shadow</u>	Specifies the shadow effect added to text
<u>text-transform</u>	Controls the capitalization of text
<u>unicode-bidi</u>	Used together with the <u>direction</u> property to set or return whether the text should be overridden to support multiple languages in the same document
<u>vertical-align</u>	Sets the vertical alignment of an element
<u>white-space</u>	Specifies how white-space inside an element is handled
<u>word-spacing</u>	Increases or decreases the space between words in a text

[◀ Previous](#)[Watch video of this tutorial !\[\]\(c7774dea93eb10ead3ed0542c77a8534_img.jpg\)](#)[Next Chapter ▶](#)

CSS Font

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

CSS font properties define the font family, boldness, size, and the style of a text.

Difference Between Serif and Sans-serif Fonts



CSS Font Families

In CSS, there are two types of font family names:

- **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
- **font family** - a specific font family (like "Times New Roman" or "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New	All monospace characters have the same width

Lucida Console



Note: On computer screens, sans-serif fonts are considered easier to read than serif fonts.

Font Family

The font family of a text is set with the font-family property.

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

Example

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

[Try it yourself »](#)

For more commonly used font combinations, look at our [Web Safe Font Combinations](#).

Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Example

```
p.normal {
    font-style: normal;
}

p.italic {
```

```

        font-style: italic;
    }

p.oblique {
    font-style: oblique;
}

```

[Try it yourself »](#)

Font Size

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like `<h1>` - `<h6>` for headings and `<p>` for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

- Sets the text to a specified size
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
- Absolute size is useful when the physical size of the output is known

Relative size:

- Sets the size relative to surrounding elements
- Allows a user to change the text size in browsers

 **Note:** If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

Set Font Size With Pixels

Setting the text size with pixels gives you full control over the text size:

Example

```

h1 {
    font-size: 40px;
}

h2 {
    font-size: 30px;
}

p {

```

```
    font-size: 14px;
}
```

[Try it yourself »](#)

Tip: If you use pixels, you can still use the zoom tool to resize the entire page.

Set Font Size With Em

To allow users to resize the text (in the browser menu), many developers use em instead of pixels.

The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: *pixels/16=em*

Example

```
h1 {
    font-size: 2.5em; /* 40px/16=2.5em */
}

h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
}

p {
    font-size: 0.875em; /* 14px/16=0.875em */
}
```

[Try it yourself »](#)

In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.

Unfortunately, there is still a problem with older versions of IE. The text becomes larger than it should when made larger, and smaller than it should when made smaller.

Use a Combination of Percent and Em

The solution that works in all browsers, is to set a default font-size in percent for the <body> element:

Example

```

body {
    font-size: 100%;
}

h1 {
    font-size: 2.5em;
}

h2 {
    font-size: 1.875em;
}

p {
    font-size: 0.875em;
}

```

[Try it yourself »](#)

Our code now works great! It shows the same text size in all browsers, and allows all browsers to zoom or resize the text!



More Examples

Set the boldness of the font

This example demonstrates how to set the boldness of a font.

Set the variant of the font

This example demonstrates how to set the variant of a font.

All the font properties in one declaration

This example demonstrates how to use the shorthand property for setting all of the font properties in one declaration.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

[Exercise 5 »](#)

All CSS Font Properties

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text

<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

[!\[\]\(25569caef9d46f0e37a5ba4bb4eaed0e_img.jpg\) Previous](#)[Watch video of this tutorial !\[\]\(b5ee4f193e8572102e3090db2261a37f_img.jpg\)](#)[Next Chapter !\[\]\(6ba58ed5429ffd926516910ee2ae6e36_img.jpg\)](#)

CSS Links

[Previous](#)Watch video of this tutorial [Next Chapter »](#)

Links can be styled in different ways.

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Example

```
a {  
    color: #FF0000;  
}
```

[Try it yourself »](#)

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

- 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

Example

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

[Try it yourself »](#)

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

Common Link Styles

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

Text Decoration

The text-decoration property is mostly used to remove underlines from links:

Example

```
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a:active {
    text-decoration: underline;
}
```

[Try it yourself »](#)

Background Color

The background-color property specifies the background color for links:

Example

```
a:link {  
    background-color: #B2FF99;  
}  
  
a:visited {  
    background-color: #FFFF85;  
}  
  
a:hover {  
    background-color: #FF704D;  
}  
  
a:active {  
    background-color: #FF704D;  
}
```

[Try it yourself »](#)



More Examples

[Add different styles to hyperlinks](#)

This example demonstrates how to add other styles to hyperlinks.

[Advanced - Create link boxes](#)

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

[◀ Previous](#)

Watch video of this tutorial

[Next Chapter ▶](#)

CSS Lists

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker

List

In HTML, there are two types of lists:

- unordered lists () - the list items are marked with bullets
- ordered lists () - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

Different List Item Markers

The type of list item marker is specified with the list-style-type property:

Example

```
ul.a {  
    list-style-type: circle;  
}  
  
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {
```

```
list-style-type: lower-alpha;  
}
```

[Try it yourself »](#)

Some of the values are for unordered lists, and some for ordered lists.

An Image as The List Item Marker

To specify an image as the list item marker, use the list-style-image property:

Example

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```

[Try it yourself »](#)

The example above does not display equally in all browsers. IE and Opera will display the image-marker a little bit higher than Firefox, Chrome, and Safari.

If you want the image-marker to be placed equally in all browsers, a crossbrowser solution is explained below.

Crossbrowser Solution

The following example displays the image-marker equally in all browsers:

Example

```
ul {  
    list-style-type: none;  
    padding: 0px;  
    margin: 0px;  
}  
  
ul li {  
    background-image: url(sqpurple.gif);  
    background-repeat: no-repeat;  
    background-position: 0px center;  
    padding-left: 15px;  
}
```

[Try it yourself »](#)

Example explained:

- For ``:
 - Set the `list-style-type` to `none` to remove the list item marker
 - Set both padding and margin to `0px` (for cross-browser compatibility)
 - For all `` in ``:
 - Set the URL of the image, and show it only once (`no-repeat`)
 - Position the image where you want it (left `0px` and vertical value: `center`)
 - Position the text in the list with `padding-left`
-

List - Shorthand property

The `list-style` property is a shorthand property. It is used to set all the list properties in one declaration:

Example

```
ul {
    list-style: square inside url("sqpurple.gif");
}
```

[Try it yourself »](#)

When using the shorthand property, the order of the property values are:

- **list-style-type** (if a `list-style-image` is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
- **list-style-position** (specifies whether the list-item markers should appear inside or outside the content flow)
- **list-style-image** (specifies an image as the list item marker)

If one of the property values above are missing, the default value for the missing property will be inserted, if any.



More Examples

[All the different list-item markers for lists](#)

This example demonstrates all the different list-item markers in CSS.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

All CSS List Properties

Property	Description
<u>list-style</u>	Sets all the properties for a list in one declaration
<u>list-style-image</u>	Specifies an image as the list-item marker
<u>list-style-position</u>	Specifies if the list-item markers should appear inside or outside the content flow
<u>list-style-type</u>	Specifies the type of list-item marker

[« Previous](#)Watch video of this tutorial [Next Chapter »](#)

CSS Tables

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

The look of an HTML table can be greatly improved with CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK
Paris spécialités	Marie Bertrand	France
The Big Cheese	Liz Nixon	USA
Vaffeljernet	Palle Ibsen	Denmark

Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for `<table>`, `<th>`, and `<td>` elements:

Example

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

[Try it yourself »](#)

Notice that the table in the example above has double borders. This is because both the table and the <th>/<td> elements have separate borders.

To display a single border for the table, use the border-collapse property.

Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

Example

```
table {
    border-collapse: collapse;
}

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

[Try it yourself »](#)

Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the <th> elements to 50px:

Example

```
table {
    width: 100%;
}

th {
    height: 50px;
}
```

[Try it yourself »](#)

Horizontal Text Alignment

The `text-align` property sets the horizontal alignment, like `left`, `right`, or `center`.

By default, the text in `<th>` elements are center-aligned and the text in `<td>` elements are left-aligned.

The following example left-aligns the text in `<th>` elements:

Example

```
th {  
    text-align: left;  
}
```

[Try it yourself »](#)

Vertical Text Alignment

The `vertical-align` property sets the vertical alignment, like `top`, `bottom`, or `middle`.

By default, the vertical alignment of text in a table is `middle` (for both `<th>` and `<td>` elements).

The following example sets the vertical text alignment to `bottom` for `<td>` elements:

Example

```
td {  
    height: 50px;  
    vertical-align: bottom;  
}
```

[Try it yourself »](#)

Table Padding

To control the space between the border and content in a table, use the `padding` property on `<td>` and `<th>` elements:

Example

```
td {  
    padding: 15px;  
}
```

[Try it yourself »](#)

Table Color

The example below specifies the color of the borders, and the text and background color of <th> elements:

Example

```
table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}
```

[Try it yourself »](#)



More Examples

[Make a fancy table](#)

This example demonstrates how to create a fancy table.

[Set the position of the table caption](#)

This example demonstrates how to position the table caption.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

[Exercise 5 »](#)

[Exercise 6 »](#)

[◀ Previous](#)

Watch video of this tutorial

[Next Chapter ▶](#)

CSS Box Model

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to add a border around elements, and to define space between elements.

The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

Example

```
div {
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
```

[Try it yourself »](#)

Width and Height of an Element

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.



Important: When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the full size of an element, you must also add padding, borders and margins.

Let's style a <div> element to have a total width of 350px:

Example

```
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
```

[Try it yourself »](#)

Let's do the math:

320px (width)

+ 20px (left + right padding)

+ 10px (left + right border)

+ 0px (left + right margin)

= 350px

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border +

top margin + bottom margin

Browsers Compatibility Issue

Internet Explorer 8 and earlier versions, include padding and border in the width property.

To fix this problem, add a <!DOCTYPE html> to the HTML page.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

[Exercise 3 »](#)

[Exercise 4 »](#)

 [Previous](#)

[Watch video of this tutorial](#) 

[Next Chapter »](#) 

CSS Border

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

CSS Border Properties

The CSS border properties allow you to specify the style, size, and color of an element's border.

Border Style

The border-style property specifies what kind of border to display.



Note: None of the border properties will have ANY effect unless the **border-style** property is set!

border-style values:

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

Try it yourself: [Set the style of the border](#)

Border Width

The border-width property is used to set the width of the border.

The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.

Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.

Example

```
p.one {
    border-style: solid;
    border-width: 5px;
}

p.two {
    border-style: solid;
    border-width: medium;
}
```

[Try it yourself »](#)

Border Color

The border-color property is used to set the color of the border. The color can be set by:

- name - specify a color name, like "red"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Hex - specify a hex value, like "#ff0000"

You can also set the border color to "transparent".

If the border color is not set it is inherited from the color property of the element.

Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.

Example

```
p.one {
    border-style: solid;
    border-color: red;
```

```
}
```

```
p.two {
    border-style: solid;
    border-color: #98bf21;
}
```

[Try it yourself »](#)

Border - Individual sides

In CSS it is possible to specify different borders for different sides:

Example

```
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}
```

[Try it yourself »](#)

The example above can also be set with a single property:

Example

```
p {
    border-style: dotted solid;
}
```

[Try it yourself »](#)

The border-style property can have from one to four values.

- **border-style: dotted solid double dashed;**
 - top border is dotted
 - right border is solid
 - bottom border is double
 - left border is dashed

- **border-style: dotted solid double;**
 - top border is dotted
 - right and left borders are solid
 - bottom border is double

- **border-style: dotted solid;**
 - top and bottom borders are dotted
 - right and left borders are solid
- **border-style: dotted;**
 - all four borders are dotted

The border-style property is used in the example above. However, it also works with border-width and border-color.

Border - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property. This is called a shorthand property.

The border property is a shorthand for the following individual border properties:

- border-width
- border-style (required)
- border-color

Example

```
p {
    border: 5px solid red;
}
```

[Try it yourself »](#)



More Examples

All the top border properties in one declaration

This example demonstrates a shorthand property for setting all of the properties for the top border in one declaration.

Set the style of the bottom border

This example demonstrates how to set the style of the bottom border.

Set the width of the left border

This example demonstrates how to set the width of the left border.

Set the color of the four borders

This example demonstrates how to set the color of the four borders. It can have from one to four

colors.

Set the color of the right border

This example demonstrates how to set the color of the right border.

Test Yourself with Exercises!

[Exercise 1 »](#)[Exercise 2 »](#)[Exercise 3 »](#)[Exercise 4 »](#)

All CSS Border Properties

Property	Description
<u>border</u>	Sets all the border properties in one declaration
<u>border-bottom</u>	Sets all the bottom border properties in one declaration
<u>border-bottom-color</u>	Sets the color of the bottom border
<u>border-bottom-style</u>	Sets the style of the bottom border
<u>border-bottom-width</u>	Sets the width of the bottom border
<u>border-color</u>	Sets the color of the four borders
<u>border-left</u>	Sets all the left border properties in one declaration
<u>border-left-color</u>	Sets the color of the left border
<u>border-left-style</u>	Sets the style of the left border
<u>border-left-width</u>	Sets the width of the left border
<u>border-right</u>	Sets all the right border properties in one declaration
<u>border-right-color</u>	Sets the color of the right border
<u>border-right-style</u>	Sets the style of the right border
<u>border-right-width</u>	Sets the width of the right border
<u>border-style</u>	Sets the style of the four borders
<u>border-top</u>	Sets all the top border properties in one declaration

<u>border-top-color</u>	Sets the color of the top border
<u>border-top-style</u>	Sets the style of the top border
<u>border-top-width</u>	Sets the width of the top border
<u>border-width</u>	Sets the width of the four borders

[◀ Previous](#)Watch video of this tutorial [Next Chapter ▶](#)

CSS Outlines

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

The outline properties specify the style, color, and width of an outline.

Examples

Draw a line around an element (outline)

This example demonstrates how to draw a line around an element, outside the border edge.

Set the style of an outline

This example demonstrates how to set the style of an outline.

Set the color of an outline

This example demonstrates how to set the color of an outline.

Set the width of an outline

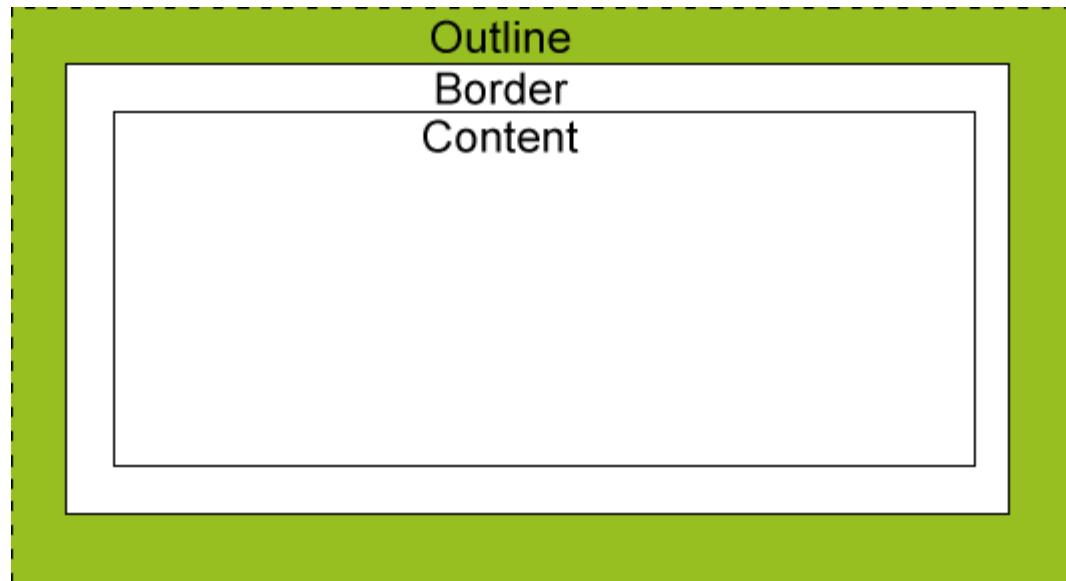
This example demonstrates how to set the width of an outline.

CSS Outline

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

However, the outline property is different from the border property.

The outline is not a part of an element's dimensions; the element's total width and height is not affected by the width of the outline.



Test Yourself with Exercises!

[Exercise 1 »](#)
[Exercise 2 »](#)
[Exercise 3 »](#)

All CSS Outline Properties

Property	Description	Values
<u>outline</u>	Sets all the outline properties in one declaration	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit
<u>outline-color</u>	Sets the color of an outline	<i>color_name</i> <i>hex_number</i> <i>rgb_number</i> invert inherit
<u>outline-style</u>	Sets the style of an outline	none dotted dashed solid double groove ridge inset outset

inherit

<u>outline-width</u>	Sets the width of an outline	thin medium thick <i>length</i> inherit
----------------------	------------------------------	---

 Previous

Watch video of this tutorial 

Next Chapter 

CSS Margin

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

The CSS margin properties define the space around elements.

Margin

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.

Possible Values

Value	Description
auto	The browser calculates a margin
<i>length</i>	Specifies a margin in px, pt, cm, etc. Default value is 0px
%	Specifies a margin in percent of the width of the containing element
inherit	Specifies that the margin should be inherited from the parent element



Note: It is also possible to use negative values, to overlap content.

Margin - Individual sides

In CSS, it is possible to specify different margins for different sides of an element:

Example

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 50px;  
}
```

[Try it yourself »](#)

Margin - Shorthand property

To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.

The shorthand property for all the margin properties is "margin":

Example

```
p {  
    margin: 100px 50px;  
}
```

[Try it yourself »](#)

The margin property can have from one to four values.

- **margin: 25px 50px 75px 100px;**
 - top margin is 25px
 - right margin is 50px
 - bottom margin is 75px
 - left margin is 100px
- **margin: 25px 50px 75px;**
 - top margin is 25px
 - right and left margins are 50px
 - bottom margin is 75px
- **margin: 25px 50px;**
 - top and bottom margins are 25px
 - right and left margins are 50px
- **margin: 25px;**
 - all four margins are 25px



More Examples

Set the top margin of a text using a cm value

This example demonstrates how to set the top margin of a text using a cm value.

Set the bottom margin of a text using a percent value

This example demonstrates how to set the bottom margin in percent, relative to the width of the containing element.

Test Yourself with Exercises!

[Exercise 1 »](#)
[Exercise 2 »](#)
[Exercise 3 »](#)

All CSS Margin Properties

Property	Description
<u>margin</u>	A shorthand property for setting the margin properties in one declaration
<u>margin-bottom</u>	Sets the bottom margin of an element
<u>margin-left</u>	Sets the left margin of an element
<u>margin-right</u>	Sets the right margin of an element
<u>margin-top</u>	Sets the top margin of an element

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

CSS Padding

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

The CSS padding properties define the space between the element border and the element content.

Padding

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

Possible Values

Value	Description
<i>length</i>	Defines a fixed padding (in pixels, pt, em, etc.)
%	Defines a padding in % of the containing element

Padding - Individual sides

In CSS, it is possible to specify different padding for different sides:

Example

```
p {
    padding-top: 25px;
    padding-right: 50px;
    padding-bottom: 25px;
    padding-left: 50px;
}
```

[Try it yourself »](#)

Padding - Shorthand property

To shorten the code, it is possible to specify all the padding properties in one property. This is called a shorthand property.

The shorthand property for all the padding properties is "padding":

Example

```
p {  
    padding: 25px 50px;  
}
```

[Try it yourself »](#)

The padding property can have from one to four values.

- **padding: 25px 50px 75px 100px;**
 - top padding is 25px
 - right padding is 50px
 - bottom padding is 75px
 - left padding is 100px
- **padding: 25px 50px 75px;**
 - top padding is 25px
 - right and left paddings are 50px
 - bottom padding is 75px
- **padding: 25px 50px;**
 - top and bottom paddings are 25px
 - right and left paddings are 50px
- **padding: 25px;**
 - all four paddings are 25px



More Examples

All the padding properties in one declaration

This example demonstrates a shorthand property for setting all of the padding properties in one declaration, can have from one to four values.

[Set the left padding](#)

This example demonstrates how to set the left padding of a <p> element.

[Set the right padding](#)

This example demonstrates how to set the right padding of a <p> element.

[Set the top padding](#)

This example demonstrates how to set the top padding of a <p> element.

[Set the bottom padding](#)

This example demonstrates how to set the bottom padding of a <p> element.

Test Yourself with Exercises!

[Exercise 1 »](#)
[Exercise 2 »](#)
[Exercise 3 »](#)

All CSS Padding Properties

Property	Description
<u>padding</u>	A shorthand property for setting all the padding properties in one declaration
<u>padding-bottom</u>	Sets the bottom padding of an element
<u>padding-left</u>	Sets the left padding of an element
<u>padding-right</u>	Sets the right padding of an element
<u>padding-top</u>	Sets the top padding of an element

[◀ Previous](#)
[Watch video of this tutorial !\[\]\(eda79f3ccaf93769389d4ab7084918ac_img.jpg\)](#)
[Next Chapter ▶](#)

CSS Dimension

[Previous](#)

[Next Chapter](#)

The CSS dimension properties allow you to control the height and width of an element.

Try it Yourself - Examples

Set the height of elements

This example demonstrates how to set the height of different elements.

Set the height of an image using percent

This example demonstrates how to set the height of an element using a percent value.

Set the width of an element using a pixel value

This example demonstrates how to set the width of an element using a pixel value.

Set the maximum height of an element

This example demonstrates how to set the maximum height of an element.

Set the maximum width of an element using percent

This example demonstrates how to set the maximum width of an element using a percent value.

Set the minimum height of an element

This example demonstrates how to set the minimum height of an element.

Set the minimum width of an element using a pixel value

This example demonstrates how to set the minimum width of an element using a pixel value.

Test Yourself with Exercises!

[Exercise 1 »](#)

[Exercise 2 »](#)

All CSS Dimension Properties

Property	Description	Values

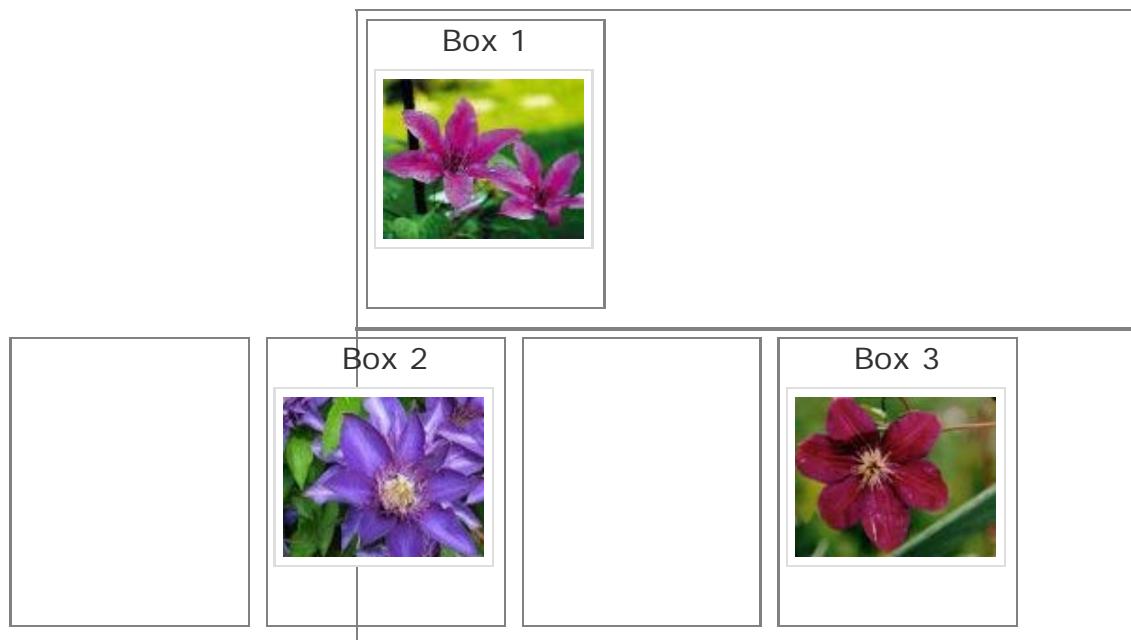
<u>height</u>	Sets the height of an element	auto <i>length</i> % inherit
<u>max-height</u>	Sets the maximum height of an element	none <i>length</i> % inherit
<u>max-width</u>	Sets the maximum width of an element	none <i>length</i> % inherit
<u>min-height</u>	Sets the minimum height of an element	<i>length</i> % inherit
<u>min-width</u>	Sets the minimum width of an element	<i>length</i> % inherit
<u>width</u>	Sets the width of an element	auto <i>length</i> % inherit

[◀ Previous](#)[Next Chapter ▶](#)

CSS Display and Visibility

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

The display property specifies if/how an element is displayed, and the visibility property specifies if an element should be visible or hidden.



Hiding an Element - display:none or visibility:hidden

Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden".

However, notice that these two methods produce different results:

visibility:hidden hides an element, but it 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 »](#)

display:none hides an element, and it will not take up any space. 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 »](#)

CSS Display - Block and Inline Elements

A block element is an element that takes up the full width available, and has a line break before and after it.

Examples of block elements:

- <h1>
- <p>
-
- <div>

An inline element only takes up as much width as necessary, and does not force line breaks.

Examples of inline elements:

-
- <a>

Changing How an Element is Displayed

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 web standards.

The following example displays elements as inline elements:

Example

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

}

[Try it yourself »](#)

The following example displays elements as block elements:

Example

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

[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 of it.



More Examples

[How to display an element as an inline element.](#)

This example demonstrates how to display an element as an inline element.

[How to make a table element collapse](#)

This example demonstrates how to make a <table> element collapse.

[◀ Previous](#)

Watch video of this tutorial

[Next Chapter ▶](#)

CSS Positioning

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

Decide which element to display in front!

Positioning can be tricky sometimes!

Elements can overlap!

Positioning

The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

There are four different positioning methods.

Static Positioning

HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

Static positioned elements are not affected by the top, bottom, left, and right properties.

Fixed Positioning

An element with a fixed position is positioned relative to the browser window, and will not move even if the window is scrolled:

Example

```
p.pos_fixed {
    position: fixed;
    top: 30px;
    right: 5px;
}
```

[Try it yourself »](#)



Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.

Fixed positioned elements are removed from the normal flow. The document and other elements behave like the fixed positioned element does not exist.

Fixed positioned elements can overlap other elements.

Relative Positioning

A relative positioned element is positioned relative to its normal position:

Example

```
h2.pos_left {
    position: relative;
    left: -20px;
}

h2.pos_right {
    position: relative;
    left: 20px;
}
```

[Try it yourself »](#)

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

Example

```
h2.pos_top {
    position: relative;
    top: -50px;
}
```

[Try it yourself »](#)

Relatively positioned elements are often used as container blocks for absolutely positioned

elements.

Absolute Positioning

An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

Example

```
h2 {
    position: absolute;
    left: 100px;
    top: 150px;
}
```

[Try it yourself »](#)

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

Absolutely positioned elements can overlap other elements.

Overlapping Elements

When elements are positioned outside the normal flow, they can overlap other elements.

The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order:

Example

```
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
```

[Try it yourself »](#)

An element with greater stack order is always in front of an element with a lower stack order.



Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.



More Examples

Set the shape of an element

This example demonstrates how to set the shape of an element. The element is clipped into this shape, and displayed.

How to show overflow in an element using scroll

This example demonstrates how to set the overflow property to create a scroll bar when an element's content is too big to fit in a specified area.

How to set the browser to automatically handle overflow

This example demonstrates how to set the browser to automatically handle overflow.

Change the cursor

This example demonstrates how to change the cursor.

All CSS Positioning Properties

Property	Description	Values
<u>bottom</u>	Sets the bottom margin edge for a positioned box	auto <i>length</i> % inherit
<u>clip</u>	Clips an absolutely positioned element	<i>shape</i> auto inherit
<u>cursor</u>	Specifies the type of cursor to be displayed	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text

[wait](#)
[help](#)

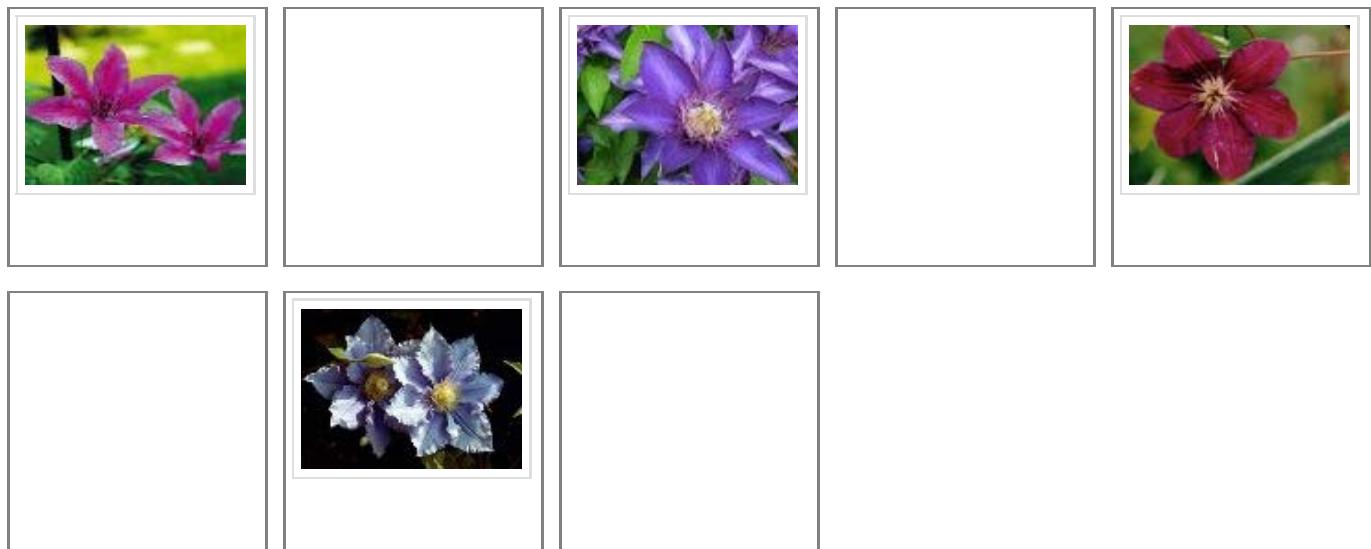
<u>left</u>	Sets the left margin edge for a positioned box	auto <i>length</i> % inherit
<u>overflow</u>	Specifies what happens if content overflows an element's box	auto hidden scroll visible inherit
<u>position</u>	Specifies the type of positioning for an element	absolute fixed relative static inherit
<u>right</u>	Sets the right margin edge for a positioned box	auto <i>length</i> % inherit
<u>top</u>	Sets the top margin edge for a positioned box	auto <i>length</i> % inherit
<u>z-index</u>	Sets the stack order of an element	<i>number</i> auto inherit

[◀ Previous](#)
[Watch video of this tutorial](#)
[Next Chapter ▶](#)

CSS Float

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

What is CSS Float?



With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is often used with images, but it is also useful when working with layouts.

How Elements Float

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

The elements after the floating element will flow around it.

The elements before the floating element will not be affected.

If an image is floated to the right, a following text flows around it, to the left:

Example

```
img {  
    float: right;  
}
```

[Try it yourself »](#)

Floating Elements Next to Each Other

If you place several floating elements after each other, they will float next to each other if there is room.

Here we have made an image gallery using the float property:

Example

```
.thumbnail {  
    float: left;  
    width: 110px;  
    height: 90px;  
    margin: 5px;  
}
```

[Try it yourself »](#)

Turning off Float - Using Clear

Elements after the floating element will flow around it. To avoid this, use the clear property.

The clear property specifies which sides of an element other floating elements are not allowed.

Add a text line into the image gallery, using the clear property:

Example

```
.text_line {  
    clear: both;  
}
```

[Try it yourself »](#)



More Examples

An image with border and margins that floats to the right in a paragraph

Let an image float to the right in a paragraph. Add border and margins to the image.

An image with a caption that floats to the right

Let an image with a caption float to the right.

Let the first letter of a paragraph float to the left

Let the first letter of a paragraph float to the left and style the letter.

Creating a horizontal menu

Use float with a list of hyperlinks to create a horizontal menu.

Creating a homepage without tables

Use float to create a homepage with a header, footer, left content and main content.

All CSS Float Properties

Property	Description	Values
<u>clear</u>	Specifies which sides of an element where other floating elements are not allowed	left right both none inherit
<u>float</u>	Specifies whether or not a box should float	left right none inherit

◀ Previous

Watch video of this tutorial

Next Chapter ▶

CSS Horizontal Align

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

In CSS, several properties are used to align elements horizontally.

Aligning Block Elements

A block element is an element that takes up the full width available, and has a line break before and after it.

Examples of block elements:

- <h1>
- <p>
- <div>

For aligning text, see the [CSS Text](#) chapter.

In this chapter we will show you how to horizontally align block elements for layout purposes.

Center Aligning Using the margin Property

Block elements can be center-aligned by setting the left and right margins to "auto".



Note: Using margin: auto; will not work in IE8 and earlier **unless a !DOCTYPE is declared.**

Setting the left and right margins to auto specifies that they should split the available margin equally. The result is a centered element:

Example

```
.center {
    margin-left: auto;
    margin-right: auto;
    width: 70%;
    background-color: #b0e0e6;
}
```

[Try it yourself »](#)

Tip: Center-aligning has no effect if the width is 100%.

Left and Right Aligning Using the position Property

One method of aligning elements is to use absolute positioning:

Example

```
.right {
    position: absolute;
    right: 0px;
    width: 300px;
    background-color: #b0e0e6;
}
```

[Try it yourself »](#)

Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

Crossbrowser Compatibility Issues

When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is a problem with IE8 and earlier, when using the position property. If a container element (in our case <div class="container">) has a specified width, and the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the position property:

Example

```
body {
    margin: 0;
```

```

    padding: 0;
}

.container {
    position: relative;
    width: 100%;
}

.right {
    position: absolute;
    right: 0px;
    width: 300px;
    background-color: #b0e0e6;
}

```

[Try it yourself »](#)

Left and Right Aligning Using the float Property

One method of aligning elements is to use the float property:

Example

```

.right {
    float: right;
    width: 300px;
    background-color: #b0e0e6;
}

```

[Try it yourself »](#)

Crossbrowser Compatibility Issues

When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is a problem with IE8 and earlier when using the float property. If the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the float property:

Example

```

body {
    margin: 0;
    padding: 0;
}

```

```
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}
```

[Try it yourself »](#)

[« Previous](#)

Watch video of this tutorial 

[Next Chapter »](#)

CSS Combinators

[Previous](#)[Next Chapter](#)

CSS Combinators



A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS3:

- descendant selector
- child selector
- adjacent sibling selector
- general sibling selector

Descendant Selector

The descendant selector matches all element that are descendants of a specified element.

The following example selects all `<p>` elements inside `<div>` elements:

Example

```
div p {  
    background-color: yellow;  
}
```

[Try it yourself »](#)

Child Selector

The child selector selects all elements that are the immediate children of a specified element.

The following example selects all `<p>` elements that are immediate children of a `<div>` element:

Example

```
div > p {  
    background-color: yellow;  
}
```

[Try it yourself »](#)

Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all `<p>` elements that are placed immediately after `<div>` elements:

Example

```
div + p {  
    background-color: yellow;  
}
```

[Try it yourself »](#)

General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all `<p>` elements that are siblings of `<div>` elements:

Example

```
div ~ p {  
    background-color: yellow;  
}
```

[Try it yourself »](#)

[« Previous](#)

[Next Chapter »](#)

CSS Pseudo-classes

[Previous](#)Watch video of this tutorial [Next Chapter](#) 

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently

Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {  
    property:value;  
}
```

Anchor Pseudo-classes

Links can be displayed in different ways:

Example

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

Try it yourself »

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective!!



a:active MUST come after a:hover in the CSS definition in order to be effective!!
Pseudo-class names are not case-sensitive.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

Example

```
a.highlight:hover {
    color: #ff0000;
}
```

Try it yourself »

When you hover over the link in the example, it will change color.

CSS - The :first-child Pseudo-class

The :first-child pseudo-class matches a specified element that is the first child of another element.



Note: For :first-child to work in IE8 and earlier, a `<!DOCTYPE>` must be declared.

Match the first <p> element

In the following example, the selector matches any <p> element that is the first child of any element:

Example

```
p:first-child {
    color: blue;
}
```

[Try it yourself »](#)

Match the first <i> element in all <p> elements

In the following example, the selector matches the first <i> element in all <p> elements:

Example

```
p i:first-child {
    color: blue;
}
```

[Try it yourself »](#)

Match all <i> elements in all first child <p> elements

In the following example, the selector matches all <i> elements in <p> elements that are the first child of another element:

Example

```
p:first-child i {
    color: blue;
}
```

[Try it yourself »](#)

CSS - The :lang Pseudo-class

The :lang pseudo-class allows you to define special rules for different languages.



Note: IE8 supports the :lang pseudo-class only if a <!DOCTYPE> is specified.

In the example below, the :lang class defines the quotation marks for <q> elements with lang="no":

Example

```

<html>
<head>
<style>
q:lang(no) {
    quotes: "~" "~";
}
</style>
</head>

<body>
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
</body>
</html>

```

[Try it yourself »](#)



More Examples

Add different styles to hyperlinks

This example demonstrates how to add other styles to hyperlinks.

Use of :focus

This example demonstrates how to use the :focus pseudo-class.

All CSS Pseudo Classes

Selector	Example	Example description
<u>:active</u>	a: active	Selects the active link
<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p: empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p: first-child	Selects every <p> elements that is the first child of its parent
<u>:first-of-type</u>	p: first-of-type	Selects every <p> element that is the first <p> element of its parent

<u>:focus</u>	input:focus	Selects the <input> element that has focus
<u>:hover</u>	a:hover	Selects links on mouse over
<u>:in-range</u>	input:in-range	Selects <input> elements with a value within a specified range
<u>:invalid</u>	input:invalid	Selects all <input> elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<u>:last-child</u>	p:last-child	Selects every <p> elements that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links
<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects <input> elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects <input> elements with a value outside a specified range
<u>:read-only</u>	input:read-only	Selects <input> elements with a "readonly"

		attribute specified
<u>:read-write</u>	input:read-write	Selects <input> elements with no "readonly" attribute
<u>:required</u>	input:required	Selects <input> elements with a "required" attribute specified
<u>:root</u>	root	Selects the document's root element
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<u>:valid</u>	input:valid	Selects all <input> elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links

All CSS Pseudo Elements

Selector	Example	Example description
<u>::after</u>	p::after	Insert content after every <p> element
<u>::before</u>	p::before	Insert content before every <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>	p::first-line	Selects the first line of every <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

[◀ Previous](#)
[Watch video of this tutorial](#)
[Next Chapter ▶](#)

CSS Pseudo-elements

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

Syntax

The syntax of pseudo-elements:

```
selector::pseudo-element {  
    property:value;  
}
```

Notice the double colon notation - **::first-line** versus **:first-line**

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between **pseudo-classes** and **pseudo-elements**.



The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1.

For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

The **::first-line** Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

The ::first-line pseudo-element can only be applied to block elements.

Example

Format the first line of the text in all <p> elements:

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

[Try it yourself »](#)

The following properties apply to the ::first-line pseudo-element:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

The ::first-letter pseudo-element can only be applied to block elements.

Example

Format the first letter of the text in all <p> elements:

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

[Try it yourself »](#)

The following properties apply to the ::first-letter pseudo- element:

- font properties
- color properties

- background properties
 - margin properties
 - padding properties
 - border properties
 - text-decoration
 - vertical-align (only if "float" is "none")
 - text-transform
 - line-height
 - float
 - clear
-

Pseudo-elements and CSS Classes

Pseudo-elements can be combined with CSS classes:

Example

```
p.intro::first-letter {  
    color: #ff0000;  
    font-size: 200%;  
}
```

[Try it yourself »](#)

The example above will display the first letter of paragraphs with class="intro", in red and in a larger size.

Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

Example

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
  
p::first-line {  
    color: #0000ff;  
    font-variant: small-caps;  
}
```

[Try it yourself »](#)

CSS - The ::before Pseudo-element

The ::before pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before each <h1> element:

Example

```
h1::before {  
    content: url(smiley.gif);  
}
```

[Try it yourself »](#)

CSS - The ::after Pseudo-element

The ::after pseudo-element can be used to insert some content after the content of an element.

The following example inserts an image after each <h1> element:

Example

```
h1::after {  
    content: url(smiley.gif);  
}
```

[Try it yourself »](#)

CSS - The ::selection Pseudo-element

The ::selection pseudo-element matches the portion of an element that is selected by a user.

The following CSS properties can be applied to ::selection: color, background, cursor, and outline.

The following example makes the selected text red on a yellow background:

Example

```
::selection {
```

```

color: red;
background: yellow;
}

```

[Try it yourself »](#)

All CSS Pseudo Elements

Selector	Example	Example description
<u>::after</u>	p::after	Insert content after every <p> element
<u>::before</u>	p::before	Insert content before every <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>	p::first-line	Selects the first line of every <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

All CSS Pseudo Classes

Selector	Example	Example description
<u>:active</u>	a:active	Selects the active link
<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p:empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p:first-child	Selects every <p> elements that is the first child of its parent
<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<u>:focus</u>	input:focus	Selects the <input> element that has focus
<u>:hover</u>	a:hover	Selects links on mouse over

<u>:in-range</u>	input:in-range	Selects <input> elements with a value within a specified range
<u>:invalid</u>	input:invalid	Selects all <input> elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<u>:last-child</u>	p:last-child	Selects every <p> elements that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links
<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects <input> elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects <input> elements with a value outside a specified range
<u>:read-only</u>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects <input> elements with no "readonly" attribute

<u>:required</u>	input:required	Selects <input> elements with a "required" attribute specified
<u>:root</u>	root	Selects the document's root element
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<u>:valid</u>	input:valid	Selects all <input> elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links

[!\[\]\(3eaf16fa7d4c79c2bb7f9a4d5a47a2b0_img.jpg\) Previous](#)[Watch video of this tutorial !\[\]\(5d4d5eb1897f9b0a671e4cb1f16bf38c_img.jpg\)](#)[Next Chapter !\[\]\(33682045bc8be0a888f70a38b4d919ad_img.jpg\)](#)

CSS Navigation Bar

[Previous](#)Watch video of this tutorial [Next Chapter](#) 

Demo: Navigation Bar

[HOME](#) [NEWS](#) [ARTICLES](#) [FORUM](#) [CONTACT](#) [ABOUT](#)

Navigation Bars

Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

In our examples we will build the navigation bar from a standard HTML list.

A navigation bar is basically a list of links, so using the `` and `` elements makes perfect sense:

Example

```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

[Try it yourself »](#)

Now let's remove the bullets and the margins and padding from the list:

Example

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
```

[Try it yourself »](#)

Example explained:

- list-style-type: none - Removes the bullets. A navigation bar does not need list markers
- Setting margins and padding to 0 to remove browser default settings

The code in the example above is the standard code used in both vertical, and horizontal navigation bars.

Vertical Navigation Bar

To build a vertical navigation bar we only need to style the `<a>` elements, in addition to the code above:

Example

```
a {
    display: block;
    width: 60px;
}
```

[Try it yourself »](#)

Example explained:

- display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width: 60px - Block elements take up the full width available by default. We want to specify a 60 px width

Tip: Also take a look at our [fully styled vertical navigation bar example](#).



Note: Always specify the width for `<a>` elements in a vertical navigation bar. If you omit the width, IE6 can produce unexpected results.

Horizontal Navigation Bar

There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items.

Both methods work fine, but if you want the links to be the same size, you have to use the floating method.

Inline List Items

One way to build a horizontal navigation bar is to specify the elements as inline, in addition to the "standard" code above:

Example

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

[Try it yourself »](#)

Example explained:

- display: inline; - By default, elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line

Floating List Items

In the example above the links have different widths.

For all the links to have an equal width, float the elements and specify a width for the <a> elements:

Example

```
li {  
    float: left;  
}  
  
a {  
    display: block;  
    width: 60px;  
}
```

[Try it yourself »](#)

Example explained:

- float: left - use float to get block elements to slide next to each other
- display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width: 60px - Since block elements take up the full width available, they cannot float next to

each other. We specify the width of the links to 60px

Tip: Also take a look at our [fully styled horizontal navigation bar example](#).

 Previous

Watch video of this tutorial 

Next Chapter 

CSS Image Gallery

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

CSS can be used to create an image gallery.



Add a
description of
the image here



Add a
description of
the image here



Add a
description of
the image here



Add a
description of
the image here

Image Gallery

The following image gallery is created with CSS:

Example

```
<html>
<head>
<style>
div.img {
    margin: 5px;
    padding: 5px;
    border: 1px solid #0000ff;
    height: auto;
    width: auto;
    float: left;
    text-align: center;
}

div.img img {
    display: inline;
    margin: 5px;
```

```

        border: 1px solid #ffffff;
    }

div.img a:hover img {
    border:1px solid #0000ff;
}

div.desc {
    text-align: center;
    font-weight: normal;
    width: 120px;
    margin: 5px;
}

```

</style>

</head>

<body>

```

<div class="img">
    <a target="_blank" href="klematis_big.htm">
        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>
<div class="img">
    <a target="_blank" href="klematis2_big.htm">
        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>
<div class="img">
    <a target="_blank" href="klematis3_big.htm">
        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>
<div class="img">
    <a target="_blank" href="klematis4_big.htm">
        
    </a>
    <div class="desc">Add a description of the image here</div>
</div>

```

```

</body>
</html>

```

[Try it yourself »](#)

[Previous](#)

Watch video of this tutorial 

[Next Chapter »](#)

CSS Image Opacity / Transparency

[Previous](#)[Watch video of this tutorial](#)[Next Chapter](#)

Creating transparent images with CSS is easy.

The CSS opacity property is a part of the CSS3 recommendation.

Example 1 - Creating a Transparent Image

The CSS3 property for transparency is **opacity**.

First we will show you how to create a transparent image with CSS.

Regular image:



The same image with transparency:



Look at the following CSS:

Example

```
img {  
    opacity: 0.4;  
    filter: alpha(opacity=40); /* For IE8 and earlier */  
}
```

[Try it yourself »](#)

IE9, Firefox, Chrome, Opera, and Safari use the property **opacity** for transparency. The opacity property can take a value from 0.0 - 1.0. A lower value makes the element more transparent.

IE8 and earlier use **filter:alpha(opacity=x)**. The x can take a value from 0 - 100. A lower value makes the element more transparent.

Example 2 - Image Transparency - Hover Effect

Mouse over the images:



The CSS looks like this:

Example

```
img {
    opacity: 0.4;
    filter: alpha(opacity=40); /* For IE8 and earlier */
}

img:hover {
    opacity: 1.0;
    filter: alpha(opacity=100); /* For IE8 and earlier */
}
```

[Try it yourself »](#)

The first CSS block is similar to the code in Example 1. In addition, we have added what should happen when a user hover over one of the images. In this case we want the image to NOT be transparent when the user hover over it.

The CSS for this is: **opacity=1**.

IE8 and earlier: **filter:alpha(opacity=100)**.

When the mouse pointer moves away from the image, the image will be transparent again.

Example 3 - Text in Transparent Box





The source code looks like this:

Example

```

<html>
<head>
<style>
div.background {
    background: url(klematis.jpg) repeat;
    border: 2px solid black;
}

div.transbox {
    margin: 30px;
    background-color: #ffffff;
    border: 1px solid black;
    opacity: 0.6;
    filter: alpha(opacity=60); /* For IE8 and earlier */
}

div.transbox p {
    margin: 5%;
    font-weight: bold;
    color: #000000;
}
</style>
</head>
<body>

<div class="background">
    <div class="transbox">
        <p>This is some text that is placed in the transparent box.</p>
    </div>
</div>

</body>
</html>
  
```

Try it yourself »

First, we create a `<div>` element (`class="background"`) with a background image, and a border. Then we create another `<div>` (`class="transbox"`) inside the first `<div>`. The `<div class="transbox">` have a background color, and a border - the div is transparent. Inside the transparent `<div>`, we add some text inside a `<p>` element.

 Previous

Watch video of this tutorial 

Next Chapter 

CSS Image Sprites

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

Image Sprites

An image sprite is a collection of images put into a single image.

A web page with many images can take a long time to load and generates multiple server requests.

Using image sprites will reduce the number of server requests and save bandwidth.

Image Sprites - Simple Example

Instead of using three separate images, we use this single image ("img_navsprites.gif"):



With CSS, we can show just the part of the image we need.

In the following example the CSS specifies which part of the "img_navsprites.gif" image to show:

Example

```
#home {
    width: 46px;
    height: 44px;
    background: url(img_navsprites.gif) 0 0;
}
```

[Try it yourself »](#)

Example explained:

- - Only defines a small transparent image because the src attribute cannot be empty. The displayed image will be the background image we specify

in CSS

- width: 46px; height: 44px; - Defines the portion of the image we want to use
- background: url(img_navsprites.gif) 0 0; - Defines the background image and its position (left 0px, top 0px)

This is the easiest way to use image sprites, now we want to expand it by using links and hover effects.

Image Sprites - Create a Navigation List

We want to use the sprite image ("img_navsprites.gif") to create a navigation list.

We will use an HTML list, because it can be a link and also supports a background image:

Example

```
#navlist {
    position: relative;
}

#navlist li {
    margin: 0;
    padding: 0;
    list-style: none;
    position: absolute;
    top: 0;
}

#navlist li, #navlist a {
    height: 44px;
    display: block;
}

#home {
    left: 0px;
    width: 46px;
    background: url('img_navsprites.gif') 0 0;
}

#prev {
    left: 63px;
    width: 43px;
    background: url('img_navsprites.gif') -47px 0;
}

#next {
    left: 129px;
    width: 43px;
    background: url('img_navsprites.gif') -91px 0;
}
```

[Try it yourself »](#)

Example explained:

- `#navlist {position: relative;}` - position is set to relative to allow absolute positioning inside it
- `#navlist li {margin: 0; padding: 0; list-style: none; position: absolute; top: 0;}` - margin and padding is set to 0, list-style is removed, and all list items are absolute positioned
- `#navlist li, #navlist a {height: 44px; display: block;}` - the height of all the images are 44px

Now start to position and style for each specific part:

- `#home {left: 0px; width: 46px;}` - Positioned all the way to the left, and the width of the image is 46px
- `#home {background: url(img_navsprites.gif) 0 0;}` - Defines the background image and its position (left 0px, top 0px)
- `#prev {left: 63px; width: 43px;}` - Positioned 63px to the right (#home width 46px + some extra space between items), and the width is 43px.
- `#prev {background: url('img_navsprites.gif') -47px 0;}` - Defines the background image 47px to the right (#home width 46px + 1px line divider)
- `#next {left: 129px; width: 43px;}` - Positioned 129px to the right (start of #prev is 63px + #prev width 43px + extra space), and the width is 43px.
- `#next {background: url('img_navsprites.gif') -91px 0;}` - Defines the background image 91px to the right (#home width 46px + 1px line divider + #prev width 43px + 1px line divider)

Image Sprites - Hover Effect

Now we want to add a hover effect to our navigation list.



The :hover selector is used to select elements when you mouse over them.

Tip: The :hover selector can be used on all elements, not only on links.

Our new image ("img_navsprites_hover.gif") contains three navigation images and three images to use for hover effects:



Because this is one single image, and not six separate files, there will be **no loading delay** when a user hovers over the image.

We only add three lines of code to add the hover effect:

Example

```
#home a:hover {
    background: url('img_navsprites_hover.gif') 0 -45px;
```

```
}
```

```
#prev a:hover {  
    background: url('img_navsprites_hover.gif') -47px -45px;  
}  
  
#next a:hover {  
    background: url('img_navsprites_hover.gif') -91px -45px;  
}
```

[Try it yourself »](#)

Example explained:

- #home a:hover {background: transparent url('img_navsprites_hover.gif') 0 -45px;} - For all three hover images we specify the same background position, only 45px further down

[Previous](#)

Watch video of this tutorial 

[Next Chapter »](#)

CSS Media Types

[Previous](#)[Next Chapter »](#)

Media Types

Some CSS properties are designed for a specific type of media. For example the "voice-family" property is designed for aural user agents.

Some other CSS properties can be used for different media types. For example, the "font-size" property can be used for both screen and print media, but perhaps with different values. A document usually needs a larger font-size on a screen than on paper, and sans-serif fonts are easier to read on the screen, while serif fonts are easier to read on paper.

The @media Rule

The @media rule makes it possible to define different style rules for different media types in the same stylesheet.

The CSS in the example below tells the browser to display a 17 pixels Verdana font on the screen. But if the page is printed, it will be in a blue 14 pixels Georgia font:

Example

```
@media screen {  
    p {  
        font-family: verdana, sans-serif;  
        font-size: 17px;  
    }  
}  
  
@media print {  
    p {  
        font-family: georgia, serif;  
        font-size: 14px;  
        color: blue;  
    }  
}
```

[Try it yourself »](#)

Other Media Types

Media Type	Description
all	Used for all media type devices
aural	Used for speech and sound synthesizers
braille	Used for braille tactile feedback devices
embossed	Used for paged braille printers
handheld	Used for small or handheld devices
print	Used for printers
projection	Used for projected presentations, like slides
screen	Used for computer screens
tty	Used for media using a fixed-pitch character grid, like teletypes and terminals
tv	Used for television-type devices

[Previous](#)[Next Chapter](#)

CSS Attribute Selectors

[Previous](#)
[Watch video of this tutorial](#)
[Next Chapter](#)

Style HTML Elements With Specific Attributes

It is possible to style HTML elements that have specific attributes or attribute values.

 **Note:** IE7 and IE8 support attribute selectors only if a !DOCTYPE is specified.

CSS [attribute] Selector

The [attribute] selector is used to select elements with a specified attribute.

The following example selects all <a> elements with a target attribute:

Example

```
a[target] {
    background-color: yellow;
}
```

[Try it yourself »](#)

CSS [attribute=value] Selector

The [attribute=value] selector is used to select elements with a specified attribute and value.

The following example selects all <a> elements with a target="_blank" attribute:

Example

```
a[target="_blank"] {
    background-color: yellow;
```

}

[Try it yourself »](#)

CSS [attribute~=value] Selector

The [attribute~=value] selector is used to select elements with an attribute value containing a specified word.

The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":

Example

```
[title~="flower"] {  
    border: 5px solid yellow;  
}
```

[Try it yourself »](#)

The example above will match elements with title="flower", title="summer flower", and title="flower new", but not title="my-flower" or title="flowers".

CSS [attribute|=value] Selector

The [attribute|=value] selector is used to select elements with the specified attribute starting with the specified value.

The following example selects all elements with a class attribute value that begins with "top":

Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen(-), like class="top-text"!

Example

```
[class|="top"] {  
    background: yellow;  
}
```

[Try it yourself »](#)

CSS [attribute^=value] Selector

The [attribute[^]=value] selector is used to select elements whose attribute value begins with a specified value.

The following example selects all elements with a class attribute value that begins with "top":

Note: The value does not have to be a whole word!

Example

```
[class^="top"] {  
    background: yellow;  
}
```

[Try it yourself »](#)

CSS [attribute\$=value] Selector

The [attribute\$=value] selector is used to select elements whose attribute value ends with a specified value.

The following example selects all elements with a class attribute value that ends with "test":

Note: The value does not have to be a whole word!

Example

```
[class$="test"] {  
    background: yellow;  
}
```

[Try it yourself »](#)

CSS [attribute*=value] Selector

The [attribute*=value] selector is used to select elements whose attribute value contains a specified value.

The following example selects all elements with a class attribute value that contains "te":

Note: The value does not have to be a whole word!

Example

```
[class*="te"] {  
    background: yellow;  
}
```

[Try it yourself »](#)

Styling Forms

The attribute selectors can be useful for styling forms without class or ID:

Example

```
input[type="text"] {  
    width: 150px;  
    display: block;  
    margin-bottom: 10px;  
    background-color: yellow;  
}  
  
input[type="button"] {  
    width: 120px;  
    margin-left: 35px;  
    display: block;  
}
```

[Try it yourself »](#)

More Examples of CSS Selectors

Use our [CSS Selector Tester](#) to demonstrate the different selectors.

For a complete reference of all the CSS selectors, please go to our [CSS Selectors Reference](#).

[◀ Previous](#)[Watch video of this tutorial](#)[Next Chapter ▶](#)

CSS3 Introduction

[◀ Previous](#)

[Next Chapter ▶](#)



CSS3 is the latest standard for CSS.

CSS3 is completely backwards-compatible with earlier versions of CSS.

This section teaches you about the new features in CSS3!



CSS3 Modules

CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.

Some of the most important CSS3 modules are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

CSS3 Recommendation

Most of the CSS3 Modules are W3C Recommendations, and CSS3 properties are implemented in

all modern browsers.

[« Previous](#)

[Next Chapter »](#)

CSS3 Borders

[Previous](#)

[Next Chapter](#)

CSS3 Borders

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop.

In this chapter you will learn about the following border properties:

- border-radius
- box-shadow
- border-image

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
border-radius	9.0	5.0 4.0 -webkit-	4.0 3.0 -moz-	5.0 3.1 -webkit-	10.5
box-shadow	9.0	10.0 4.0 -webkit-	4.0 3.5 -moz-	5.1 3.1 -webkit-	10.5
border-image	11.0	16.0 4.0 -webkit-	15.0 3.5 -moz-	6.0 3.1 -webkit-	15.0 11.0 -o-

CSS3 The border-radius Property - Rounded Corners

Adding rounded corners in CSS2 was tricky. We had to use different images for each corner.

In CSS3, creating rounded corners is easy.

In CSS3, the border-radius property is used to create rounded corners:

This box has rounded corners!

Example

Add rounded corners to a <div> element:

```
div {  
    border: 2px solid;  
    border-radius: 25px;  
}
```

[Try it yourself »](#)

CSS3 The box-shadow Property

In CSS3, the box-shadow property is used to add shadow to boxes:



Example

Add a box-shadow to a <div> element:

```
div {  
    box-shadow: 10px 10px 5px #888888;  
}
```

[Try it yourself »](#)

CSS3 The border-image Property

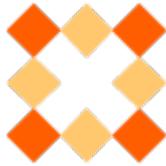
With the CSS3 border-image property you can use an image to create a border:





The `border-image` property allows you to specify an image as a border!

The original image used to create the border above:



Example

Use an image to create a border around a `<div>` element:

```
div {
    -webkit-border-image: url(border.png) 30 30 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30 30 round; /* Opera 11-12.1 */
    border-image: url(border.png) 30 30 round;
}
```

[Try it yourself »](#)

CSS3 Border Properties

Property	Description	CSS
<u>border-image</u>	A shorthand property for setting all the <code>border-image-*</code> properties	3
<u>border-radius</u>	A shorthand property for setting all the four <code>border-*radius</code> properties	3
<u>box-shadow</u>	Attaches one or more drop-shadows to the box	3

[◀ Previous](#)

[Next Chapter ▶](#)

CSS3 Backgrounds

[Previous](#)

[Next Chapter](#)

CSS3 Backgrounds

CSS3 contains several new background properties, which allow greater control of the background element.

In this chapter you will learn about the following background properties:

- background-size
- background-origin

You will also learn how to use multiple background images.



Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
background-size	9.0	4.0 1.0 -webkit-	4.0 3.6 -moz-	4.1 3.0 -webkit-	10.5 10.0 -o-
background-origin	9.0	1.0	4.0	3.0	10.5

CSS3 The background-size Property



The background-size property specifies the size of the background image.

Before CSS3, the background image size was determined by the actual size of the image. In CSS3

It is possible to specify the size of the background image, which allows us to re-use background images in different contexts.

You can specify the size in pixels or in percentages. If you specify the size as a percentage, the size is relative to the width and height of the parent element.

Example 1

Resize a background image:

```
div {
    background: url(img_flwr.gif);
    background-size: 80px 60px;
    background-repeat: no-repeat;
}
```

[Try it yourself »](#)

Example 2

Stretch the background image to completely fill the content area:

```
div {
    background: url(img_flwr.gif);
    background-size: 100% 100%;
    background-repeat: no-repeat;
}
```

[Try it yourself »](#)

CSS3 The background-origin Property

The background-origin property specifies the positioning area of the background images.

The background image can be placed within the content-box, padding-box, or border-box area.



Example

Position the background image within the content-box:

```
div {
    background: url(img_flwr.gif);
    background-repeat: no-repeat;
    background-size: 100% 100%;
    background-origin: content-box;
}
```

[Try it yourself »](#)

CSS3 Multiple Background Images

CSS3 allows you to use several background images for an element.



Example

Set two background images for the <body> element:

```
body {
    background: url(img_tree.gif), url(img_flwr.gif);
    background-size: 100% 100%;
    background-repeat: no-repeat;
}
```

[Try it yourself »](#)

CSS3 Background Properties

Property	Description	CSS
<u>background-clip</u>	Specifies the painting area of the background images	3
<u>background-origin</u>	Specifies the positioning area of the background images	3
<u>background-size</u>	Specifies the size of the background images	3

[◀ Previous](#)

[Next Chapter ▶](#)

CSS3 Gradients

[Previous](#)

[Next Chapter](#)

Gradient Background

CSS3 gradients let you display smooth transitions between two or more specified colors.

Earlier, you had to use images for these effects. However, by using CSS3 gradients you can reduce download time and bandwidth usage. In addition, elements with gradients look better when zoomed, because the gradient is generated by the browser.

CSS3 defines two types of gradients:

- **Linear Gradients (goes down/up/left/right/diagonally)**
- **Radial Gradients (defined by their center)**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
linear-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.1 -o-
radial-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.6 -o-
repeating-linear-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.1 -o-
repeating-radial-gradient	10.0	26.0 10.0 -webkit-	16.0 3.6 -moz-	6.1 5.1 -webkit-	12.1 11.6 -o-

CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

Example of Linear Gradient:



Syntax

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

Linear Gradient - Top to Bottom (this is default)

The following example shows a linear gradient that starts at the top. It starts red, transitioning to blue:

Example

A linear gradient from top to bottom:

```
#grad {
  background: -webkit-linear-gradient(red, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(red, blue); /* Standard syntax */
}
```

[Try it yourself »](#)

Linear Gradient - Left to Right

The following example shows a linear gradient that starts from the left. It starts red, transitioning to blue:

Example

A linear gradient from left to right:

```
#grad {
  background: -webkit-linear-gradient(left, red, blue); /* For Safari 5.1 to 6.0 */
```

```

background: -o-linear-gradient(right, red, blue); /* For Opera 11.1 to 12.0 */
background: -moz-linear-gradient(right, red, blue); /* For Firefox 3.6 to 15 */
background: linear-gradient(to right, red, blue); /* Standard syntax */
}

```

[Try it yourself »](#)

Linear Gradient - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to blue:

Example

A linear gradient that starts at top left (and goes to bottom right):

```

#grad {
  background: -webkit-linear-gradient(left top, red, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(bottom right, red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(bottom right, red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to bottom right, red, blue); /* Standard syntax */
}

```

[Try it yourself »](#)

Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.).

Syntax

```
background: linear-gradient(angle, color-stop1, color-stop2);
```

The angle is specified as an angle between a horizontal line and the gradient line, going counter-clockwise. In other words, 0deg creates a bottom to top gradient, while 90deg generates a left to right gradient.

The following example shows how to use angles on linear gradients:

Example

A linear gradient with a specified angle:

```
#grad {
  background: -webkit-linear-gradient(180deg, red, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(180deg, red, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(180deg, red, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(180deg, red, blue); /* Standard syntax */
}
```

[Try it yourself »](#)

Using Multiple Color Stops

The following example shows how to set multiple color stops:

Example

A linear gradient from top to bottom with multiple color stops:

```
#grad {
  background: -webkit-linear-gradient(red, green, blue); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(red, green, blue); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(red, green, blue); /* For Firefox 3.6 to 15 */
  background: linear-gradient(red, green, blue); /* Standard syntax */
}
```

[Try it yourself »](#)

The following example shows how to create a linear gradient with the color of the rainbow and some text:

Example

```
#grad {
  /* For Safari 5.1 to 6.0 */
  background: -webkit-linear-
gradient(left, red, orange, yellow, green, blue, indigo, violet);
  /* For Opera 11.1 to 12.0 */
  background: -o-linear-gradient(left, red, orange, yellow, green, blue, indigo, violet);
  /* For Fx 3.6 to 15 */
  background: -moz-linear-gradient(left, red, orange, yellow, green, blue, indigo, violet);
  /* Standard syntax */
  background: linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet);
}
```

[Try it yourself »](#)

Using Transparency

CSS3 gradients also support transparency, which can be used to create fading effects.

To add transparency, we use the `rgba()` function to define the color stops. The last parameter in the `rgba()` function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

The following example shows a linear gradient that starts from the left. It starts fully transparent, transitioning to full color red:

Example

A linear gradient from left to right, with transparency:

```
#grad {
  background: -webkit-linear-gradient(left,rgba(255,0,0,0),rgba(255,0,0,1)); /*Safari 5.1-6*/
  background: -o-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /*Opera 11.1-12*/
  background: -moz-linear-gradient(right,rgba(255,0,0,0),rgba(255,0,0,1)); /*Fx 3.6-15*/
  background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
/*Standard*/
}
```

[Try it yourself »](#)

Repeating a linear-gradient

The `repeating-linear-gradient()` function is used to repeat linear gradients:

Example

A repeating linear gradient:

```
#grad {
  /* Safari 5.1 to 6.0 */
  background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);
  /* Opera 11.1 to 12.0 */
  background: -o-repeating-linear-gradient(red, yellow 10%, green 20%);
  /* Firefox 3.6 to 15 */
  background: -moz-repeating-linear-gradient(red, yellow 10%, green 20%);
  /* Standard syntax */
  background: repeating-linear-gradient(red, yellow 10%, green 20%);
}
```

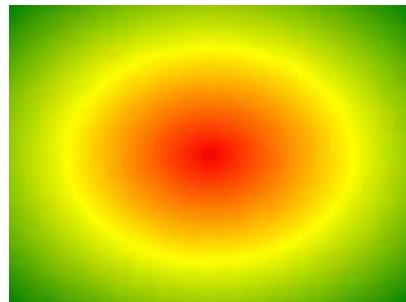
[Try it yourself »](#)

CSS3 Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

Example of Radial Gradient:



Syntax

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```

By default, shape is ellipse, size is farthest-corner, and position is center.

Radial Gradient - Evenly Spaced Color Stops (this is default)

Example

A radial gradient with evenly spaced color stops:

```
#grad {
  background: -webkit-radial-gradient(red, green, blue); /* Safari 5.1 to 6.0 */
  background: -o-radial-gradient(red, green, blue); /* For Opera 11.6 to 12.0 */
  background: -moz-radial-gradient(red, green, blue); /* For Firefox 3.6 to 15 */
  background: radial-gradient(red, green, blue); /* Standard syntax */
}
```

[Try it yourself »](#)

Radial Gradient - Differently Spaced Color Stops

Example

A radial gradient with differently spaced color stops:

```
#grad {
  background: -webkit-radial-gradient(red 5%, green 15%, blue 60%); /* Safari 5.1-6.0 */
  /*
  background: -o-radial-gradient(red 5%, green 15%, blue 60%); /* For Opera 11.6-12.0 */
  /*
  background: -moz-radial-gradient(red 5%, green 15%, blue 60%); /* For Firefox 3.6-15 */
  background: radial-gradient(red 5%, green 15%, blue 60%); /* Standard syntax */
```

}

[Try it yourself »](#)

Set Shape

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

Example

A radial gradient with the shape of a circle:

```
#grad {
  background: -webkit-radial-gradient(circle, red, yellow, green); /* Safari */
  background: -o-radial-gradient(circle, red, yellow, green); /* Opera 11.6 to 12.0 */
  background: -moz-radial-gradient(circle, red, yellow, green); /* Firefox 3.6 to 15 */
}
background: radial-gradient(circle, red, yellow, green); /* Standard syntax */
```

[Try it yourself »](#)

Use of Different Size Keywords

The size parameter defines the size of the gradient. It can take four values:

- closest-side
- farthest-side
- closest-corner
- farthest-corner

Example

A radial gradient with different size keywords:

```
#grad1 {
  /* Safari 5.1 to 6.0 */
  background: -webkit-radial-gradient(60% 55%, closest-side, blue, green, yellow, black);
  /* For Opera 11.6 to 12.0 */
  background: -o-radial-gradient(60% 55%, closest-side, blue, green, yellow, black);
  /* For Firefox 3.6 to 15 */
  background: -moz-radial-gradient(60% 55%, closest-side, blue, green, yellow, black);
  /* Standard syntax */
  background: radial-gradient(closest-side at 60% 55%, blue, green, yellow, black);
}
```

```
#grad2 {
/* Safari 5.1 to 6.0 */
background: -webkit-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* Opera 11.6 to 12.0 */
background: -o-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* For Firefox 3.6 to 15 */
background: -moz-radial-gradient(60% 55%, farthest-side,blue,green,yellow,black);
/* Standard syntax */
background: radial-gradient(farthest-side at 60% 55%,blue,green,yellow,black);
}
```

[Try it yourself »](#)

Repeating a radial-gradient

The repeating-radial-gradient() function is used to repeat radial gradients:

Example

A repeating radial gradient:

```
#grad {
/* For Safari 5.1 to 6.0 */
background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);
/* For Opera 11.6 to 12.0 */
background: -o-repeating-radial-gradient(red, yellow 10%, green 15%);
/* For Firefox 3.6 to 15 */
background: -moz-repeating-radial-gradient(red, yellow 10%, green 15%);
/* Standard syntax */
background: repeating-radial-gradient(red, yellow 10%, green 15%);
}
```

[Try it yourself »](#)

[◀ Previous](#)

[Next Chapter ▶](#)

CSS3 Text Effects

[Previous](#)

[Next Chapter](#)

CSS3 Text Effects

CSS3 contains several new text features.

In this chapter you will learn about the following text properties:

- text-shadow
- word-wrap

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-shadow	10.0	4.0	3.5	4.0	9.5
word-wrap	5.5	23.0	3.5	6.1	12.1

CSS3 Text Shadow

In CSS3, the text-shadow property applies shadow to text.

Text shadow effect!

You specify the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow:

Example

Add a shadow to a `<h1>` element:

```
h1 {
    text-shadow: 5px 5px 5px #FF0000;
}
```

[Try it yourself »](#)

CSS3 Word Wrapping

If a word is too long to fit within an area, it expands outside:

This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword.
The long word will break and wrap to the next line.

In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word:

This paragraph contains a very long word:
thisisaveryveryveryv
eryeveryverylongword
. The long word will break and wrap to the next line.

The CSS code is as follows:

Example

Allow long words to be able to break and wrap onto the next line:

```
p {  
    word-wrap: break-word;  
}
```

[Try it yourself »](#)

CSS3 Text Properties

Property	Description	CSS
<u>hanging-punctuation</u>	Specifies whether a punctuation character may be placed outside the line box	3

<u>punctuation-trim</u>	Specifies whether a punctuation character should be trimmed	3
<u>text-align-last</u>	Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"	3
<u>text-emphasis</u>	Applies emphasis marks, and the foreground color of the emphasis marks, to the element's text	3
<u>text-justify</u>	Specifies the justification method used when text-align is "justify"	3
<u>text-outline</u>	Specifies a text outline	3
<u>text-overflow</u>	Specifies what should happen when text overflows the containing element	3
<u>text-shadow</u>	Adds shadow to text	3
<u>text-wrap</u>	Specifies line breaking rules for text	3
<u>word-break</u>	Specifies line breaking rules for non-CJK scripts	3
<u>word-wrap</u>	Allows long, unbreakable words to be broken and wrap to the next line	3

 [Previous](#)

[Next Chapter](#) 

CSS3 Web Fonts

[Previous](#)

[Next Chapter](#)

With CSS3, web designers are no longer forced to use only "web-safe" fonts.

CSS3 Web Fonts - The @font-face Rule

Web fonts allow Web designers to use fonts that are not installed on the user's computer.

When you have found/bought the font you wish to use, just include the font file on your web server, and it will be automatically downloaded to the user when needed.

Your "own" fonts are defined within the CSS3 @font-face rule.

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
@font-face	9.0	4.0	3.5	3.2	10.0

Different Font Formats

TrueType Fonts (TTF)

TrueType is a font standard developed in the late 1980s, by Apple and Microsoft. TrueType is the most common font format for both the Mac OS and Microsoft Windows operating systems.

OpenType Fonts (OTF)

OpenType is a format for scalable computer fonts. It was built on TrueType, and is a registered trademark of Microsoft. OpenType fonts are used commonly today on the major computer platforms.

The Web Open Font Format (WOFF)

WOFF is a font format for use in web pages. It was developed in 2009, and is now a W3C Recommendation. WOFF is essentially OpenType or TrueType with compression and additional metadata. The goal is to support font distribution from a server to a client over a network with bandwidth constraints.

The Web Open Font Format (WOFF 2.0)

TrueType/OpenType font that provides better compression than WOFF 1.0.

SVG Fonts/Shapes

SVG fonts allow SVG to be used as glyphs when displaying text. The SVG 1.1 specification define a font module that allows the creation of fonts within an SVG document. You can also apply CSS to SVG documents, and the @font-face rule can be applied to text in SVG documents.

Embedded OpenType Fonts (EOT)

EOT fonts are a compact form of OpenType fonts designed by Microsoft for use as embedded fonts on web pages.

Browser Support for Font Formats

The numbers in the table specifies the first browser version that fully supports the font format.

Font format					
TTF/OTF	9.0*	4.0	3.5	3.1	10.0
WOFF	9.0	5.0	3.6	5.1	11.1
WOFF2	Not supported	36.0	35.0*	Not supported	26.0
SVG	Not supported	4.0	Not supported	3.2	9.0
EOT	6.0	Not supported	Not supported	Not supported	Not supported

*IE: The font format only works when set to be "installable".

*Firefox: Not supported by default, but can be enabled (need to set a flag to "true" to use WOFF2).

Using The Font You Want

In the CSS3 @font-face rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.



Tip: Always use lowercase letters for the font URL. Uppercase letters can give unexpected results in IE.

To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-family property:

Example

```
@font-face {
    font-family: myFirstFont;
    src: url(sansation_light.woff);
}

div {
    font-family: myFirstFont;
}
```

[Try it yourself »](#)

Using Bold Text

You must add another @font-face rule containing descriptors for bold text:

Example

```
@font-face {
    font-family: myFirstFont;
    src: url(sansation_bold.woff);
    font-weight: bold;
}
```

[Try it yourself »](#)

The file "sansation_bold.woff" is another font file, that contains the bold characters for the Sansation font.

Browsers will use this whenever a piece of text with the font-family "myFirstFont" should render as bold.

This way you can have many @font-face rules for the same font.

CSS3 Font Descriptors

The following table lists all the font descriptors that can be defined inside the @font-face rule:

Descriptor	Values	Description
font-family	<i>name</i>	Required. Defines a name for the font

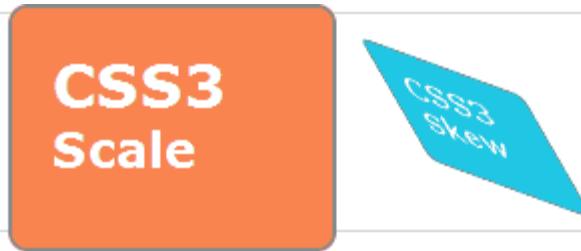
src	<i>URL</i>	Required. Defines the URL of the font file
font-stretch	normal condensed ultra-condensed extra-condensed semi-condensed expanded semi-expanded extra-expanded ultra-expanded	Optional. Defines how the font should be stretched. Default is "normal"
font-style	normal italic oblique	Optional. Defines how the font should be styled. Default is "normal"
font-weight	normal bold 100 200 300 400 500 600 700 800 900	Optional. Defines the boldness of the font. Default is "normal"
unicode-range	<i>unicode-range</i>	Optional. Defines the range of UNICODE characters the font supports. Default is "U+0-10FFFF"

[◀ Previous](#)[Next Chapter ▶](#)

CSS3 2D Transforms

[Previous](#)

[Next Chapter](#)



CSS3 Transforms

CSS3 transforms allow you to translate, rotate, scale, and skew elements.

A transformation is an effect that lets an element change shape, size and position.

CSS3 supports 2D and 3D transformations.

Browser Support for 2D Transforms

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -ms-, -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
transform	10.0 9.0 -ms-	36.0 4.0 -webkit-	16.0 3.5 -moz-	3.2 -webkit-	23.0 15.0 -webkit- 12.1 10.5 -o-
transform-origin (two-value syntax)	10.0 9.0 -ms-	36.0 4.0 -webkit-	16.0 3.5 -moz-	3.2 -webkit-	23.0 15.0 -webkit- 12.1 10.5 -o-

CSS3 2D Transforms

In this chapter you will learn about the following 2D transformation methods:

- translate()
- rotate()
- scale()
- skewX()
- skewY()
- matrix()



Tip: You will learn about 3D transformations in the next chapter.

The translate() Method



The `translate()` method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

The following example moves the `<div>` element 50 pixels to the right, and 100 pixels down from its current position:

Example

```
div {
    -ms-transform: translate(50px,100px); /* IE 9 */
    -webkit-transform: translate(50px,100px); /* Safari */
    transform: translate(50px,100px);
}
```

Try it yourself »

The rotate() Method



The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

The following example rotates the <div> element clockwise with 20 degrees:

Example

```
div {
    -ms-transform: rotate(20deg); /* IE 9 */
    -webkit-transform: rotate(20deg); /* Safari */
    transform: rotate(20deg);
}
```

[Try it yourself »](#)

Using negative values will rotate the element counter-clockwise.

The following example rotates the <div> element counter-clockwise with 20 degrees:

Example

```
div {
    -ms-transform: rotate(-20deg); /* IE 9 */
    -webkit-transform: rotate(-20deg); /* Safari */
    transform: rotate(-20deg);
}
```

[Try it yourself »](#)

The scale() Method



The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

The following example increases the <div> element to be two times of its original width, and three times of its original height:

Example

```
div {
    -ms-transform: scale(2,3); /* IE 9 */
```

```

    -webkit-transform: scale(2,3); /* Safari */
    transform: scale(2,3);
}

```

[Try it yourself »](#)

The following example decreases the <div> element to be half of its original width and height:

Example

```

div {
    -ms-transform: scale(0.5,0.5); /* IE 9 */
    -webkit-transform: scale(0.5,0.5); /* Safari */
    transform: scale(0.5,0.5);
}

```

[Try it yourself »](#)

The skewX() Method

The skewX() method skews an element along the X-axis by the given angle.

The following example skews the <div> element 20 degrees along the X-axis:

Example

```

div {
    -ms-transform: skewX(20deg); /* IE 9 */
    -webkit-transform: skewX(20deg); /* Safari */
    transform: skewX(20deg);
}

```

[Try it yourself »](#)

The skewY() Method

The skewY() method skews an element along the Y-axis by the given angle.

The following example skews the <div> element 20 degrees along the Y-axis:

Example

```

div {
    -ms-transform: skewY(20deg); /* IE 9 */
    -webkit-transform: skewY(20deg); /* Safari */
}

```

```
    transform: skewY(20deg);
}
```

[Try it yourself »](#)

The skew() Method

The skew() method skews an element along the X and Y-axis by the given angles.

The following example skews the <div> element 20 degrees along the X-axis, and 10 degrees along the Y-axis:

Example

```
div {
  -ms-transform: skew(20deg, 10deg); /* IE 9 */
  -webkit-transform: skew(20deg, 10deg); /* Safari */
  transform: skew(20deg, 10deg);
}
```

[Try it yourself »](#)

If the second parameter is not specified, it has a zero value. So, the following example skews the <div> element 20 degrees along the X-axis:

Example

```
div {
  -ms-transform: skew(20deg); /* IE 9 */
  -webkit-transform: skew(20deg); /* Safari */
  transform: skew(20deg);
}
```

[Try it yourself »](#)

The matrix() Method



The matrix() method combines all the 2D transform methods into one.

The `matrix()` method takes six parameters, containing mathematical functions, which allows you to rotate, scale, move (translate), and skew elements:

Example

```
div {
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari */
  transform: matrix(1, -0.3, 0, 1, 0, 0);
}
```

[Try it yourself »](#)

CSS3 Transform Properties

The following table lists all the 2D transform properties:

Property	Description
<code>transform</code>	Applies a 2D or 3D transformation to an element
<code>transform-origin</code>	Allows you to change the position on transformed elements

2D Transform Methods

Function	Description
<code>matrix(n,n,n,n,n,n)</code>	Defines a 2D transformation, using a matrix of six values
<code>translate(x,y)</code>	Defines a 2D translation, moving the element along the X- and the Y-axis
<code>translateX(n)</code>	Defines a 2D translation, moving the element along the X-axis
<code>translateY(n)</code>	Defines a 2D translation, moving the element along the Y-axis
<code>scale(x,y)</code>	Defines a 2D scale transformation, changing the element's width and height
<code>scaleX(n)</code>	Defines a 2D scale transformation, changing the element's width
<code>scaleY(n)</code>	Defines a 2D scale transformation, changing the element's height
<code>rotate(angle)</code>	Defines a 2D rotation, the angle is specified in the parameter

<code>skew(x-angle,y-angle)</code>	Defines a 2D skew transformation along the X- and the Y-axis
<code>skewX(angle)</code>	Defines a 2D skew transformation along the X-axis
<code>skewY(angle)</code>	Defines a 2D skew transformation along the Y-axis

[« Previous](#)[Next Chapter »](#)

CSS3 3D Transforms

[Previous](#)

[Next Chapter](#)

CSS3 3D Transforms

CSS3 allows you to format your elements using 3D transformations.

Mouse over the elements below to see the difference between a 2D and a 3D transformation:



Browser Support for 3D Transforms

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
transform	10.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-
transform-origin (three-value syntax)	10.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-
transform-style	11.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-
perspective	10.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-
perspective-origin	10.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-
backface-visibility	10.0	36.0 12.0 -webkit-	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-

CSS3 3D Transforms

In this chapter you will learn about the following 3D transformation methods:

- rotateX()
- rotateY()
- rotateZ()

The rotateX() Method



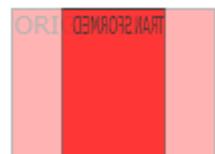
The rotateX() method rotates an element around its X-axis at a given degree:

Example

```
div {  
    -webkit-transform: rotateX(150deg); /* Safari */  
    transform: rotateX(150deg);  
}
```

[Try it yourself »](#)

The rotateY() Method



The rotateY() method rotates an element around its Y-axis at a given degree:

Example

```
div {  
    -webkit-transform: rotateY(130deg); /* Safari */  
    transform: rotateY(130deg);  
}
```

[Try it yourself »](#)

The rotateZ() Method

The rotateZ() method rotates an element around its Z-axis at a given degree:

Example

```
div {
  -webkit-transform: rotateZ(90deg); /* Safari */
  transform: rotateZ(90deg);
}
```

[Try it yourself »](#)

CSS3 Transform Properties

The following table lists all the 3D transform properties:

Property	Description
<u>transform</u>	Applies a 2D or 3D transformation to an element
<u>transform-origin</u>	Allows you to change the position on transformed elements
<u>transform-style</u>	Specifies how nested elements are rendered in 3D space
<u>perspective</u>	Specifies the perspective on how 3D elements are viewed
<u>perspective-origin</u>	Specifies the bottom position of 3D elements
<u>backface-visibility</u>	Defines whether or not an element should be visible when not facing the screen

3D Transform Methods

Function	Description
matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n)	Defines a 3D transformation, using a 4x4 matrix of 16 values
translate3d(x,y,z)	Defines a 3D translation
translateX(x)	Defines a 3D translation, using only the value for the X-axis

translateY(y)	Defines a 3D translation, using only the value for the Y-axis
translateZ(z)	Defines a 3D translation, using only the value for the Z-axis
scale3d(x,y,z)	Defines a 3D scale transformation
scaleX(x)	Defines a 3D scale transformation by giving a value for the X-axis
scaleY(y)	Defines a 3D scale transformation by giving a value for the Y-axis
scaleZ(z)	Defines a 3D scale transformation by giving a value for the Z-axis
rotate3d(x,y,z,angle)	Defines a 3D rotation
rotateX(angle)	Defines a 3D rotation along the X-axis
rotateY(angle)	Defines a 3D rotation along the Y-axis
rotateZ(angle)	Defines a 3D rotation along the Z-axis
perspective(n)	Defines a perspective view for a 3D transformed element

[◀ Previous](#)[Next Chapter ▶](#)

CSS3 Transitions

[Previous](#)

[Next Chapter](#)

CSS3 Transitions

CSS3 transitions allows you to change property values smoothly (from one value to another), over a given duration.

Example: Mouse over the element below to see a CSS3 transition effect



Browser Support for Transitions

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
transition	10.0	26.0 4.0 -webkit-	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-delay	10.0	26.0 4.0 -webkit-	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-duration	10.0	26.0 4.0 -webkit-	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-property	10.0	26.0 4.0 -webkit-	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-
transition-timing-function	10.0	26.0 4.0 -	16.0 4.0 -moz-	6.1 3.1 -	12.1 10.5 -o-

How to Use CSS3 Transitions?

To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

Note: If the duration part is not specified, the transition will have no effect, because the default value is 0.

The following example shows a 100px * 100px red <div> element. The <div> element has also specified a transition effect for the width property, with a duration of 2 seconds:

Example

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s; /* Safari */  
    transition: width 2s;  
}
```

The transition effect will start when the specified CSS property (width) changes value.

Now, let us specify a new value for the width property when a user mouses over the <div> element:

Example

```
div:hover {  
    width: 300px;  
}
```

Try it yourself »

Notice that when the cursor mouses out of the element, it will gradually change back to its original style.

Change Several Property Values

The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

Example

```
div {
    -webkit-transition: width 2s, height 4s; /* Safari */
    transition: width 2s, height 4s;
}
```

[Try it yourself »](#)

Specify the Speed Curve of the Transition

The transition-timing-function property specifies the speed curve of the transition effect.

The transition-timing-function property can have the following values:

- ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
- linear - specifies a transition effect with the same speed from start to end
- ease-in - specifies a transition effect with a slow start
- ease-out - specifies a transition effect with a slow end
- ease-in-out - specifies a transition effect with a slow start and end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

The following example shows some of the different speed curves that can be used:

Example

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

[Try it yourself »](#)

Delay the Transition Effect

The transition-delay property specifies a delay (in seconds) for the transition effect.

The following example has a 1 second delay before starting:

Example

```
div {
    -webkit-transition-delay: 1s; /* Safari */
    transition-delay: 1s;
```

}

[Try it yourself »](#)

Transition + Transformation

The following example also adds a transformation to the transition effect:

Example

```
div {  
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* Safari */  
    transition: width 2s, height 2s, transform 2s;  
}
```

[Try it yourself »](#)

More Transition Examples

The CSS3 transition properties can be specified one by one, like this:

Example

```
div {  
    transition-property: width;  
    transition-duration: 2s;  
    transition-timing-function: linear;  
    transition-delay: 1s;  
}
```

[Try it yourself »](#)

or by using the shorthand property transition:

Example

```
div {  
    transition: width 2s linear 1s;  
}
```

[Try it yourself »](#)

CSS3 Transition Properties

The following table lists all the transition properties:

Property	Description
<u>transition</u>	A shorthand property for setting the four transition properties into a single property
<u>transition-delay</u>	Specifies a delay (in seconds) for the transition effect
<u>transition-duration</u>	Specifies how many seconds or milliseconds a transition effect takes to complete
<u>transition-property</u>	Specifies the name of the CSS property the transition effect is for
<u>transition-timing-function</u>	Specifies the speed curve of the transition effect

 Previous

Next Chapter 

CSS3 Animations

[Previous](#)

[Next Chapter](#)

CSS3 Animations

CSS3 animations allows animation of most HTML elements without using JavaScript or Flash!

CSS3

Browser Support for Animations

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
@keyframes	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-
animation	10.0	4.0 -webkit-	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

What are CSS3 Animations?

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS3 animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

The @keyframes Rule

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

Example

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

[Try it yourself »](#)

Note: If the animation-duration property is not specified, the animation will have no effect, because the default value is 0.

In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

Example

```
/* The animation code */
@keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}
```

```
/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

[Try it yourself »](#)

The following example will change both the background-color and the position of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

Example

```
/* The animation code */
@keyframes example {
    0% {background-color: red; left:0px; top:0px;}
    25% {background-color: yellow; left:200px; top:0px;}
    50% {background-color: blue; left:200px; top:200px;}
    75% {background-color: green; left:0px; top:200px;}
    100% {background-color: red; left:0px; top:0px;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

[Try it yourself »](#)

Delay an Animation

The animation-delay property specifies a delay for the start of an animation.

The following example has a 2 seconds delay before starting the animation:

Example

```
div {
    width: 100px;
    height: 100px;
```

```
position: relative;
background-color: red;
animation-name: example;
animation-duration: 4s;
animation-delay: 2s;
}
```

[Try it yourself »](#)

Set How Many Times an Animation Should Run

The animation-iteration-count property specifies the number of times an animation should run.

The following example will run the animation 3 times before it stops:

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}
```

[Try it yourself »](#)

The following example uses the value "infinite" to make the animation continue for ever:

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
}
```

[Try it yourself »](#)

Run Animation in Reverse Direction or Alternate Cycles

The animation-direction property is used to let an animation run in reverse direction or alternate cycles.

The following example will run the animation in reverse direction:

Example

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-iteration-count: 3;
    animation-direction: reverse;
}
```

[Try it yourself »](#)

The following example uses the value "alternate" to make the animation first run forward, then backward, then forward:

Example

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-iteration-count: 3;
    animation-direction: alternate;
}
```

[Try it yourself »](#)

Specify the Speed Curve of the Animation

The animation-timing-function property specifies the speed curve of the animation.

The animation-timing-function property can have the following values:

- ease - specifies an animation with a slow start, then fast, then end slowly (this is default)
- linear - specifies an animation with the same speed from start to end
- ease-in - specifies an animation with a slow start

- ease-out - specifies an animation with a slow end
- ease-in-out - specifies an animation with a slow start and end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

The following example shows some of the different speed curves that can be used:

Example

```
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
```

[Try it yourself »](#)

Animation Shorthand Property

The example below uses six of the animation properties:

Example

```
div {
    animation-name: example;
    animation-duration: 5s;
    animation-timing-function: linear;
    animation-delay: 2s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
}
```

[Try it yourself »](#)

The same animation effect as above can be achieved by using the shorthand animation property:

Example

```
div {
    animation: example 5s linear 2s infinite alternate;
}
```

[Try it yourself »](#)

CSS3 Animation Properties

The following table lists the @keyframes rule and all the animation properties:

Property	Description
<u>@keyframes</u>	Specifies the animation code
<u>animation</u>	A shorthand property for setting all the animation properties (except animation-play-state and animation-fill-mode)
<u>animation-delay</u>	Specifies a delay for the start of an animation
<u>animation-direction</u>	Specifies whether an animation should play in reverse direction or alternate cycles
<u>animation-duration</u>	Specifies how many seconds or milliseconds an animation takes to complete one cycle
<u>animation-fill-mode</u>	Specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay)
<u>animation-iteration-count</u>	Specifies the number of times an animation should be played
<u>animation-name</u>	Specifies the name of the @keyframes animation
<u>animation-play-state</u>	Specifies whether the animation is running or paused
<u>animation-timing-function</u>	Specifies the speed curve of the animation

 [Previous](#)

[Next Chapter](#) 

CSS3 Multiple Columns

[Previous](#)

[Next Chapter](#)

CSS3 Multi-column Layout

The CSS multi-column layout allows easy definition of multiple columns of text - just like in newspapers:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper	suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio	dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
---	--	---

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-count	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-gap	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1
column-rule	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit-

					11.1
column-rule-color	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit 11.1
column-rule-style	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit 11.1
column-rule-width	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit 11.1
column-width	10.0	4.0 -webkit-	2.0 -moz-	3.1 -webkit-	15.0 -webkit 11.1

CSS3 Multiple-column Properties

In this chapter you will learn about the following multi-column properties:

- column-count
- column-gap
- column-rule

CSS3 Create Multiple Columns

The column-count property specifies the number of columns an element should be divided into:

Example

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

CSS3 Specify the Gap Between Columns

The column-gap property specifies the gap between the columns:

Example

Specify a 40 pixels gap between the columns:

```
div {
```

```

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}

```

[Try it yourself »](#)

CSS3 Column Rules

The column-rule property sets the width, style, and color of the rule between columns.

Example

Specify the width, style and color of the rule between columns:

```

div {
    -webkit-column-rule: 3px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 3px outset #ff00ff; /* Firefox */
    column-rule: 3px outset #ff00ff;
}

```

[Try it yourself »](#)

CSS3 Multiple Columns Properties

The following table lists all the multiple columns properties:

Property	Description	CSS
<u>column-count</u>	Specifies the number of columns an element should be divided into	3
<u>column-fill</u>	Specifies how to fill columns	3
<u>column-gap</u>	Specifies the gap between the columns	3
<u>column-rule</u>	A shorthand property for setting all the column-rule-* properties	3
<u>column-rule-color</u>	Specifies the color of the rule between columns	3
<u>column-rule-style</u>	Specifies the style of the rule between columns	3
<u>column-rule-width</u>	Specifies the width of the rule between columns	3

<u>column-span</u>	Specifies how many columns an element should span across	3
<u>column-width</u>	Specifies the width of the columns	3
<u>columns</u>	A shorthand property for setting column-width and column-count	3

[« Previous](#)[Next Chapter »](#)

CSS3 User Interface

[Previous](#)

[Next Chapter](#)

CSS3 User Interface

In CSS3, some of the new user interface features are resizing elements, box sizing, and outlining.

In this chapter you will learn about the following user interface properties:

- resize
- box-sizing
- outline-offset

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
resize	Not supported	4.0	5.0 4.0 -moz-	4.0	15.0
box-sizing	Partial from 8.0	10.0 4.0 -webkit-	29.0 2.0 -moz-	5.1 3.1 -webkit-	9.5
outline-offset	Not supported	4.0	5.0 4.0 -moz-	4.0	9.5

CSS3 Resizing

In CSS3, the resize property specifies whether or not an element should be resizable by the user.

This div element is resizable by the user (in Chrome, Firefox, Safari and Opera)

15+).

The CSS code is as follows:

Example

Specify that a <div> element should be resizable by the user:

```
div {
    resize: both;
    overflow: auto;
}
```

[Try it yourself »](#)

CSS3 Box Sizing

The box-sizing property is used to tell the browser what the sizing properties (width and height) should include.

Should they include the border-box or just the content-box which is the default value of the width and height properties.

For example, if you want two bordered boxes side by side, it can be achieved through setting box-sizing to "border-box". This forces the browser to render the box with the specified width and height, and place the border and padding inside the box.

Example

Specify two bordered boxes side by side:

```
div {
    -moz-box-sizing: border-box; /* Firefox */
    box-sizing: border-box;
    width: 50%;
    float: left;
}
```

[Try it yourself »](#)

CSS3 Outline Offset

The outline-offset property offsets an outline, and draws it beyond the border edge.

Outlines differ from borders in two ways:

- Outlines do not take up space

- Outlines may be non-rectangular

This div has an outline 15px outside the border edge.

The CSS code is as follows:

Example

Specify an outline 15px outside the border edge:

```
div {
    border: 2px solid black;
    outline: 2px solid red;
    outline-offset: 15px;
}
```

[Try it yourself »](#)

CSS3 User-interface Properties

The following table lists all the user-interface properties:

Property	Description	CSS
<u>appearance</u>	Allows you to make an element look like a standard user interface element	3
<u>box-sizing</u>	Allows you to define certain elements to fit an area in a certain way	3
<u>icon</u>	Provides the author the ability to style an element with an iconic equivalent	3
<u>nav-down</u>	Specifies where to navigate when using the arrow-down navigation key	3
<u>nav-index</u>	Specifies the tabbing order for an element	3
<u>nav-left</u>	Specifies where to navigate when using the arrow-left navigation key	3

<u>nav-right</u>	Specifies where to navigate when using the arrow-right navigation key	3
<u>nav-up</u>	Specifies where to navigate when using the arrow-up navigation key	3
<u>outline-offset</u>	Offsets an outline, and draws it beyond the border edge	3
<u>resize</u>	Specifies whether or not an element is resizable by the user	3

[« Previous](#)[Next Chapter »](#)

CSS Responsive Web Design

[Previous](#)

[Next Chapter](#)

What is Responsive Web Design?

Responsive web design is to create web sites that look good on all devices.

Responsive web design is not a program or a JavaScript.

Responsive web design is a must for mobile devices.



Creating Your Own Responsive Design

One way to create a responsive design, is to create it yourself, with simple CSS:

```
<!DOCTYPE html>
```

```

<html lang="en-US">
<head>
<style>
.article {
  float: left;
  margin: 5px;
  padding: 5px;
  width: 300px;
  height: 300px;
  border: 1px solid black;
}
</style>
</head>
<body>

<h1>W3Schools</h1>

<div class="article">
<h3>About W3Schools</h3>
<p>W3Schools is a web developer's site.  

It provides tutorials and references covering  

many aspects of web programming,  

including HTML, CSS, JavaScript, XML, SQL, PHP, ASP, etc.</p>
<p>W3schools contains thousands of code examples.  

By using an online editor,  

readers can edit examples experimentally.</p>
</div>

<div class="article">
<h3>Most Wanted Tutorials</h3>
<p>HTML5 Tutorial</p>
<p>How to build your web site with HTML5</p>
<p>CSS3 Tutorial</p>
<p>JavaScript Tutorial</p>
</div>

<div class="article">
<h3>About This Demo</h3>
<p>This demo is about responsive design.</p>
<p>Try to make the page smaller or wider,  

to see responsive design in action.</p>
</div>

</body>
</html>

```

[Try it yourself »](#)

Using Bootstrap

Another way to create a responsive design, is to use an already existing CSS framework.

Bootstrap is the most popular HTML, CSS, and JavaScript framework for responsive web design.

Bootstrap helps you develop sites that look nice at any size; screen, laptop, tablet, or phone:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
    href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>

<body>
<div class="container">

<div class="jumbotron">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>

<div class="row">
  <div class="col-md-4">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
    with a metropolitan area of over 13 million inhabitants.</p>
  </div>
  <div class="col-md-4">
    <h2>Paris</h2>
    <p>Paris is the capital and most populous city of France.</p>
  </div>
  <div class="col-md-4">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
    and the most populous metropolitan area in the world.</p>
  </div>
</div>
</div>
</body>
</html>
```

Try it yourself »

To learn more about Bootstrap read our [Bootstrap Tutorial](#).

« Previous

Next Chapter »

CSS Examples

[Previous](#)[Next Chapter](#)

CSS Selectors

[The element selector](#)[The id selector](#)[The class selector \(for all elements\)](#)[The class selector \(for only <p> elements\)](#)[Grouping selectors](#)[CSS selectors explained](#)

CSS Backgrounds

[Set the background color of a page](#)[Set the background color of different elements](#)[Set an image as the background of a page](#)[How to repeat a background image only horizontally](#)[How to position a background image](#)[A fixed background image \(this image will not scroll with the rest of the page\)](#)[All the background properties in one declaration](#)[Advanced background example](#)[Background properties explained](#)

CSS Text

[Set the text color of different elements](#)[Align the text](#)[Remove the line under links](#)[Decorate the text](#)[Control the letters in a text](#)[Indent text](#)[Specify the space between characters](#)[Specify the space between lines](#)[Set the text direction of an element](#)[Increase the white space between words](#)

[Disable text wrapping inside an element](#)

[Vertical alignment of an image inside text](#)

[Text properties explained](#)

CSS Fonts

[Set the font of a text](#)

[Set the size of the font](#)

[Set the size of the font in px](#)

[Set the size of the font in em](#)

[Set the size of the font in percent and em](#)

[Set the style of the font](#)

[Set the variant of the font](#)

[Set the boldness of the font](#)

[All the font properties in one declaration](#)

[Font properties explained](#)

CSS Links

[Add different colors to visited/unvisited links](#)

[Use of text-decoration on links](#)

[Specify a background color for links](#)

[Add other styles to hyperlinks](#)

[Advanced - Create link boxes](#)

[Link properties explained](#)

CSS Lists

[All the different list item markers in lists](#)

[Set an image as the list-item marker](#)

[Set an image as the list-item marker - Crossbrowser solution](#)

[All list properties in one declaration](#)

[List properties explained](#)

CSS Tables

[Specify a black border for table, th, and td elements](#)

[Use of border-collapse](#)

[Specify the width and height of a table](#)

[Set the horizontal alignment of content \(text-align\)](#)

[Set the vertical alignment of content \(vertical-align\)](#)

[Specify the padding for th and td elements](#)

[Specify the color of the table borders](#)

[Set the position of the table caption](#)

[Create a fancy table](#)

[Table properties explained](#)

CSS Box Model

[Specify an element with a total width of 250px](#)

[Box model explained](#)

CSS Border

[Set the width of the four borders](#)

[Set the width of the top border](#)

[Set the width of the bottom border](#)

[Set the width of the left border](#)

[Set the width of the right border](#)

[Set the style of the four borders](#)

[Set the style of the top border](#)

[Set the style of the bottom border](#)

[Set the style of the left border](#)

[Set the style of the right border](#)

[Set the color of the four borders](#)

[Set the color of the top border](#)

[Set the color of the bottom border](#)

[Set the color of the left border](#)

[Set the color of the right border](#)

[All the border properties in one declaration](#)

[Set different borders on each side](#)

[All the top border properties in one declaration](#)

[All the bottom border properties in one declaration](#)

[All the left border properties in one declaration](#)

[All the right border properties in one declaration](#)

[Border properties explained](#)

CSS Outline

[Draw a line around an element \(outline\)](#)

[Set the style of an outline](#)

[Set the color of an outline](#)

[Set the width of an outline](#)

[Outline properties explained](#)

CSS Margin

[Specify margins for an element](#)

[The margin shorthand property](#)

[Set the top margin of a text using a cm value](#)

[Set the bottom margin of a text using a percent value](#)

[Set the left margin of a text using a cm value](#)

[Margin properties explained](#)

CSS Padding

[Set the left padding of an element](#)

[Set the right padding of an element](#)

[Set the top padding of an element](#)

[Set the bottom padding of an element](#)

[All the padding properties in one declaration](#)

[Padding properties explained](#)

CSS Dimension

[Set the height of an image using a pixel value](#)

[Set the height of an image using percent](#)

[Set the width of an element using a pixel value](#)

[Set the maximum height of an element](#)

[Set the maximum width of an element using a pixel value](#)

[Set the maximum width of an element using percent](#)

[Set the minimum height of an element](#)

[Set the minimum width of an element using a pixel value](#)

[Set the minimum width of an element using percent](#)

[Dimension properties explained](#)

CSS Display

[How to hide an element \(visibility:hidden\)](#)

[How to not display an element \(display:none\)](#)

[How to display an element as an inline element](#)

[How to display an element as a block element](#)

[How to make a table element collapse](#)

Display properties explained

CSS Positioning

Position an element relative to the browser window

Position an element relative to its normal position

Position an element with an absolute value

Overlapping elements

Set the shape of an element

How to create a scroll bar when an element's content is too big to fit

How to set the browser to automatically handle overflow

Set the top edge of an image using a pixel value

Set the bottom edge of an image using a pixel value

Set the left edge of an image using a pixel value

Set the right edge of an image using a pixel value

Change the cursor

Positioning properties explained

CSS Floating

A simple use of the float property

An image with border and margins that floats to the right in a paragraph

An image with a caption that floats to the right

Let the first letter of a paragraph float to the left

Create an image gallery with the float property

Turning off float (using the clear property)

Creating a horizontal menu

Creating a homepage without tables

Float properties explained

CSS Aligning Elements

Center aligning with margin

Left/Right aligning with position

Left/Right aligning with position - Crossbrowser solution

Left/Right aligning with float

Left/Right aligning with float - Crossbrowser solution

Align properties explained

CSS Combinators

Descendant selector

Child selector

[Adjacent Sibling selector](#)

[General Sibling selector](#)

[Combinator selectors explained](#)

CSS Generated Content

[Insert the URL in parenthesis after each link with the content property](#)

[Numbering sections and sub-sections with "Section 1", "1.1", "1.2", etc.](#)

[Specify the quotation marks with the quotes property](#)

CSS Pseudo-classes

[Add different colors to a hyperlink](#)

[Add other styles to hyperlinks](#)

[Use of :focus](#)

[:first-child - match the first p element](#)

[:first-child - match the first i element in all p elements](#)

[:first-child - Match all i elements in all first child p elements](#)

[Use of :lang](#)

[Pseudo-classes explained](#)

CSS Pseudo-elements

[Make the first letter special in a text](#)

[Make the first line special in a text](#)

[Make the first letter and first line special](#)

[Use :before to insert some content before an element](#)

[Use :after to insert some content after an element](#)

[Pseudo-elements explained](#)

CSS Navigation Bars

[Fully styled vertical navigation bar](#)

[Fully styled horizontal navigation bar](#)

[Navigation bars explained](#)

CSS Image Gallery

[Image gallery](#)

[Image gallery explained](#)

CSS Image Opacity

[Creating transparent images - mouseover effect](#)

[Creating a transparent box with text on a background image](#)

[Image opacity explained](#)

CSS Image Sprites

[An image sprite](#)

[An image sprite - a navigation list](#)

[An image sprite with hover effect](#)

[Image sprites explained](#)

CSS Attribute Selectors

[Selects all <a> elements with a target attribute](#)

[Selects all <a> elements with a target="blank" attribute](#)

[Selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower"](#)

[Selects all elements with a class attribute value that begins with "top" \(must be whole word\)](#)

[Selects all elements with a class attribute value that begins with "top" \(must not be whole word\)](#)

[Selects all elements with a class attribute value that ends with "test"](#)

[Selects all elements with a class attribute value that contains "te"](#)

[Attribute selectors explained](#)

CSS3 Borders

[Add rounded corners to an element](#)

[Add a box-shadow to an element](#)

[Use an image to create a border around an element](#)

[CSS3 borders explained](#)

CSS3 Backgrounds

[Resize a background image](#)

[Stretch background image to completely fill the content area](#)

[Position the background image within the content-box](#)

[Set multiple background images for an element](#)

[CSS3 backgrounds explained](#)

CSS3 Gradients

[Linear Gradient - top to bottom](#)

[Linear Gradient - left to right](#)

[Linear Gradient - diagonal](#)

[Linear Gradient - with a specified angle](#)

[Linear Gradient - with multiple color stops](#)

[Linear Gradient - color of a rainbow + text](#)

[Linear Gradient - with transparency](#)

[Linear Gradient - a repeating linear gradient](#)

[Radial Gradient - evenly spaced color stops](#)

[Radial Gradient - differently spaced color stops](#)

[Radial Gradient - set shape](#)

[Radial Gradient - different size keywords](#)

[Radial Gradient - a repeating radial gradient](#)

[CSS3 gradients explained](#)

CSS3 Text Effects

[Add shadow to text](#)

[Allow long words to be able to break and wrap onto the next line](#)

[CSS3 text effects explained](#)

CSS3 Fonts

[Use your "own" fonts in @font-face rule](#)

[Use your "own" fonts in @font-face rule \(bold\)](#)

[CSS3 fonts explained](#)

CSS3 2D Transforms

[translate\(\) - move an element from its current position](#)

[rotate\(\) - rotate an element clockwise](#)

[rotate\(\) - rotate an element counter-clockwise](#)

[scale\(\) - increase an element](#)

[scale\(\) - decrease an element](#)

[skewX\(\) - skews an element along the X-axis](#)

[skewY\(\) - skews an element along the Y-axis](#)

[skew\(\) - skews an element along the X and Y-axis](#)

[matrix\(\) - rotate, scale, move, and skew an element](#)

[CSS3 2D transforms explained](#)

CSS3 3D Transforms

rotateX() - rotate an element around its X-axis at a given degree

rotateY() - rotate an element around its Y-axis at a given degree

rotateZ() - rotate an element around its Z-axis at a given degree

CSS3 3D transforms explained

CSS3 Transitions

Transition - change width of an element

Transition - change width and height of an element

Specify different speed curves for a transition

Specify a delay for a transition effect

Add a transformation to a transition effect

Specify all transition properties in one shorthand property

CSS3 transitions explained

CSS3 Animations

Bind an animation to an element

Animation - change background-color of an element

Animation - change background-color and position of an element

Delay an animation

Run animation 3 times before it stops

Run animation for ever

Run animation in reverse direction

Run animation in alternate cycles

Speed curves for animations

Animation shorthand property

CSS3 animations explained

CSS3 Multiple Columns

Divide the text in a div element into columns

Specify the gap between columns

Specify the width, style and color of the rule between columns

CSS3 multiple columns explained

CSS3 User Interface

resize property - Make an element resizable by the user

box-sizing property - Create two bordered boxes side by side

outline-offset property - offset an outline

CSS3 user interface explained

 Previous

Next Chapter 

CSS Quiz

[Previous](#)[Next Chapter »](#)

You can test your CSS skills with W3Schools' Quiz.

The Test

The test contains 20 questions and there is no time limit.

The test is not official, it's just a nice way to see how much you know, or don't know, about CSS.

Count Your Score

You will get 1 point for each correct answer. At the end of the Quiz, your total score will be displayed. Maximum score is 20 points.

Start the Quiz

Good luck!



W3Schools' Online Certification

The perfect solution for professionals who need to balance work, family, and career building.

More than 10 000 certificates already issued!

[Get Your Certificate »](#)

The [HTML Certificate](#) documents your knowledge of HTML.

The [HTML5 Certificate](#) documents your knowledge of advanced HTML5.

The [CSS Certificate](#) documents your knowledge of advanced CSS.

The [JavaScript Certificate](#) documents your knowledge of JavaScript and HTML DOM.

The [jQuery Certificate](#) documents your knowledge of jQuery.

The [PHP Certificate](#) documents your knowledge of PHP and SQL (MySQL).

The [XML Certificate](#) documents your knowledge of XML, XML DOM and XSLT.

 Previous

Next Chapter 

W3Schools CSS Certificate

[Previous](#)

[CSS Reference](#) [Next](#)



W3Schools offers an Online Certification Program.

The perfect solution for busy professionals who need to balance work, family, and career building.

More than 15 000 certificates already issued!

Document Your Skills

Knowledge is power, especially in the current job market. Documentation of your skills enables you to advance your career, or help you to start a new one.

Get a Certificate

Getting a certificate proves your commitment to upgrade your skills, gives you the credibility needed for more responsibilities, larger projects, and a higher salary.



[Get Your Certificate »](#)

How Does It Work?

- Study for free at W3Schools.com
- Study at your own speed
- Test your skills with W3Schools online quizzes
- Apply for your certificate by paying an exam fee
- Take your exam online, at any time, and from any location

 Previous

CSS Reference 

CSS Reference

[W3Schools Home](#)

[Next Reference](#)

 W3Schools' CSS reference is tested regularly with all major browsers.

CSS Properties

CSS Property Groups

- [Color](#)
- [Background and Borders](#)
- [Basic Box](#)
- [Flexible Box](#)
- [Text](#)
- [Text Decoration](#)
- [Fonts](#)
- [Writing Modes](#)
- [Table](#)
- [Lists and Counters](#)
- [Animation](#)
- [Transform](#)
- [Transition](#)
- [Basic User Interface](#)
- [Multi-column](#)
- [Paged Media](#)
- [Generated Content](#)
- [Filter Effects](#)
- [Image/Replaced Content](#)
- [Masking](#)
- [Speech](#)
- [Marquee](#)

The "CSS" column indicates in which CSS version the property is defined (CSS1, CSS2, or CSS3).

Color Properties

Property	Description	CSS
<u>color</u>	Sets the color of text	1
<u>opacity</u>	Sets the opacity level for an element	3

Background and Border Properties

Property	Description	CSS
<u>background</u>	Sets all the background properties in one declaration	1
<u>background-attachment</u>	Sets whether a background image is fixed or scrolls with the rest of the page	1
<u>background-color</u>	Sets the background color of an element	1
<u>background-image</u>	Sets the background image for an element	1
<u>background-position</u>	Sets the starting position of a background image	1
<u>background-repeat</u>	Sets how a background image will be repeated	1
<u>background-clip</u>	Specifies the painting area of the background	3
<u>background-origin</u>	Specifies the positioning area of the background images	3
<u>background-size</u>	Specifies the size of the background images	3
<u>border</u>	Sets all the border properties in one declaration	1
<u>border-bottom</u>	Sets all the bottom border properties in one declaration	1
<u>border-bottom-color</u>	Sets the color of the bottom border	1
<u>border-bottom-left-radius</u>	Defines the shape of the border of the bottom-left corner	3
<u>border-bottom-right-radius</u>	Defines the shape of the border of the bottom-right corner	3
<u>border-bottom-style</u>	Sets the style of the bottom border	1
<u>border-bottom-width</u>	Sets the width of the bottom border	1

<u>border-color</u>	Sets the color of the four borders	1
<u>border-image</u>	A shorthand property for setting all the border-image-* properties	3
<u>border-image-outset</u>	Specifies the amount by which the border image area extends beyond the border box	3
<u>border-image-repeat</u>	Specifies whether the image-border should be repeated, rounded or stretched	3
<u>border-image-slice</u>	Specifies the inward offsets of the image-border	3
<u>border-image-source</u>	Specifies an image to be used as a border	3
<u>border-image-width</u>	Specifies the widths of the image-border	3
<u>border-left</u>	Sets all the left border properties in one declaration	1
<u>border-left-color</u>	Sets the color of the left border	1
<u>border-left-style</u>	Sets the style of the left border	1
<u>border-left-width</u>	Sets the width of the left border	1
<u>border-radius</u>	A shorthand property for setting all the four border-*-radius properties	3
<u>border-right</u>	Sets all the right border properties in one declaration	1
<u>border-right-color</u>	Sets the color of the right border	1
<u>border-right-style</u>	Sets the style of the right border	1
<u>border-right-width</u>	Sets the width of the right border	1
<u>border-style</u>	Sets the style of the four borders	1
<u>border-top</u>	Sets all the top border properties in one declaration	1
<u>border-top-color</u>	Sets the color of the top border	1
<u>border-top-left-radius</u>	Defines the shape of the border of the top-left corner	3
<u>border-top-right-radius</u>	Defines the shape of the border of the top-right corner	3

<u>border-top-style</u>	Sets the style of the top border	1
<u>border-top-width</u>	Sets the width of the top border	1
<u>border-width</u>	Sets the width of the four borders	1
<u>box-decoration-break</u>	Sets the behaviour of the background and border of an element at page-break, or, for in-line elements, at line-break.	3
<u>box-shadow</u>	Attaches one or more drop-shadows to the box	3

Basic Box Properties

Property	Description	CSS
<u>bottom</u>	Specifies the bottom position of a positioned element	2
<u>clear</u>	Specifies which sides of an element where other floating elements are not allowed	1
<u>clip</u>	Clips an absolutely positioned element	2
<u>display</u>	Specifies how a certain HTML element should be displayed	1
<u>float</u>	Specifies whether or not a box should float	1
<u>height</u>	Sets the height of an element	1
<u>left</u>	Specifies the left position of a positioned element	2
<u>overflow</u>	Specifies what happens if content overflows an element's box	2
<u>overflow-x</u>	Specifies whether or not to clip the left/right edges of the content, if it overflows the element's content area	3
<u>overflow-y</u>	Specifies whether or not to clip the top/bottom edges of the content, if it overflows the element's content area	3
<u>padding</u>	Sets all the padding properties in one declaration	1
<u>padding-bottom</u>	Sets the bottom padding of an element	1
<u>padding-left</u>	Sets the left padding of an element	1
<u>padding-right</u>	Sets the right padding of an element	1

<u>padding-top</u>	Sets the top padding of an element	1
<u>position</u>	Specifies the type of positioning method used for an element (static, relative, absolute or fixed)	2
<u>right</u>	Specifies the right position of a positioned element	2
<u>top</u>	Specifies the top position of a positioned element	2
<u>visibility</u>	Specifies whether or not an element is visible	2
<u>width</u>	Sets the width of an element	1
<u>vertical-align</u>	Sets the vertical alignment of an element	1
<u>z-index</u>	Sets the stack order of a positioned element	2

Flexible Box Layout

Property	Description	CSS
<u>align-content</u>	Specifies the alignment between the lines inside a flexible container when the items do not use all available space.	3
<u>align-items</u>	Specifies the alignment for items inside a flexible container.	3
<u>align-self</u>	Specifies the alignment for selected items inside a flexible container.	3
<u>display</u>	Specifies how a certain HTML element should be displayed	1
<u>flex</u>	Specifies the length of the item, relative to the rest	3
<u>flex-basis</u>	Specifies the initial length of a flexible item	3
<u>flex-direction</u>	Specifies the direction of the flexible items	3
<u>flex-flow</u>	A shorthand property for the flex-direction and the flex-wrap properties	3
<u>flex-grow</u>	Specifies how much the item will grow relative to the rest	3
<u>flex-shrink</u>	Specifies how the item will shrink relative to the rest	3
<u>flex-wrap</u>	Specifies whether the flexible items should wrap or not	3

<u>justify-content</u>	Specifies the alignment between the items inside a flexible container when the items do not use all available space.	3
<u>margin</u>	Sets all the margin properties in one declaration	1
<u>margin-bottom</u>	Sets the bottom margin of an element	1
<u>margin-left</u>	Sets the left margin of an element	1
<u>margin-right</u>	Sets the right margin of an element	1
<u>margin-top</u>	Sets the top margin of an element	1
<u>max-height</u>	Sets the maximum height of an element	2
<u>max-width</u>	Sets the maximum width of an element	2
<u>min-height</u>	Sets the minimum height of an element	2
<u>min-width</u>	Sets the minimum width of an element	2
<u>order</u>	Sets the order of the flexible item, relative to the rest	3

Text Properties

Property	Description	CSS
<u>hanging-punctuation</u>	Specifies whether a punctuation character may be placed outside the line box	3
hyphens	Sets how to split words to improve the layout of paragraphs	3
<u>letter-spacing</u>	Increases or decreases the space between characters in a text	1
line-break	Specifies how/if to break lines	3
<u>line-height</u>	Sets the line height	1
overflow-wrap	Specifies whether or not the browser may break lines within words in order to prevent overflow (when a string is too long to fit its containing box)	3
<u>tab-size</u>	Specifies the length of the tab-character	3
<u>text-align</u>	Specifies the horizontal alignment of text	1

<u>text-align-last</u>	Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"	3
text-combine-upright	Specifies the combination of multiple characters into the space of a single character	3
<u>text-indent</u>	Specifies the indentation of the first line in a text-block	1
<u>text-justify</u>	Specifies the justification method used when text-align is "justify"	3
<u>text-transform</u>	Controls the capitalization of text	1
<u>white-space</u>	Specifies how white-space inside an element is handled	1
<u>word-break</u>	Specifies line breaking rules for non-CJK scripts	3
<u>word-spacing</u>	Increases or decreases the space between words in a text	1
<u>word-wrap</u>	Allows long, unbreakable words to be broken and wrap to the next line	3

Text Decoration Properties

Property	Description	CSS
<u>text-decoration</u>	Specifies the decoration added to text	1
<u>text-decoration-color</u>	Specifies the color of the text-decoration	3
<u>text-decoration-line</u>	Specifies the type of line in a text-decoration	3
<u>text-decoration-style</u>	Specifies the style of the line in a text decoration	3
<u>text-shadow</u>	Adds shadow to text	3
text-underline-position	Specifies the position of the underline which is set using the text-decoration property	3

Font Properties

Property	Description	CSS
<u>@font-face</u>	A rule that allows websites to download and use fonts	3

	other than the "web-safe" fonts	
<u>@font-feature-values</u>	Allows authors to use a common name in font-variant-alternate for feature activated differently in OpenType	3
<u>font</u>	Sets all the font properties in one declaration	1
<u>font-family</u>	Specifies the font family for text	1
font-feature-settings	Allows control over advanced typographic features in OpenType fonts	3
font-kerning	Controls the usage of the kerning information (how letters are spaced)	3
font-language-override	Controls the usage of language-specific glyphs in a typeface	3
<u>font-size</u>	Specifies the font size of text	1
<u>font-size-adjust</u>	Preserves the readability of text when font fallback occurs	3
<u>font-stretch</u>	Selects a normal, condensed, or expanded face from a font family	3
<u>font-style</u>	Specifies the font style for text	1
font-synthesis	Controls which missing typefaces (bold or italic) may be synthesized by the browser	3
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font	1
font-variant-alternates	Controls the usage of alternate glyphs associated to alternative names defined in @font-feature-values	3
font-variant-caps	Controls the usage of alternate glyphs for capital letters	3
font-variant-east-asian	Controls the usage of alternate glyphs for East Asian scripts (e.g Japanese and Chinese)	3
font-variant-ligatures	Controls which ligatures and contextual forms are used in textual content of the elements it applies to	3
font-variant-numeric	Controls the usage of alternate glyphs for numbers, fractions, and ordinal markers	3
font-variant-position	Controls the usage of alternate glyphs of smaller size	3

	positioned as superscript or subscript regarding the baseline of the font	
--	---	--

<u>font-weight</u>	Specifies the weight of a font	1
--------------------	--------------------------------	---

Writing Modes Properties

Property	Description	CSS
<u>direction</u>	Specifies the text direction/writing direction	2
text-orientation	Defines the orientation of the text in a line	3
text-combine-upright	Specifies the combination of multiple characters into the space of a single character	3
<u>unicode-bidi</u>	Used together with the <u>direction</u> property to set or return whether the text should be overridden to support multiple languages in the same document	2
writing-mode		3

Table Properties

Property	Description	CSS
<u>border-collapse</u>	Specifies whether or not table borders should be collapsed	2
<u>border-spacing</u>	Specifies the distance between the borders of adjacent cells	2
<u>caption-side</u>	Specifies the placement of a table caption	2
<u>empty-cells</u>	Specifies whether or not to display borders and background on empty cells in a table	2
<u>table-layout</u>	Sets the layout algorithm to be used for a table	2

Lists and Counters Properties

Property	Description	CSS
<u>counter-increment</u>	Increments one or more counters	2
<u>counter-reset</u>	Creates or resets one or more counters	2

<u>list-style</u>	Sets all the properties for a list in one declaration	1
<u>list-style-image</u>	Specifies an image as the list-item marker	1
<u>list-style-position</u>	Specifies if the list-item markers should appear inside or outside the content flow	1
<u>list-style-type</u>	Specifies the type of list-item marker	1

Animation Properties

Property	Description	CSS
<u>@keyframes</u>	Specifies the animation code	3
<u>animation</u>	A shorthand property for all the animation properties (except animation-play-state and animation-fill-mode)	3
<u>animation-delay</u>	Specifies a delay for the start of an animation	3
<u>animation-direction</u>	Specifies whether or not the animation should play in reverse on alternate cycles	3
<u>animation-duration</u>	Specifies how many seconds or milliseconds an animation takes to complete one cycle	3
<u>animation-fill-mode</u>	Specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay)	3
<u>animation-iteration-count</u>	Specifies the number of times an animation should be played	3
<u>animation-name</u>	Specifies the name of the @keyframes animation	3
<u>animation-play-state</u>	Specifies whether the animation is running or paused	3
<u>animation-timing-function</u>	Specifies the speed curve of an animation	3

Transform Properties

Property	Description	CSS
<u>backface-visibility</u>	Defines whether or not an element should be visible when not facing the screen	3

<u>perspective</u>	Specifies the perspective on how 3D elements are viewed	3
<u>perspective-origin</u>	Specifies the bottom position of 3D elements	3
<u>transform</u>	Applies a 2D or 3D transformation to an element	3
<u>transform-origin</u>	Allows you to change the position on transformed elements	3
<u>transform-style</u>	Specifies how nested elements are rendered in 3D space	3

Transitions Properties

Property	Description	CSS
<u>transition</u>	A shorthand property for setting the four transition properties	3
<u>transition-property</u>	Specifies the name of the CSS property the transition effect is for	3
<u>transition-duration</u>	Specifies how many seconds or milliseconds a transition effect takes to complete	3
<u>transition-timing-function</u>	Specifies the speed curve of the transition effect	3
<u>transition-delay</u>	Specifies when the transition effect will start	3

Basic User Interface Properties

Property	Description	CSS
<u>box-sizing</u>	Tells the browser what the sizing properties (width and height) should include	3
<u>content</u>	Used with the :before and :after pseudo-elements, to insert generated content	2
<u>cursor</u>	Specifies the type of cursor to be displayed	2
<u>icon</u>	Provides the author the ability to style an element with an iconic equivalent	3
<u>ime-mode</u>	Controls the state of the input method editor for text fields	3

<u>nav-down</u>	Specifies where to navigate when using the arrow-down navigation key	3
<u>nav-index</u>	Specifies the tabbing order for an element	3
<u>nav-left</u>	Specifies where to navigate when using the arrow-left navigation key	3
<u>nav-right</u>	Specifies where to navigate when using the arrow-right navigation key	3
<u>nav-up</u>	Specifies where to navigate when using the arrow-up navigation key	3
<u>outline</u>	Sets all the outline properties in one declaration	2
<u>outline-color</u>	Sets the color of an outline	2
<u>outline-offset</u>	Offsets an outline, and draws it beyond the border edge	3
<u>outline-style</u>	Sets the style of an outline	2
<u>outline-width</u>	Sets the width of an outline	2
<u>resize</u>	Specifies whether or not an element is resizable by the user	3
<u>text-overflow</u>	Specifies what should happen when text overflows the containing element	3

Multi-column Layout Properties

Property	Description	CSS
break-after	Specifies the page-, column-, or region-break behavior after the generated box	3
break-before	Specifies the page-, column-, or region-break behavior before the generated box	3
break-inside	Specifies the page-, column-, or region-break behavior inside the generated box	3
<u>column-count</u>	Specifies the number of columns an element should be divided into	3
<u>column-fill</u>	Specifies how to fill columns	3

<u>column-gap</u>	Specifies the gap between the columns	3
<u>column-rule</u>	A shorthand property for setting all the column-rule-* properties	3
<u>column-rule-color</u>	Specifies the color of the rule between columns	3
<u>column-rule-style</u>	Specifies the style of the rule between columns	3
<u>column-rule-width</u>	Specifies the width of the rule between columns	3
<u>column-span</u>	Specifies how many columns an element should span across	3
<u>column-width</u>	Specifies the width of the columns	3
<u>columns</u>	A shorthand property for setting column-width and column-count	3
widows	Sets the minimum number of lines that must be left at the top of a page when a page break occurs inside an element	2

Paged Media

Property	Description	CSS
orphans	Sets the minimum number of lines that must be left at the bottom of a page when a page break occurs inside an element	2
<u>page-break-after</u>	Sets the page-breaking behavior after an element	2
<u>page-break-before</u>	Sets the page-breaking behavior before an element	2
<u>page-break-inside</u>	Sets the page-breaking behavior inside an element	2

Generated Content for Paged Media

Property	Description	CSS
marks	Adds crop and/or cross marks to the document	3
<u>quotes</u>	Sets the type of quotation marks for embedded quotations	2

Filter Effects Properties

Property	Description	CSS
filter	Defines effects (e.g. blurring or color shifting) on an element before the element is displayed	3

Image Values and Replaced Content

Property	Description	CSS
image-orientation	Specifies a rotation in the right or clockwise direction that a user agent applies to an image (This property is likely going to be deprecated and its functionality moved to HTML)	3
image-rendering	Gives a hint to the browser about what aspects of an image are most important to preserve when the image is scaled	3
image-resolution	Specifies the intrinsic resolution of all raster images used in/on the element	3
object-fit	Specifies how the contents of a replaced element should be fitted to the box established by its used height and width	3
object-position	Specifies the alignment of the replaced element inside its box	3

Masking Properties

Property	Description	CSS
mask		3
mask-type		3

Speech Properties

Property	Description	CSS
mark	A shorthand property for setting the mark-before and	3

mark-after properties		
mark-after	Allows named markers to be attached to the audio stream	3
mark-before	Allows named markers to be attached to the audio stream	3
phonemes	Specifies a phonetic pronunciation for the text contained by the corresponding element	3
rest	A shorthand property for setting the rest-before and rest-after properties	3
rest-after	Specifies a rest or prosodic boundary to be observed after speaking an element's content	3
rest-before	Specifies a rest or prosodic boundary to be observed before speaking an element's content	3
voice-balance	Specifies the balance between left and right channels	3
voice-duration	Specifies how long it should take to render the selected element's content	3
voice-pitch	Specifies the average pitch (a frequency) of the speaking voice	3
voice-pitch-range	Specifies variation in average pitch	3
voice-rate	Controls the speaking rate	3
voice-stress	Indicates the strength of emphasis to be applied	3
voice-volume	Refers to the amplitude of the waveform output by the speech syntheses	3

Marquee Properties

Property	Description	CSS
marquee-direction	Sets the direction of the moving content	3
marquee-play-count	Sets how many times the content move	3
marquee-speed	Sets how fast the content scrolls	3
marquee-style	Sets the style of the moving content	3

 W3Schools Home

Next Reference 

CSS Selectors

In CSS, selectors are patterns used to select the element(s) you want to style.

Use our [CSS Selector Tester](#) to demonstrate the different selectors.

The "CSS" column indicates in which CSS version the property is defined (CSS1, CSS2, or CSS3).

Selector	Example	Example description	CSS
<u>.class</u>	.intro	Selects all elements with class="intro"	1
<u>#id</u>	#firstname	Selects the element with id="firstname"	1
<u>*</u>	*	Selects all elements	2
<u>element</u>	p	Selects all <p> elements	1
<u>element.element</u>	div, p	Selects all <div> elements and all <p> elements	1
<u>element element</u>	div p	Selects all <p> elements inside <div> elements	1
<u>element>element</u>	div > p	Selects all <p> elements where the parent is a <div> element	2
<u>element+element</u>	div + p	Selects all <p> elements that are placed immediately after <div> elements	2
<u>element1~element2</u>	p ~ ul	Selects every element that are preceded by a <p> element	3
<u>[attribute]</u>	[target]	Selects all elements with a target attribute	2
<u>[attribute=value]</u>	[target=_blank]	Selects all elements with target="_blank"	2

<u>[attribute~ = value]</u>	[title~ = flower]	Selects all elements with a title attribute containing the word "flower"	2
<u>[attribute =value]</u>	[lang =en]	Selects all elements with a lang attribute value starting with "en"	2
<u>[attribute^ = value]</u>	a[href^ = "https"]	Selects every <a> element whose href attribute value begins with "https"	3
<u>[attribute\$ = value]</u>	a[href\$ = ".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"	3
<u>[attribute* = value]</u>	a[href* = "w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"	3
<u>:active</u>	a: active	Selects the active link	1
<u>::after</u>	p:: after	Insert content after every <p> element	2
<u>::before</u>	p:: before	Insert content before the content of every <p> element	2
<u>:checked</u>	input: checked	Selects every checked <input> element	3
<u>:disabled</u>	input: disabled	Selects every disabled <input> element	3
<u>:empty</u>	p: empty	Selects every <p> element that has no children (including text nodes)	3
<u>:enabled</u>	input: enabled	Selects every enabled <input> element	3
<u>:first-child</u>	p: first-child	Selects every <p> element that is the first child of its parent	2
<u>::first-letter</u>	p:: first-letter	Selects the first letter of every <p> element	1
<u>::first-line</u>	p:: first-line	Selects the first line of every <p> element	1

<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent	3
<u>:focus</u>	input:focus	Selects the input element which has focus	2
<u>:hover</u>	a:hover	Selects links on mouse over	1
<u>:in-range</u>	input:in-range	Selects input elements with a value within a specified range	3
<u>:invalid</u>	input:invalid	Selects all input elements with an invalid value	3
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)	2
<u>:last-child</u>	p:last-child	Selects every <p> element that is the last child of its parent	3
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent	3
<u>:link</u>	a:link	Selects all unvisited links	1
<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element	3
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent	3
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child	3
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child	3
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent	3
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent	3
<u>:only-child</u>	p:only-child	Selects every <p> element that is	3

		the only child of its parent	
<u>:optional</u>	input:optional	Selects input elements with no "required" attribute	3
<u>:out-of-range</u>	input:out-of-range	Selects input elements with a value outside a specified range	3
<u>:read-only</u>	input:read-only	Selects input elements with the "readonly" attribute specified	3
<u>:read-write</u>	input:read-write	Selects input elements with the "readonly" attribute NOT specified	3
<u>:required</u>	input:required	Selects input elements with the "required" attribute specified	3
<u>:root</u>	:root	Selects the document's root element	3
<u>::selection</u>	::selection	Selects the portion of an element that is selected by a user	
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)	3
<u>:valid</u>	input:valid	Selects all input elements with a valid value	3
<u>:visited</u>	a:visited	Selects all visited links	1

[Previous](#)[Next Reference](#)

CSS Aural Reference

[Previous](#)

[Next Reference](#)

Aural Style Sheets

Aural style sheets use a combination of speech synthesis and sound effects to make the user listen to information, instead of reading information.

Aural presentation can be used:

- by blind people
- to help users learning to read
- to help users who have reading problems
- for home entertainment
- in the car
- by print-impaired communities

The aural presentation converts the document to plain text and feed this to a screen reader (a program that reads all the characters on the screen).

An example of an Aural style sheet:

```
h1, h2, h3, h4 {
    voice-family: male;
    richness: 80;
    cue-before: url("beep.au")
}
```

The example above will make the speech synthesizer play a sound, then speak the headers in a very rich male voice.

CSS Aural Reference

The "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
azimuth	Sets where the sound should come from	<i>angle</i> left-side	2

		far-left left center-left center center-right right far-right right-side behind leftwards rightwards	
cue	Sets the cue properties in one declaration	<i>cue-before</i> <i>cue-after</i>	2
cue-after	Specifies a sound to be played after speaking an element's content	none <i>url</i>	2
cue-before	Specifies a sound to be played before speaking an element's content	none <i>url</i>	2
elevation	Sets where the sound should come from	angle below level above higher lower	2
pause	Sets the pause properties in one declaration	<i>pause-before</i> <i>pause-after</i>	2
pause-after	Specifies a pause after speaking an element's content	<i>time</i> %	2
pause-before	Specifies a pause before speaking an element's content	<i>time</i> %	2
pitch	Specifies the speaking voice	<i>frequency</i> x-low low medium high x-high	2
pitch-range	Specifies the variation in the	<i>number</i>	2

	speaking voice. (Monotone voice or animated voice?)			
play-during	Specifies a sound to be played while speaking an element's content	auto none <i>url</i> mix repeat		2
richness	Specifies the richness of the speaking voice. (Rich voice or thin voice?)	<i>number</i>		2
speak	Specifies whether content will render aurally	normal none spell-out		2
speak-header	Specifies how to handle table headers. Should the headers be spoken before every cell, or only before a cell with a different header than the previous cell	always once		2
speak-numeral	Specifies how to speak numbers	digits continuous		2
speak-punctuation	Specifies how to speak punctuation characters	none code		2
speech-rate	Specifies the speed of the speaking	<i>number</i> x-slow slow medium fast x-fast faster slower		2
stress	Specifies the "stress" in the speaking voice	<i>number</i>		2
voice-family	Specifies the voice family of the speaking	<i>specific-voice</i> <i>generic-voice</i>		2
volume	Specifies the volume of the speaking	<i>number</i> %		2

silent
x-soft
soft
medium
loud
x-loud

 Previous

Next Reference 

CSS Web Safe Font Combinations

[Previous](#)

[Next Reference](#)

Commonly Used Font Combinations

The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems. If the browser does not support the first font, it tries the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available:

Example

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

[Try it yourself »](#)

Below are some commonly used font combinations, organized by generic family.

Serif Fonts

font-family	Example text
Georgia, serif	This is a heading This is a paragraph
"Palatino Linotype", "Book Antiqua", Palatino, serif	This is a heading This is a paragraph
"Times New Roman", Times, serif	This is a heading This is a paragraph

Sans-Serif Fonts

font-family	Example text
Arial, Helvetica, sans-serif	This is a heading This is a paragraph
"Arial Black", Gadget, sans-serif	<i>This is a heading</i> <i>This is a paragraph</i>
"Comic Sans MS", cursive, sans-serif	This is a heading <i>This is a paragraph</i>
Impact, Charcoal, sans-serif	This is a heading This is a paragraph
"Lucida Sans Unicode", "Lucida Grande", sans-serif	This is a heading This is a paragraph
Tahoma, Geneva, sans-serif	This is a heading This is a paragraph
"Trebuchet MS", Helvetica, sans-serif	This is a heading This is a paragraph
Verdana, Geneva, sans-serif	This is a heading This is a paragraph

Monospace Fonts

font-family	Example text
"Courier New", Courier, monospace	This is a heading This is a paragraph
"Lucida Console", Monaco, monospace	This is a heading This is a paragraph

 [Previous](#)

[Next Reference](#) 

CSS Units

[Previous](#)

[Next Reference](#)

CSS Units

CSS has several different units for expressing a length.

Many CSS properties take "length" values, such as width, margin, padding, font-size, border-width, etc.

Length is a number followed by a length unit, such as 10px, 2em, etc.

A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.

For some CSS properties, negative lengths are allowed.

There are two types of length units: relative and absolute.

Browser Support

The numbers in the table specify the first browser version that fully supports the length unit.

Length Unit					
em, ex, %, px, cm, mm, in, pt, pc	1.0	3.0	1.0	1.0	3.5
ch	27.0	9.0	1.0	7.0	20.0
rem	4.0	9.0	3.6	4.1	11.6
vh, vw	20.0	9.0	19.0	6.0	20.0
vmin	20.0	9.0*	19.0	6.0	20.0
vmax	26.0	Not supported	19.0	Not supported	20.0

Note: Internet Explorer 9 supports vmin with the non-standard name: vm.

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

Unit	Description	
em	Relative to the font-size of the element (2em means 2 times the size of the current font)	Try it
ex	Relative to the x-height of the current font (rarely used)	Try it
ch	Relative to width of the "0" (zero)	
rem	Relative to font-size of the root element	
vw	Relative to 1% of the width of the viewport	
vh	Relative to 1% of the height of the viewport	
vmin	Relative to 1% of viewport's smaller dimension	
vmax	Relative to 1% of viewport's larger dimension	
%		
px	pixels (1px = 1/96th of 1in)	Try it



Tip: The em and rem units are practical in creating perfectly scalable layout!

Absolute Lengths

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit	Description	
cm	centimeters	Try it

mm	millimeters	Try it
in	inches (1in = 96px = 2.54cm)	Try it
pt	points (1pt = 1/72 of 1in)	Try it
pc	picas (1pc = 12 pt)	Try it

[« Previous](#)[Next Reference »](#)

PX to EM Conversion

[Previous](#)[Next Reference](#)

PX to EM Converter

- Set a default pixel size (usually 16px)
- Then, convert a pixel value to em, based on the default pixel size
- Or, convert an em value to pixels, based on the default pixel size

Set a default pixel size:

px

Convert PX to EM:

px

Convert EM to PX:

em

Result:

Body Font Size

In the table below, select a body font size in pixels (px) to display a complete "px to em and percent" conversion table.

Tip: The default font size is usually 16px.

px	em	percent
5px	0.2500em	25.00%
6px	0.3000em	30.00%
7px	0.3500em	35.00%
8px	0.4000em	40.00%
9px	0.4500em	45.00%
10px	0.5000em	50.00%
11px	0.5500em	55.00%
12px	0.6000em	60.00%
13px	0.6500em	65.00%
14px	0.7000em	70.00%
15px	0.7500em	75.00%
16px	0.8000em	80.00%
17px	0.8500em	85.00%
18px	0.9000em	90.00%
19px	0.9500em	95.00%
20px	1.0000em	100.00%
21px	1.0500em	105.00%
22px	1.1000em	110.00%
23px	1.1500em	115.00%
24px	1.2000em	120.00%
25px	1.2500em	125.00%

What is the difference between PX, EM and Percent?

 Pixel is a static measurement, while percent and EM are relative measurements. The size of an EM or percent depends on its parent. If the text size of body is 16 pixels, then 150% or 1.5 EM will be 24 pixels ($1.5 * 16$). Look at [CSS Units](#) for more measurement units.

[« Previous](#)[Next Reference »](#)

CSS Colors

[Previous](#)

[Next Reference](#)

Colors are displayed combining RED, GREEN, and BLUE light.

Color Values

CSS colors are defined using a hexadecimal (hex) notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one of the light sources is 0 (hex 00). The highest value is 255 (hex FF).

Hex values are written as 3 double digit numbers, starting with a # sign.

Color Examples

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#COCOCO	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

[Try it yourself »](#)

16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256).

Most modern monitors are capable of displaying at least 16384 different colors.

If you look at the color table below, you will see the result of varying the red light from 0 to 255, while keeping the green and blue light at zero.

To see a full list of color mixes when the red light varies from 0 to 255, click on one of the hex or rgb values below.

Red Light	HEX	RGB
	#000000	rgb(0,0,0)
	#080000	rgb(8,0,0)
	#100000	rgb(16,0,0)
	#180000	rgb(24,0,0)
	#200000	rgb(32,0,0)
	#280000	rgb(40,0,0)
	#300000	rgb(48,0,0)
	#380000	rgb(56,0,0)
	#400000	rgb(64,0,0)
	#480000	rgb(72,0,0)
	#500000	rgb(80,0,0)
	#580000	rgb(88,0,0)
	#600000	rgb(96,0,0)
	#680000	rgb(104,0,0)
	#700000	rgb(112,0,0)
	#780000	rgb(120,0,0)
	#800000	rgb(128,0,0)

	<u>#880000</u>	<u>rgb(136,0,0)</u>
	<u>#900000</u>	<u>rgb(144,0,0)</u>
	<u>#980000</u>	<u>rgb(152,0,0)</u>
	<u>#A00000</u>	<u>rgb(160,0,0)</u>
	<u>#A80000</u>	<u>rgb(168,0,0)</u>
	<u>#B00000</u>	<u>rgb(176,0,0)</u>
	<u>#B80000</u>	<u>rgb(184,0,0)</u>
	<u>#C00000</u>	<u>rgb(192,0,0)</u>
	<u>#C80000</u>	<u>rgb(200,0,0)</u>
	<u>#D00000</u>	<u>rgb(208,0,0)</u>
	<u>#D80000</u>	<u>rgb(216,0,0)</u>
	<u>#E00000</u>	<u>rgb(224,0,0)</u>
	<u>#E80000</u>	<u>rgb(232,0,0)</u>
	<u>#F00000</u>	<u>rgb(240,0,0)</u>
	<u>#F80000</u>	<u>rgb(248,0,0)</u>
	<u>#FF0000</u>	<u>rgb(255,0,0)</u>

Shades of Gray

Gray colors are displayed using an equal amount of power to all of the light sources. To make it easier for you to select the right gray color we have compiled a table of gray shades for you:

Gray Shades	HEX	RGB
	#000000	rgb(0,0,0)
	#080808	rgb(8,8,8)
	#101010	rgb(16,16,16)
	#181818	rgb(24,24,24)

	#202020	rgb(32,32,32)
	#282828	rgb(40,40,40)
	#303030	rgb(48,48,48)
	#383838	rgb(56,56,56)
	#404040	rgb(64,64,64)
	#484848	rgb(72,72,72)
	#505050	rgb(80,80,80)
	#585858	rgb(88,88,88)
	#606060	rgb(96,96,96)
	#686868	rgb(104,104,104)
	#707070	rgb(112,112,112)
	#787878	rgb(120,120,120)
	#808080	rgb(128,128,128)
	#888888	rgb(136,136,136)
	#909090	rgb(144,144,144)
	#989898	rgb(152,152,152)
	#AOAOAO	rgb(160,160,160)
	#A8A8A8	rgb(168,168,168)
	#B0B0B0	rgb(176,176,176)
	#B8B8B8	rgb(184,184,184)
	#COCOCO	rgb(192,192,192)
	#C8C8C8	rgb(200,200,200)
	#D0D0D0	rgb(208,208,208)
	#D8D8D8	rgb(216,216,216)
	#E0E0E0	rgb(224,224,224)

	#E8E8E8	rgb(232,232,232)
	#FOFOFO	rgb(240,240,240)
	#F8F8F8	rgb(248,248,248)
	#FFFFFF	rgb(255,255,255)

Web Safe Colors?

Some years ago, when computers supported max 256 different colors, a list of 216 "Web Safe Colors" was suggested as a Web standard, reserving 40 fixed system colors.

This is not important now, since most computers can display millions of different colors, but the choice is left to you.

The 216 cross-browser color palette was created to ensure that all computers would display the colors correctly when running a 256 color palette:

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF

669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

[Previous](#)[Next Reference](#)

CSS Legal Color Values

[Previous](#)[Next Reference](#)

CSS Colors

Colors in CSS can be specified by the following methods:

- Hexadecimal colors
- RGB colors
- RGBA colors
- HSL colors
- HSLA colors
- Predefined/Cross-browser color names

Hexadecimal Colors

Hexadecimal color values are supported in all major browsers.

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color. All values must be between 0 and FF.

For example, the #0000ff value is rendered as blue, because the blue component is set to its highest value (ff) and the others are set to 0.

Example

Define different HEX colors:

```
#p1 {background-color: #ff0000;} /* red */  
#p2 {background-color: #00ff00;} /* green */  
#p3 {background-color: #0000ff;} /* blue */
```

[Try it yourself »](#)

RGB Colors

RGB color values are supported in all major browsers.

An RGB color value is specified with: `rgb(red, green, blue)`. Each parameter (red, green, and blue) defines the intensity of the color and can be an integer between 0 and 255 or a percentage value (from 0% to 100%).

For example, the `rgb(0,0,255)` value is rendered as blue, because the blue parameter is set to its highest value (255) and the others are set to 0.

Also, the following values define equal color: `rgb(0,0,255)` and `rgb(0%,0%,100%)`.

Example

Define different RGB colors:

```
#p1 {background-color: rgb(255, 0, 0);} /* red */
#p2 {background-color: rgb(0, 255, 0);} /* green */
#p3 {background-color: rgb(0, 0, 255);} /* blue */
```

[Try it yourself »](#)

RGBA Colors

RGBA color values are supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+.

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity of the object.

An RGBA color value is specified with: `rgba(red, green, blue, alpha)`. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

Example

Define different RGB colors with opacity:

```
#p1 {background-color: rgba(255, 0, 0, 0.3);} /* red with opacity */
#p2 {background-color: rgba(0, 255, 0, 0.3);} /* green with opacity */
#p3 {background-color: rgba(0, 0, 255, 0.3);} /* blue with opacity */
```

[Try it yourself »](#)

HSL Colors

HSL color values are supported in IE9+, Firefox, Chrome, Safari, and in Opera 10+.

HSL stands for hue, saturation, and lightness - and represents a cylindrical-coordinate representation of colors.

An HSL color value is specified with: `hsl(hue, saturation, lightness)`.

Hue is a degree on the color wheel (from 0 to 360) - 0 (or 360) is red, 120 is green, 240 is blue. Saturation is a percentage value; 0% means a shade of gray and 100% is the full color. Lightness is also a percentage; 0% is black, 100% is white.

Example

Define different HSL colors:

```
#p1 {background-color: hsl(120, 100%, 50%);} /* green */
#p2 {background-color: hsl(120, 100%, 75%);} /* light green */
#p3 {background-color: hsl(120, 100%, 25%);} /* dark green */
#p4 {background-color: hsl(120, 60%, 70%);} /* pastel green */
```

[Try it yourself »](#)

HSLA Colors

HSLA color values are supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+.

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity of the object.

An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

Example

Define different HSL colors with opacity:

```
#p1 {background-color: hsla(120, 100%, 50%, 0.3);} /* green with opacity */
#p2 {background-color: hsla(120, 100%, 75%, 0.3);} /* light green with opacity */
#p3 {background-color: hsla(120, 100%, 25%, 0.3);} /* dark green with opacity */
#p4 {background-color: hsla(120, 60%, 70%, 0.3);} /* pastel green with opacity */
```

[Try it yourself »](#)

Predefined/Cross-browser Color Names

140 color names are predefined in the HTML and CSS color specification. Look at our table of [predefined color names](#).

[◀ Previous](#)

[Next Reference ▶](#)

CSS Color Names

[Previous](#)
[Next Reference](#)

Color Names Supported by All Browsers

140 color names are defined in the HTML and CSS color specification (17 standard colors plus 123 more). The table below lists them all, along with their hexadecimal values.

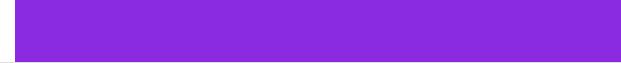
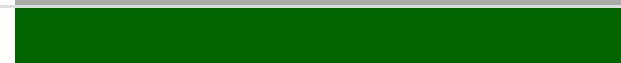
 **Tip:** The 17 standard colors are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, and yellow.

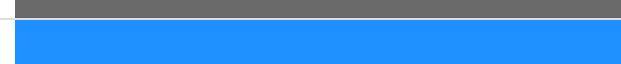
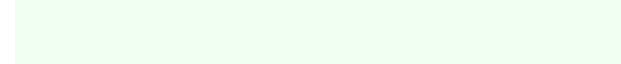
Sorted by Color Name

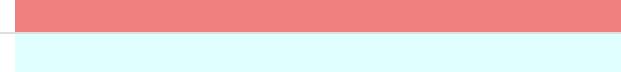
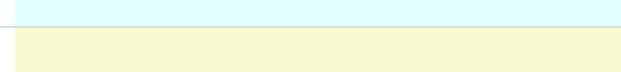
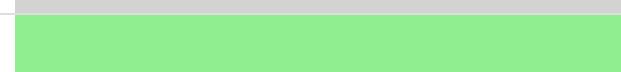
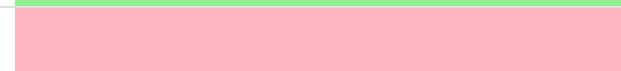
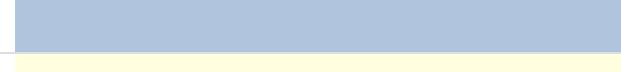
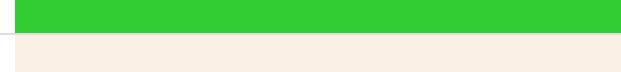
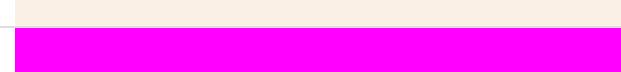
[Colors sorted by HEX values](#)

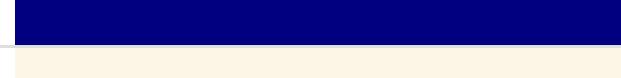
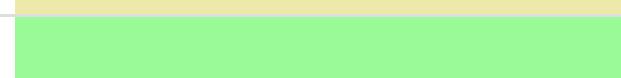
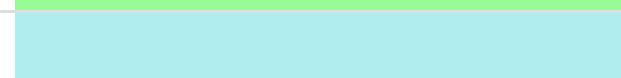
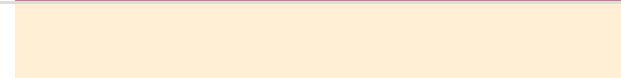
Click on a color name (or a hex value) to view the color as the background-color along with different text colors:

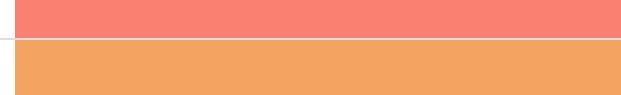
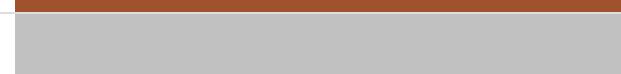
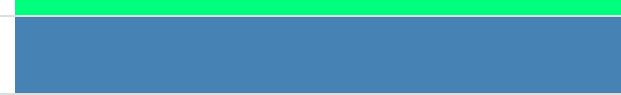
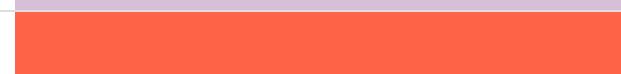
Color Name	HEX	Color	Shades	Mix
AliceBlue	#F0F8FF		Shades	Mix
AntiqueWhite	#FAEBD7		Shades	Mix
Aqua	#00FFFF		Shades	Mix
Aquamarine	#7FFFAD		Shades	Mix
Azure	#FOFFFF		Shades	Mix
Beige	#F5F5DC		Shades	Mix
Bisque	#FFE4C4		Shades	Mix
Black	#000000		Shades	Mix
BlanchedAlmond	#FFEBCD		Shades	Mix

Blue	#0000FF		Shades	Mix
BlueViolet	#8A2BE2		Shades	Mix
Brown	#A52A2A		Shades	Mix
BurlyWood	#DEB887		Shades	Mix
CadetBlue	#5F9EA0		Shades	Mix
Chartreuse	#7FFF00		Shades	Mix
Chocolate	#D2691E		Shades	Mix
Coral	#FF7F50		Shades	Mix
CornflowerBlue	#6495ED		Shades	Mix
Cornsilk	#FFF8DC		Shades	Mix
Crimson	#DC143C		Shades	Mix
Cyan	#00FFFF		Shades	Mix
DarkBlue	#00008B		Shades	Mix
DarkCyan	#008B8B		Shades	Mix
DarkGoldenRod	#B8860B		Shades	Mix
DarkGray	#A9A9A9		Shades	Mix
DarkGreen	#006400		Shades	Mix
DarkKhaki	#BDB76B		Shades	Mix
DarkMagenta	#8B008B		Shades	Mix
DarkOliveGreen	#556B2F		Shades	Mix
DarkOrange	#FF8C00		Shades	Mix
DarkOrchid	#9932CC		Shades	Mix
DarkRed	#8B0000		Shades	Mix
DarkSalmon	#E9967A		Shades	Mix
DarkSeaGreen	#8FBC8F		Shades	Mix

DarkSlateBlue	#483D8B		Shades	Mix
DarkSlateGray	#2F4F4F		Shades	Mix
DarkTurquoise	#00CED1		Shades	Mix
DarkViolet	#9400D3		Shades	Mix
DeepPink	#FF1493		Shades	Mix
DeepSkyBlue	#00BFFF		Shades	Mix
DimGray	#696969		Shades	Mix
DodgerBlue	#1E90FF		Shades	Mix
FireBrick	#B22222		Shades	Mix
FloralWhite	#FFFFA0		Shades	Mix
ForestGreen	#228B22		Shades	Mix
Fuchsia	#FF00FF		Shades	Mix
Gainsboro	#DCDCDC		Shades	Mix
GhostWhite	#F8F8FF		Shades	Mix
Gold	#FFD700		Shades	Mix
GoldenRod	#DAA520		Shades	Mix
Gray	#808080		Shades	Mix
Green	#008000		Shades	Mix
GreenYellow	#ADFF2F		Shades	Mix
HoneyDew	#F0FFF0		Shades	Mix
HotPink	#FF69B4		Shades	Mix
IndianRed	#CD5C5C		Shades	Mix
Indigo	#4B0082		Shades	Mix
Ivory	#FFFFFF		Shades	Mix
Khaki	#FOE68C		Shades	Mix

Lavender	#E6E6FA		Shades	Mix
LavenderBlush	#FFF0F5		Shades	Mix
LawnGreen	#7CFC00		Shades	Mix
LemonChiffon	#FFFACD		Shades	Mix
LightBlue	#ADD8E6		Shades	Mix
LightCoral	#F08080		Shades	Mix
LightCyan	#E0FFFF		Shades	Mix
LightGoldenRodYellow	#FAFAD2		Shades	Mix
LightGray	#D3D3D3		Shades	Mix
LightGreen	#90EE90		Shades	Mix
LightPink	#FFB6C1		Shades	Mix
LightSalmon	#FFA07A		Shades	Mix
LightSeaGreen	#20B2AA		Shades	Mix
LightSkyBlue	#87CEFA		Shades	Mix
LightSlateGray	#778899		Shades	Mix
LightSteelBlue	#B0C4DE		Shades	Mix
LightYellow	#FFFFE0		Shades	Mix
Lime	#00FF00		Shades	Mix
LimeGreen	#32CD32		Shades	Mix
Linen	#FAFOE6		Shades	Mix
Magenta	#FF00FF		Shades	Mix
Maroon	#800000		Shades	Mix
MediumAquaMarine	#66CDAA		Shades	Mix
MediumBlue	#0000CD		Shades	Mix
MediumOrchid	#BA55D3		Shades	Mix

MediumPurple	#9370DB		Shades	Mix
MediumSeaGreen	#3CB371		Shades	Mix
MediumSlateBlue	#7B68EE		Shades	Mix
MediumSpringGreen	#00FA9A		Shades	Mix
MediumTurquoise	#48D1CC		Shades	Mix
MediumVioletRed	#C71585		Shades	Mix
MidnightBlue	#191970		Shades	Mix
MintCream	#F5FFFF		Shades	Mix
MistyRose	#FFE4E1		Shades	Mix
Moccasin	#FFE4B5		Shades	Mix
NavajoWhite	#FFDEAD		Shades	Mix
Navy	#000080		Shades	Mix
OldLace	#FDF5E6		Shades	Mix
Olive	#808000		Shades	Mix
OliveDrab	#6B8E23		Shades	Mix
Orange	#FFA500		Shades	Mix
OrangeRed	#FF4500		Shades	Mix
Orchid	#DA70D6		Shades	Mix
PaleGoldenRod	#EEE8AA		Shades	Mix
PaleGreen	#98FB98		Shades	Mix
PaleTurquoise	#AFEEEE		Shades	Mix
PaleVioletRed	#DB7093		Shades	Mix
PapayaWhip	#FFEFBD		Shades	Mix
PeachPuff	#FFDAB9		Shades	Mix
Peru	#CD853F		Shades	Mix

Pink	#FFCOCB		Shades	Mix
Plum	#DDA0DD		Shades	Mix
PowderBlue	#B0E0E6		Shades	Mix
Purple	#800080		Shades	Mix
RebeccaPurple	#663399		Shades	Mix
Red	#FF0000		Shades	Mix
RosyBrown	#BC8F8F		Shades	Mix
RoyalBlue	#4169E1		Shades	Mix
SaddleBrown	#8B4513		Shades	Mix
Salmon	#FA8072		Shades	Mix
SandyBrown	#F4A460		Shades	Mix
SeaGreen	#2E8B57		Shades	Mix
SeaShell	#FFF5EE		Shades	Mix
Sienna	#A0522D		Shades	Mix
Silver	#COCOCO		Shades	Mix
SkyBlue	#87CEEB		Shades	Mix
SlateBlue	#6A5ACD		Shades	Mix
SlateGray	#708090		Shades	Mix
Snow	#FFFFFA		Shades	Mix
SpringGreen	#00FF7F		Shades	Mix
SteelBlue	#4682B4		Shades	Mix
Tan	#D2B48C		Shades	Mix
Teal	#008080		Shades	Mix
Thistle	#D8BFD8		Shades	Mix
Tomato	#FF6347		Shades	Mix

<u>Turquoise</u>	<u>#40E0D0</u>		<u>Shades</u>	<u>Mix</u>
<u>Violet</u>	<u>#EE82EE</u>		<u>Shades</u>	<u>Mix</u>
<u>Wheat</u>	<u>#F5DEB3</u>		<u>Shades</u>	<u>Mix</u>
<u>White</u>	<u>#FFFFFF</u>		<u>Shades</u>	<u>Mix</u>
<u>WhiteSmoke</u>	<u>#F5F5F5</u>		<u>Shades</u>	<u>Mix</u>
<u>Yellow</u>	<u>#FFFF00</u>		<u>Shades</u>	<u>Mix</u>
<u>YellowGreen</u>	<u>#9ACD32</u>		<u>Shades</u>	<u>Mix</u>

 [Previous](#) [\[Next Reference\]\(#\)](#)

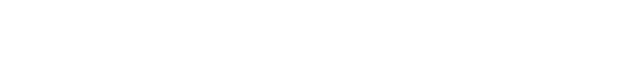
CSS Colors HEX Values

[Previous](#)
[Next Chapter](#)

Sorted by HEX Value

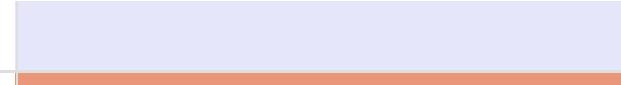
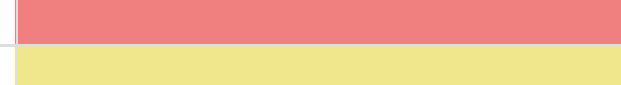
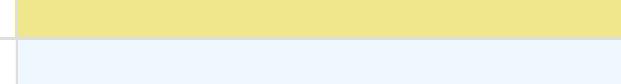
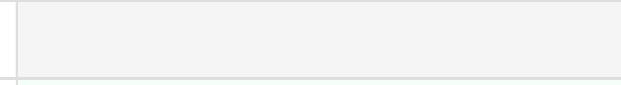
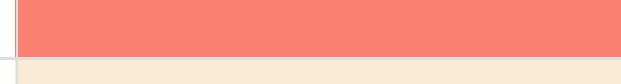
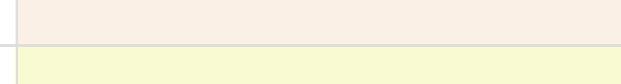
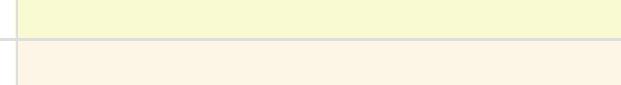
[Colors sorted by Color name](#)

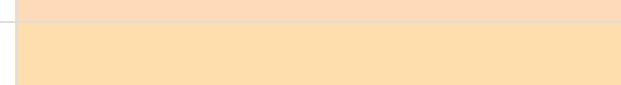
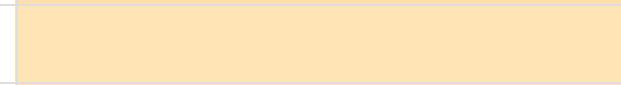
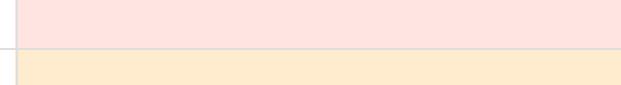
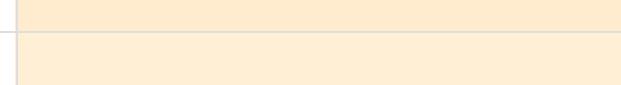
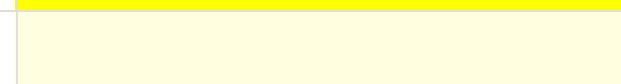
Color Name	HEX	Color	Shades	Mix
<u>Black</u>	<u>#000000</u>		<u>Shades</u>	<u>Mix</u>
<u>Navy</u>	<u>#000080</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkBlue</u>	<u>#00008B</u>		<u>Shades</u>	<u>Mix</u>
<u>MediumBlue</u>	<u>#0000CD</u>		<u>Shades</u>	<u>Mix</u>
<u>Blue</u>	<u>#0000FF</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkGreen</u>	<u>#006400</u>		<u>Shades</u>	<u>Mix</u>
<u>Green</u>	<u>#008000</u>		<u>Shades</u>	<u>Mix</u>
<u>Teal</u>	<u>#008080</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkCyan</u>	<u>#008B8B</u>		<u>Shades</u>	<u>Mix</u>
<u>DeepSkyBlue</u>	<u>#00BFFF</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkTurquoise</u>	<u>#00CED1</u>		<u>Shades</u>	<u>Mix</u>
<u>MediumSpringGreen</u>	<u>#00FA9A</u>		<u>Shades</u>	<u>Mix</u>
<u>Lime</u>	<u>#00FF00</u>		<u>Shades</u>	<u>Mix</u>
<u>SpringGreen</u>	<u>#00FF7F</u>		<u>Shades</u>	<u>Mix</u>
<u>Aqua</u>	<u>#00FFFF</u>		<u>Shades</u>	<u>Mix</u>

Cyan	#00FFFF		Shades	Mix
MidnightBlue	#191970		Shades	Mix
DodgerBlue	#1E90FF		Shades	Mix
LightSeaGreen	#20B2AA		Shades	Mix
ForestGreen	#228B22		Shades	Mix
SeaGreen	#2E8B57		Shades	Mix
DarkSlateGray	#2F4F4F		Shades	Mix
LimeGreen	#32CD32		Shades	Mix
MediumSeaGreen	#3CB371		Shades	Mix
Turquoise	#40E0DO		Shades	Mix
RoyalBlue	#4169E1		Shades	Mix
SteelBlue	#4682B4		Shades	Mix
DarkSlateBlue	#483D8B		Shades	Mix
MediumTurquoise	#48D1CC		Shades	Mix
Indigo	#4B0082		Shades	Mix
DarkOliveGreen	#556B2F		Shades	Mix
CadetBlue	#5F9EA0		Shades	Mix
CornflowerBlue	#6495ED		Shades	Mix
RebeccaPurple	#663399		Shades	Mix
MediumAquaMarine	#66CDAA		Shades	Mix
DimGray	#696969		Shades	Mix
SlateBlue	#6A5ACD		Shades	Mix
OliveDrab	#6B8E23		Shades	Mix
SlateGray	#708090		Shades	Mix
LightSlateGray	#778899		Shades	Mix
				

MediumSlateBlue	#7B68EE		Shades	Mix
LawnGreen	#7CFC00		Shades	Mix
Chartreuse	#7FFF00		Shades	Mix
Aquamarine	#7FFFD4		Shades	Mix
Maroon	#800000		Shades	Mix
Purple	#800080		Shades	Mix
Olive	#808000		Shades	Mix
Gray	#808080		Shades	Mix
SkyBlue	#87CEEB		Shades	Mix
LightSkyBlue	#87CEFA		Shades	Mix
BlueViolet	#8A2BE2		Shades	Mix
DarkRed	#8B0000		Shades	Mix
DarkMagenta	#8B008B		Shades	Mix
SaddleBrown	#8B4513		Shades	Mix
DarkSeaGreen	#8FBC8F		Shades	Mix
LightGreen	#90EE90		Shades	Mix
MediumPurple	#9370DB		Shades	Mix
DarkViolet	#9400D3		Shades	Mix
PaleGreen	#98FB98		Shades	Mix
DarkOrchid	#9932CC		Shades	Mix
YellowGreen	#9ACD32		Shades	Mix
Sienna	#A0522D		Shades	Mix
Brown	#A52A2A		Shades	Mix
DarkGray	#A9A9A9		Shades	Mix
LightBlue	#ADD8E6		Shades	Mix

GreenYellow	#ADFF2F		Shades	Mix
PaleTurquoise	#AFEEEE		Shades	Mix
LightSteelBlue	#B0C4DE		Shades	Mix
PowderBlue	#B0E0E6		Shades	Mix
FireBrick	#B22222		Shades	Mix
DarkGoldenRod	#B8860B		Shades	Mix
MediumOrchid	#BA55D3		Shades	Mix
RosyBrown	#BC8F8F		Shades	Mix
DarkKhaki	#BDB76B		Shades	Mix
Silver	#COCOCO		Shades	Mix
MediumVioletRed	#C71585		Shades	Mix
IndianRed	#CD5C5C		Shades	Mix
Peru	#CD853F		Shades	Mix
Chocolate	#D2691E		Shades	Mix
Tan	#D2B48C		Shades	Mix
LightGray	#D3D3D3		Shades	Mix
Thistle	#D8BFD8		Shades	Mix
Orchid	#DA70D6		Shades	Mix
GoldenRod	#DAA520		Shades	Mix
PaleVioletRed	#DB7093		Shades	Mix
Crimson	#DC143C		Shades	Mix
Gainsboro	#DCDCDC		Shades	Mix
Plum	#DDA0DD		Shades	Mix
BurlyWood	#DEB887		Shades	Mix
LightCyan	#E0FFFF		Shades	Mix

<u>Lavender</u>	<u>#E6E6FA</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkSalmon</u>	<u>#E9967A</u>		<u>Shades</u>	<u>Mix</u>
<u>Violet</u>	<u>#EE82EE</u>		<u>Shades</u>	<u>Mix</u>
<u>PaleGoldenRod</u>	<u>#EEE8AA</u>		<u>Shades</u>	<u>Mix</u>
<u>LightCoral</u>	<u>#F08080</u>		<u>Shades</u>	<u>Mix</u>
<u>Khaki</u>	<u>#FOE68C</u>		<u>Shades</u>	<u>Mix</u>
<u>AliceBlue</u>	<u>#FOF8FF</u>		<u>Shades</u>	<u>Mix</u>
<u>HoneyDew</u>	<u>#FOFFF0</u>		<u>Shades</u>	<u>Mix</u>
<u>Azure</u>	<u>#FOFFFF</u>		<u>Shades</u>	<u>Mix</u>
<u>SandyBrown</u>	<u>#F4A460</u>		<u>Shades</u>	<u>Mix</u>
<u>Wheat</u>	<u>#F5DEB3</u>		<u>Shades</u>	<u>Mix</u>
<u>Beige</u>	<u>#F5F5DC</u>		<u>Shades</u>	<u>Mix</u>
<u>WhiteSmoke</u>	<u>#F5F5F5</u>		<u>Shades</u>	<u>Mix</u>
<u>MintCream</u>	<u>#F5FFFA</u>		<u>Shades</u>	<u>Mix</u>
<u>GhostWhite</u>	<u>#F8F8FF</u>		<u>Shades</u>	<u>Mix</u>
<u>Salmon</u>	<u>#FA8072</u>		<u>Shades</u>	<u>Mix</u>
<u>AntiqueWhite</u>	<u>#FAEBD7</u>		<u>Shades</u>	<u>Mix</u>
<u>Linen</u>	<u>#FAFOE6</u>		<u>Shades</u>	<u>Mix</u>
<u>LightGoldenRodYellow</u>	<u>#FAFAD2</u>		<u>Shades</u>	<u>Mix</u>
<u>OldLace</u>	<u>#FDF5E6</u>		<u>Shades</u>	<u>Mix</u>
<u>Red</u>	<u>#FF0000</u>		<u>Shades</u>	<u>Mix</u>
<u>Fuchsia</u>	<u>#FF00FF</u>		<u>Shades</u>	<u>Mix</u>
<u>Magenta</u>	<u>#FF00FF</u>		<u>Shades</u>	<u>Mix</u>
<u>DeepPink</u>	<u>#FF1493</u>		<u>Shades</u>	<u>Mix</u>
<u>OrangeRed</u>	<u>#FF4500</u>		<u>Shades</u>	<u>Mix</u>

<u>Tomato</u>	<u>#FF6347</u>		<u>Shades</u>	<u>Mix</u>
<u>HotPink</u>	<u>#FF69B4</u>		<u>Shades</u>	<u>Mix</u>
<u>Coral</u>	<u>#FF7F50</u>		<u>Shades</u>	<u>Mix</u>
<u>DarkOrange</u>	<u>#FF8C00</u>		<u>Shades</u>	<u>Mix</u>
<u>LightSalmon</u>	<u>#FFA07A</u>		<u>Shades</u>	<u>Mix</u>
<u>Orange</u>	<u>#FFA500</u>		<u>Shades</u>	<u>Mix</u>
<u>LightPink</u>	<u>#FFB6C1</u>		<u>Shades</u>	<u>Mix</u>
<u>Pink</u>	<u>#FFC0CB</u>		<u>Shades</u>	<u>Mix</u>
<u>Gold</u>	<u>#FFD700</u>		<u>Shades</u>	<u>Mix</u>
<u>PeachPuff</u>	<u>#FFDAB9</u>		<u>Shades</u>	<u>Mix</u>
<u>NavajoWhite</u>	<u>#FFDEAD</u>		<u>Shades</u>	<u>Mix</u>
<u>Moccasin</u>	<u>#FFE4B5</u>		<u>Shades</u>	<u>Mix</u>
<u>Bisque</u>	<u>#FFE4C4</u>		<u>Shades</u>	<u>Mix</u>
<u>MistyRose</u>	<u>#FFE4E1</u>		<u>Shades</u>	<u>Mix</u>
<u>BlanchedAlmond</u>	<u>#FFEBBC</u>		<u>Shades</u>	<u>Mix</u>
<u>PapayaWhip</u>	<u>#FFEFDD</u>		<u>Shades</u>	<u>Mix</u>
<u>LavenderBlush</u>	<u>#FFF0F5</u>		<u>Shades</u>	<u>Mix</u>
<u>SeaShell</u>	<u>#FFF5EE</u>		<u>Shades</u>	<u>Mix</u>
<u>Cornsilk</u>	<u>#FFF8DC</u>		<u>Shades</u>	<u>Mix</u>
<u>LemonChiffon</u>	<u>#FFFACD</u>		<u>Shades</u>	<u>Mix</u>
<u>FloralWhite</u>	<u>#FFFAFO</u>		<u>Shades</u>	<u>Mix</u>
<u>Snow</u>	<u>#FFFafa</u>		<u>Shades</u>	<u>Mix</u>
<u>Yellow</u>	<u>#FFFF00</u>		<u>Shades</u>	<u>Mix</u>
<u>LightYellow</u>	<u>#FFFFE0</u>		<u>Shades</u>	<u>Mix</u>
<u>Ivory</u>	<u>#FFFFFF</u>		<u>Shades</u>	<u>Mix</u>

<u>White</u>	<u>#FFFFFF</u>		<u>Shades</u>	<u>Mix</u>
------------------------------	--------------------------------	--	-------------------------------	----------------------------

[◀ Previous](#)[Next Chapter ▶](#)

CSS3 Browser Support Reference

[Previous](#)
[Next](#)


W3Schools' Browser support reference is tested regularly with all major browsers.

CSS3 Browser support

The table below lists all CSS3 properties and their browser support:

Property	IE	Firefox	Chrome	Safari	Opera					
<u>align-content</u>		11		28		21		7		12.1
<u>align-items</u>		11		20		21		7		12.1
<u>align-self</u>		11		20		21		7		12.1
<u>@keyframes</u>		10		16		-webkit- 4		4		12.1
<u>animation</u>		10		16		-webkit- 4		4		12.1
<u>animation-name</u>		10		16		-webkit- 4		4		12.1
<u>animation-duration</u>		10		16		-webkit- 3		4		12.1
<u>animation-timing-function</u>		10		16		-webkit- 4		4		12.1
<u>animation-delay</u>		10		16		-webkit- 4		4		12.1
<u>animation-iteration-count</u>		10		16		-webkit- 4		4		12.1
<u>animation-direction</u>		10		16		-webkit- 4		4		12.1
<u>animation-play-state</u>		10		16		-webkit- 4		4		12.1
<u>backface-visibility</u>		-ms- 10		16		36		-webkit-		15
<u>background-clip</u>		9		4		4		3		10.5
<u>background-origin</u>		9		4		4		3		10.5

<u>background-size</u>		9		4		4		4.1		10
<u>border-bottom-left-radius</u>		9		4		5		5		10.5
<u>border-bottom-right-radius</u>		9		4		5		5		10.5
<u>border-image</u>		11		15		16		6		15
<u>border-image-outset</u>		11		15		15		6		15
<u>border-image-repeat</u>		11		15		15		6		15
<u>border-image-slice</u>		11		15		15		6		15
<u>border-image-source</u>		11		15		15		6		15
<u>border-image-width</u>		11		13		15		6		15
<u>border-radius</u>		9		4		5		5		10.5
<u>border-top-left-radius</u>		9		4		5		5		10.5
<u>border-top-right-radius</u>		9		4		5		5		10.5
box-decoration-break										
<u>box-shadow</u>		9		4		10		5.1		10.5
<u>box-sizing</u>		8		29		10		5.1		9.5
break-after		10								11.1
break-before		10								11.1
break-inside										
<u>column-count</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15
<u>column-fill</u>				-moz- 13						
<u>column-gap</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15
<u>column-rule</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15
<u>column-rule-color</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15
<u>column-rule-style</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15
<u>column-rule-width</u>		10		-moz- 2		-webkit- 4		-webkit-		-webkit- 15
<u>column-span</u>		10				-webkit- 4		-webkit-		-webkit- 15
<u>column-width</u>		10		-moz- 2		-webkit- 4		-webkit- 3		-webkit- 15

		10		-moz-	9		-webkit-	4		-webkit-	3		-webkit-	15
<u>columns</u>														
fit														
<u>flex</u>		11		28			29			6.1		12.1		
<u>flex-basis</u>		11		28			29			6.1		12.1		
<u>flex-direction</u>		11		28			29			6.1		12.1		
<u>flex-flow</u>		11		28			29			6.1		12.1		
<u>flex-grow</u>		11		28			29			6.1		12.1		
<u>flex-shrink</u>		11		28			29			6.1		12.1		
<u>flex-wrap</u>		11		28			29			6.1		12.1		
<u>@font-face</u>		9		3.6			4			3		10		
@font-feature-values														
font-feature-settings		10		34			16					25		
font-kerning														
font-language-override														
<u>font-size-adjust</u>							3							
<u>font-stretch</u>														
font-synthesis														
font-variant														
font-variant-alternates														
font-variant-caps														
font-variant-east-asian														
font-variant-ligatures														
font-variant-numeric														
font-variant-position														
<u>hanging-punctuation</u>														
hyphens		10		6			13			5.1				
icon														

image-orientation		26									
image-rendering		-moz-				11.6					
image-resolution											
<u>justify-content</u>		11		28		29		6.1		12.1	
line-break											
mark											
mark-after											
mark-before											
marks											
marquee-direction				-webkit-		-webkit-					
marquee-play-count				-webkit-		-webkit-					
marquee-speed				-webkit-		-webkit-					
marquee-style				-webkit-		-webkit-					
<u>nav-down</u>								11.5			
<u>nav-index</u>								11.5			
<u>nav-left</u>								11.5			
<u>nav-right</u>								11.5			
<u>nav-up</u>								11.5			
object-fit											
object-position											
<u>opacity</u>		9		2		4		3.1		9	
<u>order</u>		11		28		29		-webkit-6.1		12.1	
<u>outline-offset</u>				3.5		4		3		10.5	
overflow-wrap		9		3.5				3		9.5	
<u>overflow-x</u>		9		3.5		4		3		9.5	
<u>overflow-y</u>		9				4		3		9.5	
@page		8		19				5		6	

<u>perspective</u>		10		16		36		4		23
<u>perspective-origin</u>		10		16		36		4		23
<u>resize</u>				5		4		4		15
rest										
rest-after										
rest-before										
<u>ruby-align</u>										
<u>tab-size</u>				-moz- 4		21		6.1		15
<u>text-align-last</u>		5.5		-moz- 12		35				
text-combine-upright										
<u>text-decoration-color</u>				-moz- 6						
<u>text-decoration-line</u>				-moz- 6						
<u>text-decoration-style</u>				-moz- 6						
<u>text-justify</u>		5.5								
text-orientation										
<u>text-overflow</u>		6		7		4		3.1		11
<u>text-shadow</u>		10		3.5		4		4		9.6
text-underline-position										
<u>transform</u>		10		16		36		3.2		23
<u>transform-origin</u>		10		16		36		3.2		23
<u>transform-style</u>		11		16		36		4.0		23
<u>transition</u>		10		16		26		6.1		12.1
<u>transition-delay</u>		10		16		26		6.1		12.1
<u>transition-duration</u>		10		16		26		6.1		12.1
<u>transition-property</u>		10		16		26		6.1		12.1
<u>transition-timing-function</u>		10		16		26		6.1		12.1
<u>word-break</u>		5.5		15		4		3.1		15

word-wrap	 5.5	 3.5	 4	 3.1	 10.5
writing-mode					

Icon Explanations

 Supported by Internet Explorer

 Supported by Firefox

 Supported by Google Chrome

 Supported by Safari

 Supported by Opera

 Supported by Internet Explorer with the prefix -ms-

 Supported by Firefox with the prefix -moz-

 Supported by Google Chrome with the prefix -webkit-

 Supported by Safari with the prefix -webkit-

 Supported by Opera with the prefix -webkit-

For some properties you see the browser icon and a number:

 11

 26

 29

 5

 15

The number indicates in which browser version the property was first supported.

 Previous

Next 

CSS align-content Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Align the items of the flexible <div> element:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-flex-flow: row wrap; /* Safari 6.1+ */
    -webkit-align-content: space-around; /* Safari 7.0+ */
    display: flex;
    flex-flow: row wrap;
    align-content: space-around;
}
```

[Try it yourself »](#)

Definition and Usage

The align-content property aligns the flexible container's items when the items do not use all available space on the cross-axis (vertically).

Tip: Use the [justify-content](#) property to align the items on the main-axis (horizontally).

Note: There must be multiple lines of items for this property to have any effect.

Default value: stretch

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.alignContent="center"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- specify the first version that worked with a prefix.

Property					
align-content	21.0	11.0	28.0	7.0 -webkit-	12.1

CSS Syntax

```
align-content: stretch|center|flex-start|flex-end|space-between|space-around|initial|inherit;
```

Property Values

Value	Description	Play it
stretch	Default value. Items are stretched to fit the container	Play it »
center	Items are positioned at the center of the container	Play it »
flex-start	Items are positioned at the beginning of the container	Play it »
flex-end	Items are positioned at the end of the container	Play it »
space-between	Items are positioned with space between the lines	Play it »
space-around	Items are positioned with space before, between, and after the lines	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [align-items property](#)

CSS Reference: [align-self property](#)

CSS Reference: [justify-content property](#)

HTML DOM reference: [alignContent property](#)

 Previous

Complete  Reference

Next 

CSS align-items Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Center the alignments for all the items of the flexible <div> element:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-align-items: center; /* Safari 7.0+ */
    display: flex;
    align-items: center;
}
```

[Try it yourself »](#)

Definition and Usage

The align-items property specifies the default alignment for items inside the flexible container.

Tip: Use the align-self property of each item to override the align-items property.

Default value: stretch

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.alignItems="center"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- specify the first version that worked with a prefix.

Property					
align-items	21.0	11.0	20.0	7.0 -webkit-	12.1

CSS Syntax

```
align-items: stretch|center|flex-start|flex-end|baseline|initial|inherit;
```

Property Values

Value	Description	Play it
stretch	Default. Items are stretched to fit the container	Play it »
center	Items are positioned at the center of the container	Play it »
flex-start	Items are positioned at the beginning of the container	Play it »
flex-end	Items are positioned at the end of the container	Play it »
baseline	Items are positioned at the baseline of the container	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [align-content property](#)

CSS Reference: [align-self property](#)

HTML DOM reference: [alignItems property](#)

Previous

Complete CSS Reference

Next

CSS align-items property

CSS align-self Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Center the alignments for one of the items inside a flexible element:

```
#myBlueDiv {
    -webkit-align-self: center; /* Safari 7.0+ */
    align-self: center;
}
```

[Try it yourself »](#)

Definition and Usage

The align-self property specifies the alignment for the selected item inside the flexible container.

Note: The align-self property overrides the flexible container's align-items property.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.alignSelf="center"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- specify the first version that worked with a prefix.

Property					
align-self	21.0	11.0	20.0	7.0 -webkit-	12.1

CSS Syntax

```
align-self: auto|stretch|center|flex-start|flex-end|baseline|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Default. The element inherits its parent container's align-items property, or "stretch" if it has no parent container	Play it »
stretch	The element is positioned to fit the container	Play it »
center	The element is positioned at the center of the container	Play it »
flex-start	The element are positioned at the beginning of the container	Play it »
flex-end	The element is positioned at the end of the container	Play it »
baseline	The element is positioned at the baseline of the container	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [align-content property](#)

CSS Reference: [align-items property](#)

HTML DOM reference: [alignSelf property](#)

Previous

Complete CSS Reference

Next

CSS align-self property

CSS3 animation Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Binding an animation to a <div> element, using the shorthand property:

```
div {
    -webkit-animation: mymove 5s infinite; /* Chrome, Safari, Opera */
    animation: mymove 5s infinite;
}
```

[Try it yourself »](#)

Definition and Usage

The animation property is a shorthand property for six of the animation properties:

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

Note: Always specify the animation-duration property, otherwise the duration is 0, and will never be played.

Default value: none 0 ease 0 1 normal none running

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animation="mymove 5s infinite"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation: name duration timing-function delay iteration-count direction fill-mode play-state;
```

Property Values

Value	Description
<u>animation-name</u>	Specifies the name of the keyframe you want to bind to the selector
<u>animation-duration</u>	Specifies how many seconds or milliseconds an animation takes to complete
<u>animation-timing-function</u>	Specifies the speed curve of the animation
<u>animation-delay</u>	Specifies a delay before the animation will start
<u>animation-iteration-count</u>	Specifies how many times an animation should be played
<u>animation-direction</u>	Specifies whether or not the animation should play in reverse on alternate cycles
<u>animation-fill-mode</u>	Specifies what values are applied by the animation outside the time it is executing

<u>animation-play-state</u>	Specifies whether the animation is running or paused
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animation property](#)

 Previous

Complete  Reference

Next 

CSS3 animation-delay Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Wait two seconds, then start the animation:

```
div {
    -webkit-animation-delay: 2s; /* Chrome, Safari, Opera */
    animation-delay: 2s;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The animation-delay property specifies a delay for the start of an animation.

The animation-delay value is defined in seconds (s) or milliseconds (ms).

Default value: 0

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationDelay="1s"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-delay	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-delay: time|initial|inherit;
```

Property Values

Value	Description	Play it
<i>time</i>	Optional. Defines the number of seconds (s) or milliseconds (ms) to wait before the animation will start. Default value is 0. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

More Examples

Example

Using negative values, notice that the animation starts 2 seconds into the animation cycle:

```
div {
    -webkit-animation-delay: -2s; /* Chrome, Safari, Opera */
    animation-delay: -2s;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationDelay property](#)

 Previous

Complete  Reference

Next 

CSS3 animation-direction Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Do the animation once, then do the animation backwards:

```
div {
    -webkit-animation-direction: alternate; /* Chrome, Safari, Opera */
    animation-direction: alternate;
}
```

[Try it yourself »](#)

Definition and Usage

The animation-direction property defines whether an animation should play in reverse direction or in alternate cycles.

Default value: normal

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationDirection="reverse"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-direction	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-direction: normal|reverse|alternate|alternate-reverse|initial|inherit;
```

Property Values

Value	Description	Play it
normal	Default value. The animation should be played as normal	Play it »
reverse	The animation should play in reverse direction	Play it »
alternate	The animation will be played as normal every odd time (1,3,5,etc..) and in reverse direction every even time (2,4,6,etc...)	Play it »
alternate-reverse	The animation will be played in reverse direction every odd time (1,3,5,etc..) and in a normal direction every even time (2,4,6,etc...)	Play it »
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

Note: The "reverse" and "alternate-reverse" values are not supported in Safari.

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationDirection property](#)

◀ Previous

Complete Reference

Next ▶

CSS3 animation-fill-mode Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Animate something moving from one place to another and have it stay there:

```
div {
    -webkit-animation-fill-mode: forwards; /* Chrome, Safari, Opera */
    animation-fill-mode: forwards;
}
```

[Try it yourself »](#)

Definition and Usage

The animation-fill-mode property specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay).

By default, CSS animations do not affect the element until the first keyframe is "played", and then stops once the last keyframe has completed. The animation-fill-mode property can override this behavior.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationFillMode="forwards"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-fill-mode	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-fill-mode: none|forwards|backwards|both|initial|inherit;
```

Property Values

Value	Description
none	Default value. The animation will not apply any styles to the target element before or after it is executing
forwards	After the animation ends (determined by <code>animation-iteration-count</code>), the animation will apply the property values for the time the animation ended
backwards	The animation will apply the property values defined in the keyframe that will start the first iteration of the animation, during the period defined by <code>animation-delay</code> . These are either the values of the from keyframe (when <code>animation-direction</code> is "normal" or "alternate") or those of the to keyframe (when <code>animation-direction</code> is "reverse" or "alternate-reverse")
both	The animation will follow the rules for both forwards and backwards. That is, it will extend the animation properties in both directions
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationFillMode property](#)

 Previous

Complete  Reference

Next 

CSS3 animation-iteration-count Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Play the animation three times:

```
div {
    -webkit-animation-iteration-count: 3; /* Chrome, Safari, Opera */
    animation-iteration-count: 3;
}
```

[Try it yourself »](#)

Definition and Usage

The animation-iteration-count property specifies the number of times an animation should be played.

Default value: 1

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationIterationCount="infinite"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-iteration-count	4.0 - webkit-	10.0	16.0 5.0 -moz-	4.0 - webkit-	15.0 - webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-iteration-count: number|infinite|initial|inherit;
```

Property Values

Value	Description	Play it
<i>number</i>	A number that defines how many times an animation should be played. Default value is 1	Play it »
<i>infinite</i>	Specifies that the animation should be played infinite times (for ever)	Play it »
<i>initial</i>	Sets this property to its default value. Read about <i>initial</i>	
<i>inherit</i>	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationIterationCount property](#)

Previous

Complete Reference

Next

CSS3 animation-name Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify a name for the @keyframes animation:

```
div {
    -webkit-animation-name: mymove; /* Chrome, Safari, Opera */
    animation-name: mymove;
}
```

[Try it yourself »](#)

Definition and Usage

The animation-name property specifies a name for the @keyframes animation.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationName="myNEWmove"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.



Property					
animation-name	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-name: keyframename|none|initial|inherit;
```

Property Values

Value	Description
<i>keyframename</i>	Specifies the name of the keyframe you want to bind to the selector
none	Default value. Specifies that there will be no animation (can be used to override animations coming from the cascade)
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationName property](#)

Previous

Complete Reference

Next

CSS3 animation-play-state Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Pause an animation:

```
div {
    -webkit-animation-play-state: paused; /* Chrome, Safari, Opera */
    animation-play-state: paused;
}
```

[Try it yourself »](#)

Definition and Usage

The animation-play-state property specifies whether the animation is running or paused.

Note: Use this property in a JavaScript to pause an animation in the middle of a cycle.

Default value: running

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationPlayState="paused"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-play-state	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-play-state: paused|running|initial|inherit;
```

Property Values

Value	Description	Play it
paused	Specifies that the animation is paused	Play it »
running	Default value. Specifies that the animation is running	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationPlayState property](#)

Previous

Complete CSS Reference

Next

CSS3 animation-timing-function Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Play an animation with the same speed from beginning to end:

```
div {
    -webkit-animation-timing-function: linear; /* Chrome, Safari, Opera */
    animation-timing-function: linear;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The animation-timing-function specifies the speed curve of an animation.

The speed curve defines the TIME an animation uses to change from one set of CSS styles to another.

The speed curve is used to make the changes smoothly.

Default value: ease

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.animationTimingFunction="linear"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
animation-timing-function	4.0 - webkit-	10.0	16.0 5.0 -moz-	4.0 - webkit-	15.0 - webkit- 12.1 12.0 -o-

CSS Syntax

```
animation-timing-function: linear|ease|ease-in|ease-out|cubic-
bezier(n,n,n,n)|initial|inherit;
```

The animation-timing-function uses a mathematical function, called the Cubic Bezier curve, to make the speed curve. You can use your own values in this function, or use one of the pre-defined values:

Property Values

Value	Description	Play it
linear	The animation has the same speed from start to end	Play it »
ease	Default value. The animation has a slow start, then fast, before it ends slowly	Play it »
ease-in	The animation has a slow start	Play it »
ease-out	The animation has a slow end	Play it »
ease-in-out	The animation has both a slow start and a slow end	Play it »
cubic-bezier(n,n,n,n)	Define your own values in the cubic-bezier function Possible values are numeric values from 0 to 1	
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

Tip: Try the different values in the Try it Yourself section below.



More Examples

Example

To better understand the different timing function values:
Here are five different <div> elements with five different values:

```
/* For Chrome, Safari and Opera */
#div1 {-webkit-animation-timing-function: linear;}
#div2 {-webkit-animation-timing-function: ease;}
#div3 {-webkit-animation-timing-function: ease-in;}
#div4 {-webkit-animation-timing-function: ease-out;}
#div5 {-webkit-animation-timing-function: ease-in-out;}

/* Standard syntax */
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
```

[Try it yourself »](#)

Example

Same as the example above, but the speed curves are defined with the cubic-bezier function:

```
/* For Chrome, Safari and Opera */
#div1 {-webkit-animation-timing-function: cubic-bezier(0,0,1,1);}
#div2 {-webkit-animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-webkit-animation-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-webkit-animation-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-webkit-animation-timing-function: cubic-bezier(0.42,0,0.58,1);}

/* Standard syntax */
#div1 {animation-timing-function: cubic-bezier(0,0,1,1);}
#div2 {animation-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {animation-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {animation-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {animation-timing-function: cubic-bezier(0.42,0,0.58,1);}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

HTML DOM reference: [animationTimingFunction property](#)

 Previous

Complete  Reference

Next 

CSS3 backface-visibility Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Hide the backside of a rotated <div> element:

```
div {  
    -webkit-backface-visibility: hidden; /* Chrome, Safari, Opera */  
    backface-visibility: hidden;  
}
```

[Try it yourself »](#)

Definition and Usage

The backface-visibility property defines whether or not an element should be visible when not facing the screen.

This property is useful when an element is rotated, and you do not want to see its backside.

To better understand this property, [view a demo](#).

Default value: visible

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.backfaceVisibility = "hidden"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
backface-visibility	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-

CSS Syntax

```
backface-visibility: visible|hidden|initial|inherit;
```

Property Values

Value	Description	Play it
visible	Default value. The backside is visible	Play it »
hidden	The backside is not visible	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

HTML DOM reference: [backfaceVisibility property](#)

Previous

Complete Reference

Next

CSS background-attachment Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

How to specify a fixed background-image:

```
body {  
    background-image: url('w3css.gif');  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}
```

[Try it yourself »](#)

Definition and Usage

The background-attachment property sets whether a background image is fixed or scrolls with the rest of the page.

Default value: scroll

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.backgroundAttachment="fixed"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
background-attachment	1.0	4.0	1.0	1.0	3.5

Note: Internet Explorer 8 and earlier versions do not support multiple background images on one element.

CSS Syntax

```
background-attachment: scroll|fixed|local|initial|inherit;
```

Property Values

Value	Description
scroll	The background scrolls along with the element. This is default
fixed	The background is fixed with regard to the viewport
local	The background scrolls along with the element's contents
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS Background](#)

HTML DOM reference: [backgroundAttachment property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS3 background-clip Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the painting area of the background:

```
div {
    background-color: yellow;
    background-clip: content-box;
}
```

[Try it yourself »](#)

Definition and Usage

The background-clip property specifies the painting area of the background.

Default value: border-box

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.backgroundClip="content-box"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
----------	--------	-------------------	---------	--------	-------

background-clip	4.0	9.0	4.0	3.0	10.5
-----------------	-----	-----	-----	-----	------

CSS Syntax

```
background-clip: border-box|padding-box|content-box|initial|inherit;
```

Property Values

Value	Description	Play it
border-box	Default value. The background is clipped to the border box	Play it »
padding-box	The background is clipped to the padding box	Play it »
content-box	The background is clipped to the content box	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS3 tutorial: [CSS3 Backgrounds](#)

HTML DOM reference: [backgroundClip property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS background-color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the background-color of different elements:

```
body {
    background-color: yellow;
}

h1 {
    background-color: #00ff00;
}

p {
    background-color: rgb(255,0,255);
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The background-color property sets the background color of an element.

The background of an element is the total size of the element, including padding and border (but not the margin).

Tip: Use a background color and a text color that makes the text easy to read.

Default value: transparent

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version:

CSS1

JavaScript syntax: `object.style.backgroundColor="#00FF00"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
background-color	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
background-color: color|transparent|initial|inherit;
```

Property Values

Value	Description	Play it
<code>color</code>	Specifies the background color. Look at CSS Color Values for a complete list of possible color values.	Play it »
<code>transparent</code>	Specifies that the background color should be transparent. This is default	Play it »
<code>initial</code>	Sets this property to its default value. Read about initial	Play it »
<code>inherit</code>	Inherits this property from its parent element. Read about inherit	



More Examples

Example

Set the background-color of a part of the text:

```
span.highlight {
    background-color: yellow;
```

}

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Background](#)

HTML DOM reference: [backgroundColor property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS background-image Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set a background-image for the <body> element:

```
body {
    background-image: url("paper.gif");
    background-color: #cccccc;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The background-image property sets one or more background images for an element.

The background of an element is the total size of the element, including padding and border (but not the margin).

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

Tip: Always set a background-color to be used if the image is unavailable.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.backgroundImage="url(smiley.gif)"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
background-image	1.0	4.0	1.0	1.0	3.5

Note: IE8 and earlier do not support multiple background images on one element.

CSS Syntax

```
background-image: url | none | initial | inherit;
```

Property Values

Value	Description
url('URL')	The URL to the image. To specify more than one image, separate the URLs with a comma
none	No background image will be displayed. This is default
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Example

Set multiple background images for the <body> element:

```
body {
    background-image: url("img_tree.gif"), url("img_flwr.gif");
    background-color: #cccccc;
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Background](#)

CSS3 tutorial: [CSS3 Backgrounds](#)

HTML DOM reference: [backgroundImage property](#)

 Previous

Complete  Reference

Next 

CSS3 background-origin Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Position the background image relative to the content box:

```
div {  
    background-image: url('smiley.gif');  
    background-repeat: no-repeat;  
    background-position: left;  
    background-origin: content-box;  
}
```

[Try it yourself »](#)

Definition and Usage

The background-origin property specifies what the background-position property should be relative to.

Note: If the background-attachment property for the background image is "fixed", this property has no effect.

Default value: padding-box

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.backgroundOrigin="content-box"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
background-origin	4.0	9.0	4.0	3.0	10.5

CSS Syntax

```
background-origin: padding-box|border-box|content-box|initial|inherit;
```

Property Values

Value	Description	Play it
padding-box	Default value. The background image is positioned relative to the padding box	Play it »
border-box	The background image is positioned relative to the border box	Play it »
content-box	The background image is positioned relative to the content box	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS3 tutorial: [CSS3 Backgrounds](#)

HTML DOM reference: [backgroundOrigin property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS3 background-origin property

CSS background-position Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

How to position a background-image:

```
body {  
    background-image: url('smiley.gif');  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: center;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The background-position property sets the starting position of a background image.

Tip: By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

Default value: 0% 0%

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.backgroundPosition= "center"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
background-position	1.0	4.0	1.0	1.0	3.5

Note: IE8 and earlier do not support multiple background images on one element.

CSS Syntax

```
background-position: value;
```

Property Values

Value	Description	Play it
left top	If you only specify one keyword, the other value will be "center"	Play it »
left center		
left bottom		
right top		
right center		
right bottom		
center top		
center center		
center bottom		
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. . Default value is: 0% 0%	Play it »
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units . If you only specify one value, the other value will be 50%. You can mix % and positions	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

inherit



More Examples

Example

How to position a background-image using percent:

```
body {  
    background-image: url('smiley.gif');  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: 30% 20%;  
}
```

[Try it yourself »](#)

Example

How to position a background-image using pixels:

```
body {  
    background-image: url('smiley.gif');  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: 50px 100px;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Background](#)

CSS reference: [background-image property](#)

HTML DOM reference: [backgroundPosition property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS background-repeat Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Repeat a background-image only vertically:

```
body {
    background-image: url("paper.gif");
    background-repeat: repeat-y;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The background-repeat property sets if/how a background image will be repeated.

By default, a background-image is repeated both vertically and horizontally.

Tip: The background image is placed according to the background-position property. If no background-position is specified, the image is always placed at the element's top left corner.

Default value: repeat

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.backgroundRepeat="repeat-x"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
background-repeat	1.0	4.0	1.0	1.0	3.5

Note: IE8 and earlier do not support multiple background images on one element.

CSS Syntax

```
background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;
```

Property Values

Value	Description	Play it
repeat	The background image will be repeated both vertically and horizontally. This is default	Play it »
repeat-x	The background image will be repeated only horizontally	Play it »
repeat-y	The background image will be repeated only vertically	Play it »
no-repeat	The background-image will not be repeated	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

How to repeat a background image both vertically and horizontally (default for background images):

```
body {
    background-image: url("paper.gif");
}
```

[Try it yourself »](#)

Example

Repeat a background image only horizontally:

```
body {  
    background-image: url("paper.gif");  
    background-repeat: repeat-x;  
}
```

[Try it yourself »](#)

Example

Display a background image only one time, with no-repeat:

```
body {  
    background-image: url("paper.gif");  
    background-repeat: no-repeat;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Background](#)

CSS reference: [background-position property](#)

HTML DOM reference: [backgroundRepeat property](#)

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS3 background-size Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the size of a background image:

```
div {
    background: url(img_flwr.gif);
    background-size: 80px 60px;
    background-repeat: no-repeat;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The background-size property specifies the size of the background images.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.backgroundSize="60px 120px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
background-size	4.0 1.0 -webkit-	9.0	4.0 3.6 -moz-	4.1 3.0 -webkit-	10.5 10.0 -o-

CSS Syntax

```
background-size: auto|Length|cover|contain|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Default value. The background-image contains its width and height	Play it »
length	Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"	Play it »
percentage	Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"	Play it »
cover	Scale the background image to be as large as possible so that the background area is completely covered by the background image. Some parts of the background image may not be in view within the background positioning area	Play it »
contain	Scale the image to the largest size such that both its width and its height can fit inside the content area	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

More Examples



Example

Stretch the background image to completely cover the content area:

```
div {  
    background: url(img_flwr.gif);  
    background-size: 100% 100%;  
    background-repeat: no-repeat;  
}
```

[Try it yourself »](#)

Example

Stretch the background image so that exactly four copies fit horizontally:

```
div {  
    background: url(w3css.gif);  
    background-size: 25%;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Backgrounds](#)

HTML DOM reference: [backgroundSize property](#)

[« Previous](#)

[Complete CSS Reference](#)

[Next »](#)

CSS border Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the style of the four border:

```
p {  
    border: 5px solid red;  
}
```

[Try it yourself »](#)

Definition and Usage

The border shorthand property sets all the border properties in one declaration.

The properties that can be set, are (in order): border-width, border-style, and border-color.

It does not matter if one of the values above are missing, e.g. border:solid #ff0000; is allowed.

Default value: medium none *color*

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: *object.style.border="3px solid blue"*

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
border	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
border: border-width border-style border-color|initial|inherit;
```

Property Values

Value	Description
<u>border-width</u>	Specifies the width of the border. Default value is "medium"
<u>border-style</u>	Specifies the style of the border. Default value is "none"
<u>border-color</u>	Specifies the color of the border. Default value is the color of the element
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS Border](#)

CSS tutorial: [CSS Box Model](#)

HTML DOM reference: [border property](#)

Previous

Complete Reference

Next

CSS border-bottom Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the style of the bottom border:

```
p {  
    border-style: solid;  
    border-bottom: thick dotted #ff0000;  
}
```

[Try it yourself »](#)

Definition and Usage

The border-bottom shorthand property sets all the bottom border properties in one declaration.

The properties that can be set, are (in order): border-bottom-width, border-bottom-style, and border-bottom-color.

It does not matter if one of the values above are missing, e.g. border-bottom:solid #ff0000; is allowed.

Default value: medium none *color*

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: *object.style.borderBottom="3px solid blue"*

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
border-bottom	1.0	4.0	1.0	1.0	3.5

Note: See individual browser support for each value below.

CSS Syntax

```
border-bottom: border-width border-style border-color|initial|inherit;
```

Property Values

Value	Description
<u>border-bottom-width</u>	Specifies the width of the bottom border. Default value is "medium"
<u>border-bottom-style</u>	Specifies the style of the bottom border. Default value is "none"
<u>border-bottom-color</u>	Specifies the color of the bottom border. Default value is the color of the element
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS Border](#)

HTML DOM reference: [borderBottom property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

CSS border-bottom-color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the color of the bottom border:

```
p {
    border-style: solid;
    border-bottom-color: #ff0000;
}
```

[Try it yourself »](#)

Definition and Usage

The border-bottom-color property sets the color of an element's bottom border.

Note: Always declare the border-style property before the border-bottom-color property. An element must have borders before you can change the color.

Default value: The current color of the element

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.borderBottomColor= "blue"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
border-bottom-color	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
border-bottom-color: color|transparent|initial|inherit;
```

Property Values

Value	Description	Play it
color	Specifies the background color. Look at CSS Color Values for a complete list of possible color values. Default color is the color of the element	Play it »
transparent	Specifies that the border color should be transparent	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Border](#)

CSS reference: [border-bottom property](#)

HTML DOM reference: [borderBottomColor property](#)

Previous

Complete CSS Reference

Next



CSS3 border-bottom-right-radius Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Add a rounded border to the bottom-right corner of a <div> element:

```
div {  
    border: 2px solid;  
    border-bottom-right-radius: 2em;  
}
```

[Try it yourself »](#)

Definition and Usage

The border-bottom-right-radius property defines the shape of the border of the bottom-right corner.

Tip: This property allows you to add rounded borders to elements!

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.borderBottomRightRadius="25px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
border-bottom-right-radius	5.0 4.0 -webkit-	9.0	4.0 3.0 -moz-	5.0 3.1 -webkit-	10.5

CSS Syntax

```
border-bottom-right-radius: Length|% [Length|%]|initial|inherit;
```

Note: The two length or percentage values of the border-bottom-right-radius properties define the radii of a quarter ellipse that defines the shape of the corner of the outer border edge. The first value is the horizontal radius, the second the vertical radius. If the second value is omitted it is copied from the first. If either length is zero, the corner is square, not rounded. Percentages for the horizontal radius refer to the width of the border box, whereas percentages for the vertical radius refer to the height of the border box.

Property Values

Value	Description	Play it
<i>length</i>	Defines the shape of the bottom-right corner. Default value is 0	Play it »
%	Defines the shape of the bottom-right corner in %	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS3 tutorial: [CSS3 Borders](#)

HTML DOM reference: [borderBottomRightRadius property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

WEB HOSTING

[UK Reseller Hosting](#)

WEB BUILDING

[FREE Website BUILDER](#)

[Free HTML5 Templates](#)

W3SCHOOLS EXAMS

[HTML, CSS, JavaScript, PHP, jQuery, and XML Certifications](#)

SHARE THIS PAGE

COLOR PICKER



[REPORT ERROR](#)

[PRINT PAGE](#)

[FORUM](#)

[ABOUT](#)

Top 10 Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [SQL Tutorial](#)
- [PHP Tutorial](#)
- [jQuery Tutorial](#)
- [Bootstrap Tutorial](#)
- [Angular Tutorial](#)
- [ASP.NET Tutorial](#)
- [XML Tutorial](#)

Top 10 References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [Browser Statistics](#)
- [HTML DOM](#)
- [PHP Reference](#)
- [jQuery Reference](#)
- [HTML Colors](#)
- [HTML Character Sets](#)
- [XML DOM](#)

Top 10 Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [HTML DOM Examples](#)
- [PHP Examples](#)
- [jQuery Examples](#)
- [XML Examples](#)
- [XML DOM Examples](#)
- [ASP Examples](#)
- [SVG Examples](#)

Web Certificates

- [HTML Certificate](#)
- [HTML5 Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [jQuery Certificate](#)
- [PHP Certificate](#)
- [XML Certificate](#)

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 1999-2015 by Refsnes Data. All Rights Reserved.



CSS3 border-bottom-right-radius property

CSS border-top-style Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the style of the top border:

```
p {  
    border-style: solid;  
    border-top-style: dotted;  
}
```

[Try it yourself »](#)

Definition and Usage

The border-top-style property sets the style of an element's top border.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.borderTopStyle="dashed"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
----------	--------	-------------------	---------	--------	-------

border-top-style	1.0	5.5	1.0	1.0	9.2
------------------	-----	-----	-----	-----	-----

Note: The value "hidden" is not supported in IE7 and earlier. IE8 requires a !DOCTYPE. IE9 and later support "hidden".

CSS Syntax

```
border-top-
style:none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit;
```

Property Values

Value	Description	Play it
none	Specifies no border. This is default	Play it »
hidden	The same as "none", except in border conflict resolution for table elements	Play it »
dotted	Specifies a dotted border	Play it »
dashed	Specifies a dashed border	Play it »
solid	Specifies a solid border	Play it »
double	Specifies a double border	Play it »
groove	Specifies a 3D grooved border. The effect depends on the border-color value	Play it »
ridge	Specifies a 3D ridged border. The effect depends on the border-color value	Play it »
inset	Specifies a 3D inset border. The effect depends on the border-color value	Play it »
outset	Specifies a 3D outset border. The effect depends on the border-color value	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Border](#)

CSS reference: [border-top property](#)

HTML DOM reference: [borderTopStyle property](#)

 Previous

Complete  Reference

Next 

CSS border-top-width Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the width of the top border:

```
p {  
    border-style: solid;  
    border-top-width: 15px;  
}
```

[Try it yourself »](#)

Definition and Usage

The border-top-width property sets the width of an element's top border.

Note: Always declare the border-style property before the border-top-width property. An element must have borders before you can change the width.

Default value: medium

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.borderTopWidth="5px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
border-top-width	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
border-top-width: medium|thin|thick|Length|initial|inherit;
```

Property Values

Value	Description	Play it
medium	Specifies a medium top border. This is default	Play it »
thin	Specifies a thin top border	Play it »
thick	Specifies a thick top border	Play it »
length	Allows you to define the thickness of the top border	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Border](#)

CSS reference: [border-top property](#)

HTML DOM reference: [borderTopWidth property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS border-width Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the width of the four borders:

```
p {  
    border-style: solid;  
    border-width: 15px;  
}
```

[Try it yourself »](#)

Definition and Usage

The border-width property sets the width of an element's four borders. This property can have from one to four values.

Examples:

- **border-width:thin medium thick 10px;**
 - top border is thin
 - right border is medium
 - bottom border is thick
 - left border is 10px
- **border-width:thin medium thick;**
 - top border is thin
 - right and left borders are medium
 - bottom border is thick
- **border-width:thin medium;**
 - top and bottom borders are thin
 - right and left borders are medium

- **border-width:thin;**
 - all four borders are thin

Note: Always declare the border-style property before the border-width property. An element must have borders before you can set the width.

Default value: medium

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.borderWidth="1px 5px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
border-width	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
border-width: medium|thin|thick|Length|initial|inherit;
```

Property Values

Value	Description	Play it
medium	Specifies a medium border. This is default	Play it »
thin	Specifies a thin border	Play it »
thick	Specifies a thick border	Play it »
length	Allows you to define the thickness of the border	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

[inherit](#)

Related Pages

CSS tutorial: [CSS Border](#)

HTML DOM reference: [borderWidth property](#)

 [Previous](#)

[Complete CSS Reference](#)

[Next](#) 

CSS bottom Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the bottom edge of the image to 5px above the bottom edge of its containing element:

```
img {  
    position: absolute;  
    bottom: 5px;  
}
```

[Try it yourself »](#)

Definition and Usage

For absolutely positioned elements, the bottom property sets the bottom edge of an element to a unit above/below the bottom edge of its containing element.

For relatively positioned elements, the bottom property sets the bottom edge of an element to a unit above/below its normal position.

Note: If "position:static", the bottom property has no effect.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.bottom="10px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
bottom	1.0	5.0	1.0	1.0	6.0

CSS Syntax

```
bottom: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Lets the browser calculate the bottom edge position. This is default	Play it »
length	Sets the bottom edge position in px, cm, etc. Negative values are allowed	Play it »
%	Sets the bottom edge position in % of containing element. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

CSS reference: [top property](#)

CSS reference: [left property](#)

CSS reference: [right property](#)

HTML DOM reference: [bottom property](#)

Previous

Complete Reference

Next

CSS3 box-shadow Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Add a box-shadow to a <div> element:

```
div {
  box-shadow: 10px 10px 5px #888888;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The box-shadow property attaches one or more drop-shadows to the box.

Default value: none

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.boxShadow = "10px 20px 30px blue"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
box-shadow	10.0 4.0 -webkit-	9.0	4.0 3.5 -moz-	5.1 3.1 -webkit-	10.5

CSS Syntax

`box-shadow: none|h-shadow v-shadow blur spread color |inset|initial|inherit;`

Note: The box-shadow property attaches one or more drop-shadows to the box. The property is a comma-separated list of shadows, each specified by 2-4 length values, an optional color, and an optional inset keyword. Omitted lengths are 0.

Property Values

Value	Description	Play it
none	Default value. No shadow is displayed	Play it »
<i>h-shadow</i>	Required. The position of the horizontal shadow. Negative values are allowed	Play it »
<i>v-shadow</i>	Required. The position of the vertical shadow. Negative values are allowed	Play it »
<i>blur</i>	Optional. The blur distance	Play it »
<i>spread</i>	Optional. The size of shadow. Negative values are allowed	Play it »
<i>color</i>	Optional. The color of the shadow. The default value is black. Look at CSS Color Values for a complete list of possible color values.	Play it »
Note: In Safari (on PC) the color parameter is required. If you do not specify the color, the shadow is not displayed at all.		
<i>inset</i>	Optional. Changes the shadow from an outer shadow (outset) to an inner shadow	Play it »
<i>initial</i>	Sets this property to its default value. Read about initial	Play it »
<i>inherit</i>	Inherits this property from its parent element. Read about inherit	



More Examples

Images thrown on the table

This example demonstrates how to create "polaroid" pictures and rotate the pictures.

Related Pages

CSS3 tutorial: [CSS3 Borders](#)

HTML DOM reference: [boxShadow property](#)

Previous

Complete Reference

Next

CSS3 box-sizing Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify two bordered boxes side by side:

```
div {
    box-sizing: border-box;
    width: 50%;
    float: left;
}
```

[Try it yourself »](#)

Definition and Usage

The box-sizing property is used to tell the browser what the sizing properties (width and height) should include.

Should they include the border-box or just the content-box which is the default value of the width and height properties.

For example, if you want two bordered boxes side by side, it can be achieved through setting box-sizing to "border-box". This forces the browser to render the box with the specified width and height, and place the border and padding inside the box.

Default value: content-box

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.boxSizing="border-box"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
box-sizing	10.0 4.0 -webkit-	8.0	29.0 2.0 -moz-	5.1 3.2 -webkit-	9.5

CSS Syntax

```
box-sizing: content-box|border-box|initial|inherit;
```

Property Values

Value	Description
content-box	Default. The width and height properties (and min/max properties) includes only the content. Border, padding, or margin are not included
border-box	The width and height properties (and min/max properties) includes content, padding and border, but not the margin
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS3 tutorial: [CSS3 User Interface](#)

HTML DOM reference: [boxSizing property](#)

Previous

Complete Reference

Next

CSS caption-side Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the placement of a table caption:

```
caption {  
    caption-side: bottom;  
}
```

[Try it yourself »](#)

Definition and Usage

The caption-side property specifies the placement of a table caption.

Default value: top

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.captionSide= "bottom"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
caption-side	1.0	8.0	1.0	1.0	4.0

Note: IE8 supports the caption-side property only if a !DOCTYPE is specified.

CSS Syntax

```
caption-side: top|bottom|initial|inherit;
```

Property Values

Value	Description	Play it
top	Puts the caption above the table. This is default	Play it »
bottom	Puts the caption below the table	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Table](#)

HTML DOM reference: [captionSide property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS clear Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

No floating elements allowed on the left or the right side of a specified <p> element:

```
img {  
    float: left;  
}  
  
p.clear {  
    clear: both;  
}
```

[Try it yourself »](#)

Definition and Usage

The clear property specifies which side(s) of an element other floating elements are not allowed.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.clear="both"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
clear	1.0	5.0	1.0	1.0	6.0

CSS Syntax

```
clear: none|left|right|both|initial|inherit;
```

Property Values

Value	Description
none	Default. Allows floating elements on both sides
left	No floating elements allowed on the left side
right	No floating elements allowed on the right side
both	No floating elements allowed on either the left or the right side
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS Float](#)

HTML DOM reference: [clear property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS clip Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Clip an image:

```
img {
    position: absolute;
    clip: rect(0px,60px,200px,0px);
}
```

[Try it yourself »](#)

Definition and Usage

What happens if an image is larger than its containing element? - The clip property lets you specify a rectangle to clip an absolutely positioned element. The rectangle is specified as four coordinates, all from the top-left corner of the element to be clipped.

Note: The clip property does not work if "overflow:visible".

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.clip = "rect(0px,50px,50px,0px)"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
clip	1.0	8.0	1.0	1.0	7.0

CSS Syntax

```
clip: auto|shape|initial|inherit;
```

Property Values

Value	Description	Play it
auto	No clipping will be applied. This is default	Play it »
shape	Clips an element. The only valid value is: rect (<i>top, right, bottom, left</i>)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

HTML DOM reference: [clip property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the text-color for different elements:

```
body {
    color: red;
}

h1 {
    color: #00ff00;
}

p {
    color: rgb(0,0,255);
}
```

[Try it yourself »](#)

Definition and Usage

The color property specifies the color of text.

Tip: Use a background color and a text color that makes the text easy to read.

Default value: *not specified*

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.color="#0000FF"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
color	1.0	3.0	1.0	1.0	3.5

CSS Syntax

```
color: color|initial|inherit;
```

Property Values

Value	Description	Play it
color	Specifies the text color. Look at CSS Color Values for a complete list of possible color values.	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [color property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS3 column-count Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Divide the text in the <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-count property specifies the number of columns an element should be divided into.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columnCount=3`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-count	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-count: number|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>number</i>	The optimal number of columns into which the content of the element will be flowed	Play it »
auto	Default value. The number of columns will be determined by other properties, like e.g. "column-width"	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
  -moz-column-gap: 40px; /* Firefox */
  column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {  
    -webkit-column-count: 3; /* Chrome, Safari, Opera */  
    -moz-column-count: 3; /* Firefox */  
    column-count: 3;  
  
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */  
    -moz-column-gap: 40px; /* Firefox */  
    column-gap: 40px;  
  
    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */  
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */  
    column-rule: 4px outset #ff00ff;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnCount property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS3 column-fill Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify how to fill columns:

```
div {
    -moz-column-fill: auto; /* Firefox */
    column-fill: auto;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-fill property specifies how to fill columns, balanced or not.

Default value: balance

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.columnFill="auto"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.

Property					
column-fill	Not supported	Not supported	13.0 -moz-	Not supported	Not supported

CSS Syntax

```
column-fill: balance|auto|initial|inherit;
```

Property Values

Value	Description
balance	Default value. Columns are balanced. Browsers should minimize the variation in column length
auto	Columns are filled sequentially, and they will have different lengths
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
  -moz-column-gap: 40px; /* Firefox */
  column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;

  -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
  -moz-column-gap: 40px; /* Firefox */
  column-gap: 40px;

  -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
  -moz-column-rule: 4px outset #ff00ff; /* Firefox */
  column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnFill property](#)

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS3 column-gap Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify a 40 pixels gap between the columns:

```
div {
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-gap property specifies the gap between the columns.

Note: If there is a column-rule between columns, it will appear in the middle of the gap.

Default value: normal

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columnGap="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-gap	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-gap: Length|normal|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	A specified length that will set the gap between the columns	Play it »
normal	Default value. Specifies a normal gap between the columns. W3C suggests a value of 1em	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {  
    -webkit-column-count: 3; /* Chrome, Safari, Opera */  
    -moz-column-count: 3; /* Firefox */  
    column-count: 3;  
  
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */  
    -moz-column-gap: 40px; /* Firefox */  
    column-gap: 40px;  
  
    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */  
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */  
    column-rule: 4px outset #ff00ff;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnGap property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS3 column-rule Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the width, style and color of the rule between columns:

```
div {
    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-rule property is a shorthand property for setting all the column-rule-* properties.

The column-rule property sets the width, style, and color of the rule between columns.

Default value: medium none *color*

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: *object.style.columnRule= "3px outset blue"*

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-rule	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-rule: column-rule-width column-rule-style column-rule-color|initial|inherit;
```

Property Values

Value	Description
<u>column-rule-width</u>	Sets the width of the rule between columns. Default value is medium
<u>column-rule-style</u>	Sets the style of the rule between columns. Default value is none
<u>column-rule-color</u>	Sets the color of the rule between columns. Default value is the color of the element
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {  
    -webkit-column-count: 3; /* Chrome, Safari, Opera */  
    -moz-column-count: 3; /* Firefox */  
    column-count: 3;  
  
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */  
    -moz-column-gap: 40px; /* Firefox */  
    column-gap: 40px;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnRule property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS3 column-rule-color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the color of the rule between columns:

```
div {
    -webkit-column-rule-color: #ff0000; /* Chrome, Safari, Opera */
    -moz-column-rule-color: #ff0000; /* Firefox */
    column-rule-color: #ff0000;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-rule-color property specifies the color of the rule between columns.

Default value: The current color of the element

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columnRuleColor="#0000ff"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-rule-color	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-rule-color: color|initial|inherit;
```

Property Values

Value	Description	Play it
color	Specifies the color of the rule. Look at CSS Color Values for a complete list of possible color values	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnRuleColor property](#)

[Previous](#)[Complete CSS Reference](#)[Next](#)

CSS3 column-rule-style Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the style of the rule between columns:

```
div {
    -webkit-column-rule-style: dotted; /* Chrome, Safari, Opera */
    -moz-column-rule-style: dotted; /* Firefox */
    column-rule-style: dotted;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-rule-style property specifies the style of the rule between columns.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.columnRuleStyle="dotted"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-rule-style	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

column-rule-style:

none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit;

Property Values

Value	Description	Play it
none	Default value. Defines no rule	Play it »
hidden	Defines a hidden rule	Play it »
dotted	Defines a dotted rule	Play it »
dashed	Defines a dashed rule	Play it »
solid	Defines a solid rule	Play it »
double	Defines a double rule	Play it »
groove	Specifies a 3D grooved rule. The effect depends on the width and color values	Play it »
ridge	Specifies a 3D ridged rule. The effect depends on the width and color values	Play it »
inset	Specifies a 3D inset rule. The effect depends on the width and color values	Play it »
outset	Specifies a 3D outset rule. The effect depends on the width and color values	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
}
```

```
-moz-column-rule: 4px outset #ff00ff; /* Firefox */  
column-rule: 4px outset #ff00ff;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnRuleStyle property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS3 column-rule-width Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the width of the rule between columns:

```
div {
    -webkit-column-rule-width: 1px; /* Chrome, Safari, Opera */
    -moz-column-rule-width: 1px; /* Firefox */
    column-rule-width: 1px;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-rule-width property specifies the width of the rule between columns.

Default value: medium

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columnRuleWidth="10px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-rule-width	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-rule-width: medium|thin|thick|Length|initial|inherit;
```

Property Values

Value	Description	Play it
medium	Default value. Defines a medium rule	Play it »
thin	Defines a thin rule	Play it »
thick	Defines a thick rule	Play it »
length	Specifies the width of the rule	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnRuleWidth property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS3 column-span Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let the `<h2>` element span across all columns:

```
h2 {
    -webkit-column-span: all; /* Chrome, Safari, Opera */
    column-span: all;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-span property specifies how many columns an element should span across.

Default value: 1

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.columnSpan="all"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by `-webkit-` specify the first version that worked with a prefix.

Property					
column-span	4.0 -webkit-	10.0	Not supported	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-span: 1|all|initial|inherit;
```

Property Values

Value	Description	Play it
1	Default value. The element should span across one column	Play it »
all	The element should span across all columns	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the

columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnSpan property](#)

 [Previous](#)

[Complete CSS Reference](#)

[Next](#) 

CSS3 column-width Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the width of the columns:

```
div {
    -webkit-column-width: 100px; /* Chrome, Safari, Opera */
    -moz-column-width: 100px; /* Firefox */
    column-width: 100px;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The column-width property specifies the width of the columns.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columnWidth="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
column-width	4.0 -webkit-	10.0	2.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
column-width: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Deafult value. The column width will be determined by the browser	Play it »
length	A length that specifies the width of the columns	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columnWidth property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS3 columns Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the width and number of columns:

```
div {
    -webkit-columns: 100px 3; /* Chrome, Safari, Opera */
    -moz-columns: 100px 3; /* Firefox */
    columns: 100px 3;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The columns property is a shorthand property for setting column-width and column-count.

Default value: auto auto

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.columns="100px 3"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
columns	4.0 -webkit-	10.0	9.0 -moz-	3.1 -webkit-	15.0 -webkit- 11.1

CSS Syntax

```
columns: auto|column-width column-count|initial|inherit;
```

Property Values

Value	Description
auto	Default value. Sets both the column-width to "auto" and the column-count to "auto"
<u>column-width</u>	The width of the columns
<u>column-count</u>	The number of columns
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Column-count

Divide the text in a <div> element into three columns:

```
div {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
}
```

[Try it yourself »](#)

Column-gap

Divide the text in a <div> element into three columns, and specify a 40 pixels gap between the columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
}
```

[Try it yourself »](#)

Column-rule

Specify the width, style and color of the rule between columns.

```
div {
    -webkit-column-count: 3; /* Chrome, Safari, Opera */
    -moz-column-count: 3; /* Firefox */
    column-count: 3;

    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;

    -webkit-column-rule: 4px outset #ff00ff; /* Chrome, Safari, Opera */
    -moz-column-rule: 4px outset #ff00ff; /* Firefox */
    column-rule: 4px outset #ff00ff;
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Multiple Columns](#)

HTML DOM reference: [columns property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS3 columns property

CSS content Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

The following example inserts the URL in parenthesis after each link:

```
a::after {  
    content: " (" attr(href) ")";  
}
```

[Try it yourself »](#)

Definition and Usage

The content property is used with the :before and :after pseudo-elements, to insert generated content.

Default value: normal

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: You can't give an element a pseudo-class by using JavaScript, but there are other ways to get the same result:

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
content	1.0	8.0	1.0	1.0	4.0

Note: IE8 only supports the content property if a !DOCTYPE is specified.

CSS Syntax

```
content: normal|none|counter|attr|string|open-quote|close-quote|no-open-quote|no-close-quote|url|initial|inherit;
```

Property Values

Value	Description	Example
normal	Default value. Sets the content, if specified, to normal, which default is "none" (which is nothing)	Try it »
none	Sets the content, if specified, to nothing	Try it »
counter	Sets the content as a counter	Try it »
attr (<i>attribute</i>)	Sets the content as one of the selector's attribute	Try it »
<i>string</i>	Sets the content to the text you specify	Try it »
open-quote	Sets the content to be an opening quote	Try it »
close-quote	Sets the content to be a closing quote	Try it »
no-open-quote	Removes the opening quote from the content, if specified	Try it »
no-close-quote	Removes the closing quote from the content, if specified	Try it »
url(<i>url</i>)	Sets the content to be some kind of media (an image, a sound, a video, etc.)	Try it »
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS reference: [:before pseudo element](#)

CSS reference: [:after pseudo element](#)

 Previous

Complete  Reference

Next 

CSS counter-increment Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

A way to number sections and sub-sections with "Section 1", "1.1", "1.2", etc.:

```
body {  
    counter-reset: section;  
}  
  
h1 {  
    counter-reset: subsection;  
}  
  
h1:before {  
    counter-increment: section;  
    content: "Section " counter(section) ". ";  
}  
  
h2:before {  
    counter-increment: subsection;  
    content: counter(section) "." counter(subsection) " ";  
}
```

[Try it yourself »](#)

Definition and Usage

The counter-increment property increments one or more counter values.

The counter-increment property is usually used together with the counter-reset property and the content property.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)**Version:** CSS2**JavaScript syntax:** `object.style.counterIncrement = "subsection";`[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
counter-increment	4.0	8.0	2.0	3.1	9.6

Note: IE8 supports the counter-increment property only if a !DOCTYPE is specified.

CSS Syntax

```
counter-increment: none|id|initial|inherit;
```

Property Values

Value	Description
none	Default value. No counters will be incremented
<i>id number</i>	The <i>id</i> defines which counter that should be incremented. The <i>number</i> sets how much the counter will increment on each occurrence of the selector. The default increment is 1. 0 or negative values, are allowed. If the <i>id</i> refers to a counter that has not been initialized by counter-reset, the default initial value is 0
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS reference: [:before pseudo element](#)

CSS reference: [:after pseudo element](#)

CSS reference: [content property](#)

CSS reference: [counter-reset property](#)

HTML DOM reference: [counterIncrement property](#)

 Previous

Complete  Reference

Next 

CSS counter-reset Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

A way to number sections and sub-sections with "Section 1", "1.1", "1.2", etc.:

```
body {  
    counter-reset: section;  
}  
  
h1 {  
    counter-reset: subsection;  
}  
  
h1:before {  
    counter-increment: section;  
    content: "Section " counter(section) ". ";  
}  
  
h2:before {  
    counter-increment: subsection;  
    content: counter(section) "." counter(subsection) " ";  
}
```

[Try it yourself »](#)

Definition and Usage

The counter-reset property creates or resets one or more counters.

The counter-reset property is usually used together with the counter-increment property and the content property.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)**Version:** CSS2**JavaScript syntax:** `object.style.counterReset="section"`[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
counter-reset	4.0	8.0	2.0	3.1	9.6

Note: IE8 supports the counter-reset property only if a !DOCTYPE is specified.

CSS Syntax

```
counter-reset: none|name number|initial|inherit;
```

Property Values

Value	Description
none	Default value. No counters will be reset
<i>name</i>	The <i>name</i> defines which counter that should be reset
<i>number</i>	The <i>id</i> defines which counter that should be reset. The <i>number</i> sets the value the counter is set to on each occurrence of the selector. The default reset value is 0
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS reference: [:before pseudo element](#)

CSS reference: [:after pseudo element](#)

CSS reference: [content property](#)

CSS reference: [counter-increment property](#)

HTML DOM reference: [counterReset property](#)

 Previous

Complete  Reference

Next 

CSS cursor Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Some different cursors:

```
span.crosshair {  
    cursor: crosshair;  
}  
  
span.help {  
    cursor: help;  
}  
  
span.wait {  
    cursor: wait;  
}
```

[Try it yourself »](#)

Definition and Usage

The cursor property specifies the type of cursor to be displayed when pointing on an element.

Default value: auto

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.cursor="crosshair"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
cursor	5.0	5.5	4.0	5.0	9.6

CSS Syntax

```
cursor: value;
```

Property Values

Value	Description	Play it
alias	The cursor indicates an alias of something is to be created	Play it »
all-scroll	The cursor indicates that something can be scrolled in any direction	Play it »
auto	Default. The browser sets a cursor	Play it »
cell	The cursor indicates that a cell (or set of cells) may be selected	Play it »
context-menu	The cursor indicates that a context-menu is available	Play it »
col-resize	The cursor indicates that the column can be resized horizontally	Play it »
copy	The cursor indicates something is to be copied	Play it »
crosshair	The cursor render as a crosshair	Play it »
default	The default cursor	Play it »
e-resize	The cursor indicates that an edge of a box is to be moved right (east)	Play it »
ew-resize	Indicates a bidirectional resize cursor	Play it »
grab	The cursor indicates that something can be grabbed	Play it »
grabbing	The cursor indicates that something can be grabbed	Play it »

help	The cursor indicates that help is available	Play it »
move	The cursor indicates something is to be moved	Play it »
n-resize	The cursor indicates that an edge of a box is to be moved up (north)	Play it »
ne-resize	The cursor indicates that an edge of a box is to be moved up and right (north/east)	Play it »
nesw-resize	Indicates a bidirectional resize cursor	Play it »
ns-resize	Indicates a bidirectional resize cursor	Play it »
nw-resize	The cursor indicates that an edge of a box is to be moved up and left (north/west)	Play it »
nwse-resize	Indicates a bidirectional resize cursor	Play it »
no-drop	The cursor indicates that the dragged item cannot be dropped here	Play it »
none	No cursor is rendered for the element	Play it »
not-allowed	The cursor indicates that the requested action will not be executed	Play it »
pointer	The cursor is a pointer and indicates a link	Play it »
progress	The cursor indicates that the program is busy (in progress)	Play it »
row-resize	The cursor indicates that the row can be resized vertically	Play it »
s-resize	The cursor indicates that an edge of a box is to be moved down (south)	Play it »
se-resize	The cursor indicates that an edge of a box is to be moved down and right (south/east)	Play it »
sw-resize	The cursor indicates that an edge of a box is to be moved down and left (south/west)	Play it »
text	The cursor indicates text that may be selected	Play it »
URL	A comma separated list of URLs to custom cursors. Note: Always specify a generic cursor at the end of the list, in case	Play it »

none of the URL-defined cursors can be used

vertical-text	The cursor indicates vertical-text that may be selected	Play it »
w-resize	The cursor indicates that an edge of a box is to be moved left (west)	Play it »
wait	The cursor indicates that the program is busy	Play it »
zoom-in	The cursor indicates that something can be zoomed in	Play it »
zoom-out	The cursor indicates that something can be zoomed out	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

HTML DOM reference: [cursor property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS direction Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the text direction to "right-to-left":

```
div {
    direction: rtl;
}
```

[Try it yourself »](#)

Definition and Usage

The direction property specifies the text direction/writing direction.

Tip: Use this property together with the unicode-bidi property to set or return whether the text should be overridden to support multiple languages in the same document.

Default value: ltr

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.direction="rtl"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
direction	2.0	5.5	1.0	1.3	9.2

CSS Syntax

```
direction: ltr|rtl|initial|inherit;
```

Property Values

Value	Description	Play it
ltr	The writing direction is left-to-right. This is default	Play it »
rtl	The writing direction is right-to-left	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [direction property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS display Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Display <p> elements as inline elements:

```
p.inline {
    display: inline;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The display property specifies the type of box used for an HTML element.

Default value: inline

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.display="none"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



display	4.0	8.0 Partial from 5.5	3.0	3.1	7.0
---------	-----	----------------------------	-----	-----	-----

Note: The values "inline-table", "table", "table-caption", "table-cell", "table-column", "table-column-group", "table-row", and "table-row-group" are not supported in IE7 and earlier. IE8 requires a !DOCTYPE. IE9 supports the values.

Note: The values "flex" and "inline-flex" requires a prefix to work in Safari. For "flex" use "display: -webkit-flex", for "inline-flex" use "display: -webkit-inline-flex;".

CSS Syntax

```
display: value;
```

Property Values

Value	Description	Play it
inline	Default value. Displays an element as an inline element (like)	Play it »
block	Displays an element as a block element (like <p>)	Play it »
flex	Displays an element as a block-level flex container. New in CSS3	
inline-block	Displays an element as an inline-level block container. The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box	
inline-flex	Displays an element as an inline-level flex container. New in CSS3	
inline-table	The element is displayed as an inline-level table	
list-item	Let the element behave like a element	Play it »
run-in	Displays an element as either block or inline, depending on context	
table	Let the element behave like a <table> element	
table-caption	Let the element behave like a <caption> element	

table-column-group	Let the element behave like a <colgroup> element
table-header-group	Let the element behave like a <thead> element
table-footer-group	Let the element behave like a <tfoot> element
table-row-group	Let the element behave like a <tbody> element
table-cell	Let the element behave like a <td> element
table-column	Let the element behave like a <col> element
table-row	Let the element behave like a <tr> element
none	The element will not be displayed at all (has no effect on layout)
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>



More Examples

Example

A demonstration of how to use the inherit property value:

```
body {
    display: inline
}

p {
    display: inherit
}
```

[Try it yourself »](#)

Example

Set the direction of some flexible items inside a <div> element in reverse order:

```
div {  
  display: -webkit-flex; /* Safari */  
  -webkit-flex-direction: row-reverse; /* Safari 6.1+ */  
  display: flex;  
  flex-direction: row-reverse;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Display and visibility](#)

HTML DOM reference: [display property](#)

[◀ Previous](#)

Complete **CSS** Reference

[Next ▶](#)

CSS empty-cells Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Hide border and background on empty cells in a table:

```
table {
    border-collapse: separate;
    empty-cells: hide;
}
```

[Try it yourself »](#)

Definition and Usage

The empty-cells property sets whether or not to display borders and background on empty cells in a table (only for the "separated borders" model).

Default value: show

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.emptyCells="hide"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera	Microsoft Edge
empty-cells	1	1	1	1	1	1



	empty-cells	1.0		8.0		1.0		1.2		4.0
--	-------------	-----	--	-----	--	-----	--	-----	--	-----

Note: IE8 supports the empty-cells property only if a !DOCTYPE is specified.

CSS Syntax

```
empty-cells: show|hide|initial|inherit;
```

Property Values

Value	Description	Play it
show	Background and borders are shown on empty cells. This is default	Play it »
hide	No background or borders are shown on empty cells	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Table](#)

HTML DOM reference: [emptyCells property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS flex Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let all the flexible items be the same length, regardless of its content:

```
#main div {
    -webkit-flex: 1; /* Safari 6.1+ */
    -ms-flex: 1; /* IE 10 */
    flex: 1;
}
```

[Try it yourself »](#)

Definition and Usage

The flex property specifies the length of the item, relative to the rest of the flexible items inside the same container.

The flex property is a shorthand for the flex-grow, flex-shrink, and the flex-basis properties.

Note: If the element is not a flexible item, the flex property has no effect.

Default value: 0 1 auto

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.flex="1"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -ms- or -moz- specify the first version that worked with a prefix.

Property					
flex	29.0 21.0 -webkit-	11.0 10.0 -ms-	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex: flex-grow flex-shrink flex-basis|auto|initial|inherit;
```

Property Values

Value	Description
flex-grow	A number specifying how much the item will grow relative to the rest of the flexible items
flex-shrink	A number specifying how much the item will shrink relative to the rest of the flexible items
flex-basis	The length of the item. Legal values: "auto", "inherit", or a number followed by "%", "px", "em" or any other length unit
auto	Same as 1 1 auto.
initial	Same as 0 1 auto. Read about initial
none	Same as 0 0 auto.
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

CSS Reference: [flex-wrap property](#)

HTML DOM reference: [flex property](#)

 Previous

Complete  CSS Reference

Next 

CSS flex-basis Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the initial length of the second flex-item to 80 pixels:

```
div:nth-of-type(2) {
    -webkit-flex-basis: 80px; /* Safari 6.1+ */
    flex-basis: 80px;
}
```

[Try it yourself »](#)

Definition and Usage

The flex-basis property specifies the initial length of a flexible item.

Note: If the element is not a flexible item, the flex-basis property has no effect.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.flexBasis="200px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-basis	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-basis: number|auto|initial|inherit;
```

Property Values

Value	Description	Play it
number	A length unit, or percentage, specifying the initial length of the flexible item(s)	Play it »
auto	Default value. The length is equal to the length of the flexible item. If the item has no length specified, the length will be according to its content	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

CSS Reference: [flex-wrap property](#)

HTML DOM reference: [flexBasis property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS flex-direction Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the direction of the flexible items inside the <div> element in reverse order:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-flex-direction: row-reverse; /* Safari 6.1+ */
    display: flex;
    flex-direction: row-reverse;
}
```

[Try it yourself »](#)

Definition and Usage

The flex-direction property specifies the direction of the flexible items.

Note: If the element is not a flexible item, the flex-direction property has no effect.

Default value: row

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.flexDirection="column-reverse"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-direction	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-direction: row|row-reverse|column|column-reverse|initial|inherit;
```

Property Values

Value	Description	Play it
row	Default value. The flexible items are displayed horizontally, as a row	Play it »
row-reverse	Same as row, but in reverse order	Play it »
column	The flexible items are displayed vertically, as a column	Play it »
column-reverse	Same as column, but in reverse order	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-wrap property](#)

CSS Reference: [flex property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

CSS Reference: [flex-basis property](#)

HTML DOM reference: [flexDirection property](#)

[Previous](#)

Complete **CSS** Reference

[Next](#)

CSS flex-flow Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Make the flexible items display in reverse order, and wrap if necessary:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-flex-flow: row-reverse wrap; /* Safari 6.1+ */
    display: flex;
    flex-flow: row-reverse wrap;
}
```

[Try it yourself »](#)

Definition and Usage

The flex-flow property is a shorthand property for the flex-direction and the flex-wrap properties.

The flex-direction property specifies the direction of the flexible items.

The flex-wrap property specifies whether the flexible items should wrap or not.

Note: If the elements are not flexible items, the flex-flow property has no effect.

Default value: row nowrap

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.flexFlow="column nowrap"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-flow	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-flow: flex-direction flex-wrap | initial | inherit;
```

Property Values

Value	Description	Play it
<i>flex-direction</i>	Possible values: row row-reverse column column-reverse initial inherit Default value is "row". Specifying the direction of the flexible items	Play it »
<i>flex-wrap</i>	Possible values: nowrap wrap wrap-reverse initial inherit Default value is "nowrap". Specifying whether the flexible items should wrap or not	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »

inherit

Inherits this property from its parent element. [Read about *inherit*](#)

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-wrap property](#)

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

HTML DOM reference: [flexFlow property](#)

◀ Previous

Complete CSS Reference

Next ▶

CSS flex-grow Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let the second flex-item grow three times wider than the rest:

```
/* Safari 6.1+ */
div:nth-of-type(1) {-webkit-flex-grow: 1;}
div:nth-of-type(2) {-webkit-flex-grow: 3;}
div:nth-of-type(3) {-webkit-flex-grow: 1;}

/* Standard syntax */
div:nth-of-type(1) {flex-grow: 1;}
div:nth-of-type(2) {flex-grow: 3;}
div:nth-of-type(3) {flex-grow: 1;}
```

[Try it yourself »](#)

Definition and Usage

The flex-grow property specifies how much the item will grow relative to the rest of the flexible items inside the same container.

Note: If the element is not a flexible item, the flex-grow property has no effect.

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.flexGrow="5"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-grow	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-grow: number|initial|inherit;
```

Property Values

Value	Description	Play it
number	A number specifying how much the item will grow relative to the rest of the flexible items. Default value is 0	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-shrink property](#)

CSS Reference: [flex-wrap property](#)

HTML DOM reference: [flexGrow property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS flex-shrink Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let the second flex-item shrink three times more than the rest:

```
/* Safari 6.1+ */
div:nth-of-type(2) {
    -webkit-flex-shrink: 3;
}

/* Standard syntax */
div:nth-of-type(2) {
    flex-shrink: 3;
}
```

[Try it yourself »](#)

Definition and Usage

The flex-shrink property specifies how the item will shrink relative to the rest of the flexible items inside the same container.

Note: If the element is not a flexible item, the flex-shrink property has no effect.

Default value: 1

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.flexShrink="5"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-shrink	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-shrink: number|initial|inherit;
```

Property Values

Value	Description	Play it
<i>number</i>	A number specifying how much the item will shrink relative to the rest of the flexible items. Default value is 1	Play it »
<i>initial</i>	Sets this property to its default value. Read about initial	Play it »
<i>inherit</i>	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-wrap property](#)

HTML DOM reference: [flexShrink property](#)

◀ Previous

Complete CSS Reference

Next ▶

CSS flex-wrap Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Make the flexible items wrap if necessary:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-flex-wrap: wrap; /* Safari 6.1+ */
    display: flex;
    flex-wrap: wrap;
}
```

[Try it yourself »](#)

Definition and Usage

The flex-wrap property specifies whether the flexible items should wrap or not.

Note: If the elements are not flexible items, the flex-wrap property has no effect.

Default value: nowrap

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.flexWrap="nowrap"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
flex-wrap	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
flex-wrap: nowrap|wrap|wrap-reverse|initial|inherit;
```

Property Values

Value	Description	Play it
nowrap	Default value. Specifies that the flexible items will not wrap	Play it »
wrap	Specifies that the flexible items will wrap if necessary	Play it »
wrap-reverse	Specifies that the flexible items will wrap, if necessary, in reverse order	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

HTML DOM reference: [flexWrap property](#)

[Previous](#)

Complete **CSS** Reference

[Next](#)

CSS float Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let an image float to the right:

```
img {
  float: right;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The float property specifies whether or not a box (an element) should float.

Note: Absolutely positioned elements ignores the float property!

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.cssFloat="left"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
float	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
float: none|left|right|initial|inherit;
```

Property Values

Value	Description	Play it
none	The element is not floated, and will be displayed just where it occurs in the text. This is default	Play it »
left	The element floats to the left	Play it »
right	The element floats the right	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Let the first letter of a paragraph float to the left

Let the first letter of a paragraph float to the left and style the letter.

Creating a horizontal menu

Use float with a list of hyperlinks to create a horizontal menu.

Creating a homepage without tables

Use float to create a homepage with a header, footer, left content and main content.

Related Pages

CSS tutorial: [CSS Float](#)

HTML DOM reference: [cssFloat property](#)

[Previous](#)

Complete [CSS](#) Reference

[Next](#)

CSS font Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify all the font properties in one declaration:

```
p.ex1 {
    font: 15px arial, sans-serif;
}

p.ex2 {
    font: italic bold 12px/30px Georgia, serif;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The font shorthand property sets all the font properties in one declaration.

The properties that can be set, are (in order): "font-style font-variant font-weight font-size/line-height font-family"

The font-size and font-family values are required. If one of the other values are missing, the default values will be inserted, if any.

Note: The line-height property sets the space between lines.

Default value: *The default value of all the font properties*

Inherited: yes

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version:

CSS1

JavaScript syntax: *object.style.font="italic small-caps bold 12px arial,sans-serif"***Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font	1.0	4.0	1.0	1.0	3.5

Note: See individual browser support for each value below.

CSS Syntax

```
font: font-style font-variant font-weight font-size/line-height font-family|caption|icon|menu|message-box|small-caption|status-bar|initial|inherit;
```

Property Values

Property/Value	Description
<i>font-style</i>	Specifies the font style. Default value is "normal". See font-style for possible values
<i>font-variant</i>	Specifies the font variant. Default value is "normal". See font-variant for possible values
<i>font-weight</i>	Specifies the font weight. Default value is "normal". See font-weight for possible values
<i>font-size/line-height</i>	Specifies the font size and the line-height. Default value is "normal". See font-size and line-height for possible values
<i>font-family</i>	Specifies the font family. Default value depends on the browser. See font-family for possible values
<i>caption</i>	Uses the font that are used by captioned controls (like buttons, drop-downs, etc.)
<i>icon</i>	Uses the font that are used by icon labels

menu	Uses the fonts that are used by dropdown menus
message-box	Uses the fonts that are used by dialog boxes
small-caption	A smaller version of the caption font
status-bar	Uses the fonts that are used by the status bar
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Example

A demonstration of some of the other font property values.

```
<p style="font:caption">The browser font used in captioned controls.</p>
<p style="font:icon">The browser font used in icon labels.</p>
<p style="font:menu">The browser font used in dropdown menus.</p>
<p style="font:message-box">The browser font used in dialog boxes.</p>
<p style="font:small-caption">A smaller version of the caption font.</p>
<p style="font:status-bar">The browser font used in the status bar.</p>
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Font](#)

HTML DOM reference: [font property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS3 @font-face Rule

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify a font named "myFirstFont", and specify the URL where it can be found:

```
@font-face {  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

With the @font-face rule, web designers do no longer have to use one of the "web-safe" fonts.

In the new @font-face rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.

Tip: Use lowercase letters for the font URL. Uppercase letters can give unexpected results in IE!

To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-family property:

```
div {  
    font-family: myFirstFont;  
}
```

Browser Support

The @font-face rule is supported in Internet Explorer, Firefox, Opera, Chrome, and Safari.

The numbers in the table specifies the first browser version that fully supports the font format.

Font format					
TTF/OTF	9.0*	4.0	3.5	3.1	10.0
WOFF	9.0	5.0	3.6	5.1	11.1
WOFF2	Not supported	36.0	35.0*	Not supported	26.0
SVG	Not supported	4.0	Not supported	3.2	9.0
EOT	6.0	Not supported	Not supported	Not supported	Not supported

*IE: The font format only works when set to be "installable".

*Firefox: Disabled by default, but can be enabled (need to set a flag to "true" to use WOFF2).

Syntax

```
@font-face {
    font-properties
}
```

Font descriptor	Values	Description
font-family	<i>name</i>	Required. Defines the name of the font.
src	<i>URL</i>	Required. Defines the URL(s) where the font should be downloaded from
font-stretch	normal condensed ultra-condensed extra-condensed semi-condensed expanded semi-expanded extra-expanded ultra-expanded	Optional. Defines how the font should be stretched. Default value is "normal"
font-style	normal	Optional. Defines how the font should be styled.

	italic oblique	Default value is "normal"
font-weight	normal bold 100 200 300 400 500 600 700 800 900	Optional. Defines the boldness of the font. Default value is "normal"
unicode-range	<i>unicode-range</i>	Optional. Defines the range of unicode characters the font supports. Default value is "U+0-10FFFF"



Try it Yourself - Examples

Example

You must add another @font-face rule containing descriptors for bold text:

```
@font-face {
    font-family: myFirstFont;
    src: url(sansation_bold.woff);
    font-weight: bold;
}
```

[Try it yourself »](#)

The file "sansation_bold.woff" is another font file, that contains the bold characters for the Sansation font.

Browsers will use this whenever a piece of text with the font-family "myFirstFont" should render as bold.

This way you can have many @font-face rules for the same font.

Related Pages

CSS3 tutorial: [CSS3 Fonts](#)

 Previous

Complete  Reference

Next 

CSS font-family Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify the font for a paragraph:

```
p {  
    font-family: "Times New Roman", Georgia, Serif;  
}
```

[Try it yourself »](#)

Definition and Usage

The font-family property specifies the font for an element.

The font-family property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

There are two types of font family names:

- **family-name** - The name of a font-family, like "times", "courier", "arial", etc.
- **generic-family** - The name of a generic-family, like "serif", "sans-serif", "cursive", "fantasy", "monospace".

Start with the font you want, and always end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Note: Separate each value with a comma.

Note: If a font name contains white-space, it must be quoted. Single quotes must be used when using the "style" attribute in HTML.

Default value: *depends on the browser*

Inherited: yes

Animatable: no. [Read about animatable](#)**Version:** CSS1**JavaScript syntax:** `object.style.fontFamily="Verdana,sans-serif"`[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-family	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
font-family: font|initial|inherit;
```

Property Values

Value	Description	Play it
<i>family-name</i>	A prioritized list of font family names and/or generic family names	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Font](#)

CSS reference: [font property](#)

HTML DOM reference: [fontFamily property](#)

[Previous](#)

Complete [CSS](#) Reference

[Next](#)

CSS font-size Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the font size for different HTML elements:

```
h1 {  
    font-size: 250%;  
}  
  
h2 {  
    font-size: 200%;  
}  
  
p {  
    font-size: 100%;  
}
```

[Try it yourself »](#)

Definition and Usage

The font-size property sets the size of a font.

Default value: medium

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.fontSize="14px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-size	1.0	5.5	1.0	1.0	7.0

CSS Syntax

```
font-size:medium|xx-small|x-small|small|large|x-large|xx-large|smaller|larger|length|initial|inherit;
```

Property Values

Value	Description	Play it
medium	Sets the font-size to a medium size. This is default	Play it »
xx-small	Sets the font-size to an xx-small size	Play it »
x-small	Sets the font-size to an extra small size	Play it »
small	Sets the font-size to a small size	Play it »
large	Sets the font-size to a large size	Play it »
x-large	Sets the font-size to an extra large size	Play it »
xx-large	Sets the font-size to an xx-large size	Play it »
smaller	Sets the font-size to a smaller size than the parent element	Play it »
larger	Sets the font-size to a larger size than the parent element	Play it »
length	Sets the font-size to a fixed size in px, cm, etc.	Play it »
%	Sets the font-size to a percent of the parent element's font size	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Font](#)

CSS reference: [font property](#)

HTML DOM reference: [fontSize property](#)

 Previous

Complete  Reference

Next 

CSS3 font-size-adjust Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

By specifying the font-size-adjust property, the browser will adjust the font size to be the same regardless of the font family ("verdana" has the aspect value 0.58)

```
div {
    font-size-adjust: 0.58;
}
```

[Try it yourself »](#)

Definition and Usage

The font-size-adjust property gives you better control of the font size when the first selected font is not available.

When a font is not available, the browser uses the second specified font. This could result in a big change for the font size. To prevent this, use the font-size-adjust property.

All fonts have an "aspect value" which is the size-difference between the lowercase letter "x" and the uppercase letter "X".

When the browser knows the "aspect value" for the first selected font, the browser can figure out what font-size to use when displaying text with the second choice font.

Default value: none

Inherited: yes

Animatable: yes. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.fontSizeAdjust="0.58"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-size-adjust	Not supported	Not supported	3.0	Not supported	Not supported

Syntax

```
font-size-adjust: number|none|initial|inherit;
```

Property Values

Value	Description
<i>number</i>	Defines the aspect value to use
none	Default value. No font size adjustment
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

HTML DOM reference: [fontSizeAdjust property](#)

◀ Previous

Complete Reference

Next ▶

CSS3 font-stretch Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Make the text in all <div> elements wider:

```
div {
    font-stretch: expanded;
}
```

[Try it yourself »](#)

Definition and Usage

The font-stretch property allows you to make text wider or narrower.

Default value: normal

Inherited: yes

Animatable: yes. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.fontStretch = "expanded"`

Browser Support

None of the major browsers support the font-stretch property.

Property					
font-stretch	Not	Not	Not	Not	Not

	supported	supported	supported	supported	supported
--	-----------	-----------	-----------	-----------	-----------

Syntax

```
font-stretch: ultra-condensed|extra-condensed|condensed|semi-condensed|normal|semi-expanded|expanded|extra-expanded|ultra-expanded|initial|inherit;
```

Property Values

Value	Description
ultra-condensed	Makes the text as narrow as it gets
extra-condensed	Makes the text narrower than condensed, but not as narrow as ultra-condensed
condensed	Makes the text narrower than semi-condensed, but not as narrow as extra-condensed
semi-condensed	Makes the text narrower than normal, but not as narrow as condensed
normal	Default value. No font stretching
semi-expanded	Makes the text wider than normal, but not as wide as expanded
expanded	Makes the text wider than semi-expanded, but not as wide as extra-expanded
extra-expanded	Makes the text wider than expanded, but not as wide as ultra-expanded
ultra-expanded	Makes the text as wide as it gets
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

[Previous](#)

Complete CSS Reference

[Next](#)

CSS font-style Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set different font styles for three paragraphs:

```
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}
```

[Try it yourself »](#)

Definition and Usage

The font-style property specifies the font style for a text.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.fontStyle="italic"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-style	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
font-style: normal|italic|oblique|initial|inherit;
```

Property Values

Value	Description	Play it
normal	The browser displays a normal font style. This is default	Play it »
italic	The browser displays an italic font style	Play it »
oblique	The browser displays an oblique font style	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Font](#)

CSS reference: [font property](#)

HTML DOM reference: [fontStyle property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS font-variant Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set a paragraph to a small-caps font:

```
p.small {  
    font-variant: small-caps;  
}
```

[Try it yourself »](#)

Definition and Usage

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

The font-variant property specifies whether or not a text should be displayed in a small-caps font.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.fontVariant="small-caps"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-variant	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
font-variant: normal|small-caps|initial|inherit;
```

Property Values

Value	Description	Play it
normal	The browser displays a normal font. This is default	Play it »
small-caps	The browser displays a small-caps font	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Font](#)

CSS reference: [font property](#)

HTML DOM reference: [fontVariant property](#)

[◀ Previous](#)

Complete Reference

[Next ▶](#)

CSS font-weight Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set different font weight for three paragraphs:

```
p.normal {  
    font-weight: normal;  
}  
  
p.thick {  
    font-weight: bold;  
}  
  
p.thicker {  
    font-weight: 900;  
}
```

[Try it yourself »](#)

Definition and Usage

The font-weight property sets how thick or thin characters in text should be displayed.

Default value: normal

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.fontWeight = "bold"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
font-weight	2.0	4.0	1.0	1.3	3.5

CSS Syntax

```
font-weight: normal|bold|bolder|lighter|number|initial|inherit;
```

Property Values

Value	Description	Play it
normal	Defines normal characters. This is default	Play it »
bold	Defines thick characters	Play it »
bolder	Defines thicker characters	Play it »
lighter	Defines lighter characters	Play it »
100 200 300 400 500 600 700 800 900	Defines from thin to thick characters. 400 is the same as normal, and 700 is the same as bold	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Font](#)

CSS reference: [font property](#)

HTML DOM reference: [fontWeight property](#)

 Previous

Complete  Reference

Next 

CSS3 hanging-punctuation Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Place a punctuation mark (if any) outside the start edge of the first line in the <p> element:

```
p {
    hanging-punctuation: first;
}
```

Definition and Usage

The hanging-punctuation property specifies whether a punctuation mark may be placed outside the line box at the start or at the end of a full line of text.

Default value: none

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.hangingPunctuation="first"`

Browser Support

None of the major browsers support the hanging-punctuation property.

Property					
hanging-punctuation	Not supported				

CSS Syntax

```
hanging-punctuation: none|first|last|allow-end|force-end|initial|inherit;
```

Property Values

Value	Description
none	No punctuation mark may be placed outside the line box at the start or at the end of a full line of text
first	Punctuation may hang outside the start edge of the first line
last	Punctuation may hang outside the end edge of the last line
allow-end	Punctuation may hang outside the end edge of all lines if the punctuation does not otherwise fit prior to justification
force-end	Punctuation may hang outside the end edge of all lines. If justification is enabled on this line, then it will force the punctuation to hang
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

[« Previous](#)[Complete CSS Reference](#)[Next »](#)

CSS height Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the height and width of a paragraph:

```
p.ex {  
    height: 100px;  
    width: 100px;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The height property sets the height of an element.

Note: The height property does not include padding, borders, or margins; it sets the height of the area inside the padding, border, and margin of the element!

Note: The [min-height](#) and [max-height](#) properties override height.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.height="500px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
height	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
height: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	The browser calculates the height. This is default	Play it »
length	Defines the height in px, cm, etc.	Play it »
%	Defines the height in percent of the containing block	
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

Set the height of an element using a percent value:

```
img {
    height: 50%;
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS tutorial: [CSS Box model](#)

CSS reference: [width property](#)

HTML DOM reference: [height property](#)

 Previous

Complete  Reference

Next 

CSS3 icon Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Style an element with an iconic equivalent:

```
img {
    content: icon;
    icon: url(imgicon.png);
}
```

Definition and Usage

The icon property provides the author the ability to style an element with an iconic equivalent.

Note: An element's icon is not used unless the "content" property is set to the value "icon"!

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.icon="url(image.png)"`

Browser Support

None of the major browsers support the icon property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera

icon	Not supported				
------	---------------	---------------	---------------	---------------	---------------

CSS Syntax

```
icon: auto|URL|initial|inherit;
```

Property Values

Value	Description
auto	Uses a default generic icon provided by the browser
URL	Refers to one or more icons in a comma separated list
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS justify-content Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Make some space around the items of the flexible <div> element:

```
div {
    display: -webkit-flex; /* Safari */
    -webkit-justify-content: space-around; /* Safari 6.1+ */
    display: flex;
    justify-content: space-around;
}
```

[Try it yourself »](#)

Definition and Usage

The justify-content property aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

Tip: Use the [align-content](#) property to align the items on the cross-axis (vertically).

Default value: flex-start

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.justifyContent= "space-between"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
justify-content	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
justify-content: flex-start|flex-end|center|space-between|space-around|initial|inherit;
```

Property Values

Value	Description	Play it
flex-start	Default value. Items are positioned at the beginning of the container	Play it »
flex-end	Items are positioned at the end of the container	Play it »
center	Items are positioned at the center of the container	Play it »
space-between	Items are positioned with space between the lines	Play it »
space-around	Items are positioned with space before, between, and after the lines	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS Reference: [align-content property](#)

CSS Reference: [align-items property](#)

CSS Reference: [align-self property](#)

HTML DOM reference: [justifyContent property](#)

[◀ Previous](#)

Complete **CSS** Reference

[Next ▶](#)

CSS3 @keyframes Rule

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Make an element move gradually 200px down:

```
/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
    from {top: 0px;}
    to {top: 200px;}
}

/* Standard syntax */
@keyframes mymove {
    from {top: 0px;}
    to {top: 200px;}
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The @keyframes rule specifies the animation code.

The animation is created by gradually changing from one set of CSS styles to another.

During the animation, you can change the set of CSS styles many times.

Specify when the style change will happen in percent, or with the keywords "from" and "to", which is the same as 0% and 100%. 0% is the beginning of the animation, 100% is when the animation is complete.

Tip: For best browser support, you should always define both the 0% and the 100% selectors.

Note: Use the animation properties to control the appearance of the animation, and also to bind the animation to selectors.

Browser Support

The numbers in the table specifies the first browser version that fully supports the rule.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Rule					
@keyframes	4.0 -webkit-	10.0	16.0 5.0 -moz-	4.0 -webkit-	15.0 -webkit- 12.10 12.0 -o-

CSS Syntax

```
@keyframes animationname {keyframes-selector {css-styles;}}
```

Property Values

Value	Description
animationname	Required. Defines the name of the animation.
keyframes-selector	Required. Percentage of the animation duration. Legal values: 0-100% from (same as 0%) to (same as 100%) Note: You can have many keyframes-selectors in one animation.
css-styles	Required. One or more legal CSS style properties



More Examples

Example

Add many keyframe selectors in one animation:

```
/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
```

```

0%   {top: 0px;}
25%  {top: 200px;}
50%  {top: 100px;}
75%  {top: 200px;}
100% {top: 0px;}
}

/* Standard syntax */
@keyframes mymove {
  0%   {top: 0px;}
  25%  {top: 200px;}
  50%  {top: 100px;}
  75%  {top: 200px;}
  100% {top: 0px;}
}

```

[Try it yourself »](#)

Example

Change many CSS styles in one animation:

```

/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
  0%   {top: 0px; background: red; width: 100px;}
  100% {top: 200px; background: yellow; width: 300px;}
}

/* Standard syntax */
@keyframes mymove {
  0%   {top: 0px; background: red; width: 100px;}
  100% {top: 200px; background: yellow; width: 300px;}
}

```

[Try it yourself »](#)

Example

Many keyframe selectors with many CSS styles:

```

/* Chrome, Safari, Opera */
@-webkit-keyframes mymove {
  0%   {top: 0px; left: 0px; background: red;}
  25%  {top: 0px; left: 100px; background: blue;}
  50%  {top: 100px; left: 100px; background: yellow;}
  75%  {top: 100px; left: 0px; background: green;}
  100% {top: 0px; left: 0px; background: red;}
}

```

```
/* Standard syntax */  
@keyframes mymove {  
    0%   {top: 0px; left: 0px; background: red;}  
    25%  {top: 0px; left: 100px; background: blue;}  
    50%  {top: 100px; left: 100px; background: yellow;}  
    75%  {top: 100px; left: 0px; background: green;}  
    100% {top: 0px; left: 0px; background: red;}  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Animations](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS left Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the left edge of the image 5px to the right of the left edge of its containing element:

```
img {  
    position: absolute;  
    left: 5px;  
}
```

[Try it yourself »](#)

Definition and Usage

For absolutely positioned elements, the left property sets the left edge of an element to a unit to the left/right of the left edge of its containing element.

For relatively positioned elements, the left property sets the left edge of an element to a unit to the left/right to its normal position.

Note: If "position:static", the left property has no effect.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.left="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
left	1.0	5.5	1.0	1.0	5.0

CSS Syntax

```
left: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Lets the browser calculate the left edge position. This is default	Play it »
length	Sets the left edge position in px, cm, etc. Negative values are allowed	Play it »
%	Sets the left edge position in % of containing element. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

CSS reference: [right property](#)

CSS reference: [bottom property](#)

CSS reference: [top property](#)

HTML DOM reference: [left property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS left property

CSS letter-spacing Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the letter spacing for `<h1>` and `<h2>` elements:

```
h1 {  
    letter-spacing: 2px;  
}  
  
h2 {  
    letter-spacing: -3px;  
}
```

[Try it yourself »](#)

Definition and Usage

The letter-spacing property increases or decreases the space between characters in a text.

Default value: normal

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.letterSpacing="3px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
letter-spacing	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
letter-spacing: normal|length|initial|inherit;
```

Property Values

Value	Description	Play it
normal	No extra space between characters. This is default	Play it »
length	Defines an extra space between characters (negative values are allowed)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [letterSpacing property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS line-height Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the line height in percent:

```
p.small {  
    line-height: 90%;  
}  
  
p.big {  
    line-height: 200%;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The line-height property specifies the line height.

Note: Negative values are not allowed.

Default value: normal

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.lineHeight="30px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
line-height	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
line-height: normal|number|length|initial|inherit;
```

Property Values

Value	Description	Play it
normal	A normal line height. This is default	Play it »
number	A number that will be multiplied with the current font size to set the line height	Play it »
length	A fixed line height in px, pt, cm, etc.	Play it »
%	A line height in percent of the current font size	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

Specify the space between the lines in two paragraphs using a pixel value.

```
p.small {
    line-height: 10px;
}

p.big {
    line-height: 30px;
}
```

[Try it yourself »](#)

Example

Specify the space between the lines in two paragraphs using a number value.

```
p.small {  
    line-height: 0.5;  
}  
  
p.big {  
    line-height: 2;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [lineHeight property](#)

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS list-style Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify all the list properties in one declaration:

```
ul {
    list-style: square url("sqpurple.gif");
}
```

[Try it yourself »](#)

Definition and Usage

The list-style shorthand property sets all the list properties in one declaration.

The properties that can be set, are (in order): list-style-type, list-style-position, list-style-image.

If one of the values above are missing, e.g. "list-style:circle inside;", the default value for the missing property will be inserted, if any.

Default value: disc outside none

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.listStyle="decimal inside"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
list-style	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
list-style: list-style-type list-style-position list-style-image|initial|inherit;
```

Property Values

Value	Description
<i>list-style-type</i>	Specifies the type of list-item marker. See list-style-type for possible values
<i>list-style-position</i>	Specifies where to place the list-item marker. See list-style-position for possible values
<i>list-style-image</i>	Specifies the type of list-item marker. See list-style-image for possible values
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS List](#)

HTML DOM reference: [listStyle property](#)

Previous

Complete Reference

Next

CSS list-style-image Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify an image as the list-item marker in a list:

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```

[Try it yourself »](#)

Definition and Usage

The list-style-image property replaces the list-item marker with an image.

Note: Always specify the list-style-type property in addition. This property is used if the image for some reason is unavailable.

Default value: none

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.listStyleImage="url('smiley.gif')"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
list-style-image	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
list-style-image: none|url|initial|inherit;
```

Property Values

Value	Description	Play it
none	No image will be displayed. Instead, the list-style-type property will define what type of list marker will be rendered. This is default	Play it »
url	The path to the image to be used as a list-item marker	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS List](#)

CSS reference: [list-style property](#)

HTML DOM reference: [listStyleImage property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS list-style-position Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify that the list-item markers should appear inside the content flow (results in an extra indentation):

```
ul {  
    list-style-position: inside;  
}
```

[Try it yourself »](#)

Definition and Usage

The list-style-position property specifies if the list-item markers should appear inside or outside the content flow.

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

Default value: outside

Inherited: yes

Animatable: no. Read about *animatable***Version:** CSS1**JavaScript syntax:** `object.style.listStylePosition="inside"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
list-style-position	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
list-style-position: inside|outside|initial|inherit;
```

Property Values

Value	Description	Play it
inside	Indents the marker and the text. The bullets appear inside the content flow	Play it »
outside	Keeps the marker to the left of the text. The bullets appears outside the content flow. This is default	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS List](#)CSS reference: [list-style property](#)HTML DOM reference: [listStylePosition property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS list-style-type Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set some different list styles:

```
ul.circle {list-style-type: circle;}
ul.square {list-style-type: square;}
ol.upper-roman {list-style-type: upper-roman;}
ol.lower-alpha {list-style-type: lower-alpha;}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The list-style-type specifies the type of list-item marker in a list.

Default value: disc

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.listStyleType="square"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
list-style-type	1.0	4.0	1.0	1.0	3.5

Note: Internet Explorer and Opera 12 and earlier versions do not support the values: cjk-ideographic, hebrew, hiragana, hiragana-iroha, katakana, and katakana-iroha.

Note: IE8, and earlier, only support the property values: decimal-leading-zero, lower-greek, lower-latin, upper-latin, armenian, georgian, and inherit if a DOCTYPE is specified!

CSS Syntax

```
list-style-type: value;
```

Property Values

Value	Description	Play it
disc	Default value. The marker is a filled circle	Play it »
armenian	The marker is traditional Armenian numbering	Play it »
circle	The marker is a circle	Play it »
cjk-ideographic	The marker is plain ideographic numbers	Play it »
decimal	The marker is a number	Play it »
decimal-leading-zero	The marker is a number with leading zeros (01, 02, 03, etc.)	Play it »
georgian	The marker is traditional Georgian numbering	Play it »
hebrew	The marker is traditional Hebrew numbering	Play it »
hiragana	The marker is traditional Hiragana numbering	Play it »
hiragana-iroha	The marker is traditional Hiragana iroha numbering	Play it »
katakana	The marker is traditional Katakana numbering	Play it »
katakana-iroha	The marker is traditional Katakana iroha numbering	Play it »
lower-alpha	The marker is lower-alpha (a, b, c, d, e, etc.)	Play it »
lower-greek	The marker is lower-greek	Play it »

lower-latin	The marker is lower-latin (a, b, c, d, e, etc.)	Play it »
lower-roman	The marker is lower-roman (i, ii, iii, iv, v, etc.)	Play it »
none	No marker is shown	Play it »
square	The marker is a square	Play it »
upper-alpha	The marker is upper-alpha (A, B, C, D, E, etc.)	Play it »
upper-latin	The marker is upper-latin (A, B, C, D, E, etc.)	Play it »
upper-roman	The marker is upper-roman (I, II, III, IV, V, etc.)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

This example demonstrates all the different list-item markers:

```
ul.a {list-style-type: circle;}
ul.b {list-style-type: disc;}
ul.c {list-style-type: square;}

ol.d {list-style-type: armenian;}
ol.e {list-style-type: cjk-ideographic;}
ol.f {list-style-type: decimal;}
ol.g {list-style-type: decimal-leading-zero;}
ol.h {list-style-type: georgian;}
ol.i {list-style-type: hebrew;}
ol.j {list-style-type: hiragana;}
ol.k {list-style-type: hiragana-iroha;}
ol.l {list-style-type: katakana;}
ol.m {list-style-type: katakana-iroha;}
ol.n {list-style-type: lower-alpha;}
ol.o {list-style-type: lower-greek;}
ol.p {list-style-type: lower-latin;}
ol.q {list-style-type: lower-roman;}
ol.r {list-style-type: upper-alpha;}
ol.s {list-style-type: upper-latin;}
ol.t {list-style-type: upper-roman; }

ol.u {list-style-type: none;}
```

```
ol.v {list-style-type: inherit;}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS List](#)

CSS reference: [list-style property](#)

HTML DOM reference: [listStyleType property](#)

[« Previous](#)

[Complete CSS Reference](#)

[Next »](#)

CSS margin Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set all the four margins of a <p> element:

```
p {  
    margin: 2cm 4cm 3cm 4cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The margin shorthand property sets all the margin properties in one declaration. This property can have from one to four values.

Examples:

- **margin:10px 5px 15px 20px;**
 - top margin is 10px
 - right margin is 5px
 - bottom margin is 15px
 - left margin is 20px
- **margin:10px 5px 15px;**
 - top margin is 10px
 - right and left margins are 5px
 - bottom margin is 15px
- **margin:10px 5px;**
 - top and bottom margins are 10px
 - right and left margins are 5px

- margin:10px;**
- all four margins are 10px

Note: Negative values are allowed.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.margin="100px 50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
margin	1.0	6.0	1.0	1.0	3.5

CSS Syntax

```
margin: length|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a margin in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a margin in percent of the width of the containing element	Play it »
auto	The browser calculates a margin	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Margin](#)

CSS tutorial: [CSS Box Model](#)

HTML DOM reference: [margin property](#)

 Previous

Complete  Reference

Next 



CSS margin-bottom Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the bottom margin for a `<p>` element:

```
p {  
    margin-bottom: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The margin-bottom property sets the bottom margin of an element.

Note: Negative values are allowed.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.marginBottom="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
margin-bottom	1.0	6.0	1.0	1.0	3.5

CSS Syntax

```
margin-bottom: Length|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed bottom margin in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a bottom margin in percent of the width of the containing element	Play it »
auto	The browser calculates a bottom margin	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Margin](#)

HTML DOM reference: [marginBottom property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

WEB HOSTING
UK Reseller Hosting

WEB BUILDING
FREE Website BUILDER
Free HTML5 Templates

W3SCHOOLS EXAMS

HTML, CSS, JavaScript, PHP, jQuery, and XML Certifications

SHARE THIS PAGE

COLOR PICKER



[REPORT ERROR](#)
[PRINT PAGE](#)
[FORUM](#)
[ABOUT](#)

Top 10 Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [SQL Tutorial](#)
- [PHP Tutorial](#)
- [jQuery Tutorial](#)
- [Bootstrap Tutorial](#)
- [Angular Tutorial](#)
- [ASP.NET Tutorial](#)
- [XML Tutorial](#)

Top 10 References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [Browser Statistics](#)
- [HTML DOM](#)
- [PHP Reference](#)
- [jQuery Reference](#)
- [HTML Colors](#)
- [HTML Character Sets](#)
- [XML DOM](#)

Top 10 Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [HTML DOM Examples](#)
- [PHP Examples](#)
- [jQuery Examples](#)
- [XML Examples](#)
- [XML DOM Examples](#)
- [ASP Examples](#)
- [SVG Examples](#)

Web Certificates

- [HTML Certificate](#)
- [HTML5 Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [jQuery Certificate](#)
- [PHP Certificate](#)
- [XML Certificate](#)

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 1999-2015 by Refsnes Data. All Rights Reserved.



CSS margin-left Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the left margin for a <p> element:

```
p {  
    margin-left: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The margin-left property sets the left margin of an element.

Note: Negative values are allowed.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.marginLeft="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
----------	--------	-------------------	---------	--------	-------

margin-left	1.0	6.0	1.0	1.0	3.5
-------------	-----	-----	-----	-----	-----

CSS Syntax

```
margin-left: Length|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed left margin in px, pt, cm, etc. Default value is 0px	Play it »
%	Specifies a left margin in percent of the width of the containing element	Play it »
auto	The browser calculates a left margin	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Margin](#)

HTML DOM reference: [marginLeft property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS margin-right Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the right margin for a <p> element:

```
p {  
    margin-right: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The margin-right property sets the right margin of an element.

Note: Negative values are allowed.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.marginRight="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property					



margin-right	1.0	6.0	1.0	1.0	3.5
--------------	-----	-----	-----	-----	-----

CSS Syntax

```
margin-right: Length|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed right margin in px, pt, cm, etc. Default value is 0px	Play it »
%	Specifies a right margin in percent of the width of the containing element	Play it »
auto	The browser calculates a right margin	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Margin](#)

HTML DOM reference: [marginRight property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS margin-top Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the top margin for a <p> element:

```
p {  
    margin-top: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The margin-top property sets the top margin of an element.

Note: Negative values are allowed.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.marginTop="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property					



margin-top	1.0	6.0	1.0	1.0	3.5
------------	-----	-----	-----	-----	-----

CSS Syntax

```
margin-top: Length|auto|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed top margin in px, pt, cm, etc. Default value is 0px	Play it »
%	Specifies a top margin in percent of the width of the containing element	Play it »
auto	The browser calculates a top margin	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Margin](#)

HTML DOM reference: [marginTop property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS max-height Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the maximum height of a <p> element:

```
p {  
    max-height: 50px;  
}
```

[Try it yourself »](#)

Definition and Usage

The max-height property is used to set the maximum height of an element.

This prevents the value of the height property from becoming larger than max-height.

Note: The value of the max-height property overrides height.

Default value: none

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.maxHeight="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
max-height	1.0	7.0	1.0	2.0.2	7.0

CSS Syntax

```
max-height: none|Length|initial|inherit;
```

Property Values

Value	Description	Play it
none	No maximum height. This is default	Play it »
length	Defines the maximum height in px, cm, etc.	Play it »
%	Defines the maximum height in percent of the containing block	
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS reference: [min-height property](#)

HTML DOM reference: [maxHeight property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS max-width Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the maximum width of a <p> element:

```
p {  
    max-width: 100px;  
}
```

[Try it yourself »](#)

Definition and Usage

The max-width property is used to set the maximum width of an element.

This prevents the value of the width property from becoming larger than max-width.

Note: The value of the max-width property overrides width.

Default value: none

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.maxWidth="600px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
max-width	1.0	7.0	1.0	2.0.2	7.0

CSS Syntax

```
max-width: none|Length|initial|inherit;
```

Property Values

Value	Description	Play it
none	No maximum width. This is default	Play it »
length	Defines the maximum width in px, cm, etc.	Play it »
%	Defines the maximum width in percent of the containing block	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS reference: [min-width property](#)

HTML DOM reference: [maxWidth property](#)

[◀ Previous](#)

Complete Reference

[Next ▶](#)

CSS3 @media Rule

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Change the background-color if the document is smaller than 300 pixels wide:

```
@media screen and (max-width: 300px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The @media rule is used to define different style rules for different media types/devices.

In CSS2 this was called media types, while in CSS3 it is called media queries.

Media queries look at the capability of the device, and can be used to check many things, such as:

- width and height of the browser window
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution
- and much more

Browser Support

The numbers in the table specifies the first browser version that fully supports the @media rule.



Rule					
@media	21	9	3.5	4.0	9

CSS Syntax

```
@media not|only mediatype and (media feature) {
    CSS-Code;
}
```

You can also have different *stylesheets* for different media:

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)"
      href="mystylesheet.css">
```

Media Types

Value	Description
all	Used for all media type devices
aural	Deprecated. Used for speech and sound synthesizers
braille	Deprecated. Used for braille tactile feedback devices
embossed	Deprecated. Used for paged braille printers
handheld	Deprecated. Used for small or handheld devices
print	Used for printers
projection	Deprecated. Used for projected presentations, like slides
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud
tty	Deprecated. Used for media using a fixed-pitch character grid, like teletypes and terminals
tv	Deprecated. Used for television-type devices

Media Features

Value	Description
-------	-------------

aspect-ratio	Specifies the ratio between the width and the height of the display area
color	Specifies the number of bits per color component for the output device
color-index	Specifies the number of colors the device can display
device-aspect-ratio	Specifies the ratio between the width and the height of the device
device-height	Specifies the height of the device, such as a computer screen
device-width	Specifies the width of the device, such as a computer screen
grid	Specifies whether the device is a grid device or not
height	Specifies the height of the display area, such as a browser window
max-aspect-ratio	Specifies the minimum ratio between the width and the height of the display area
max-color	Specifies the maximum number of bits per color component for the output device
max-color-index	Specifies the maximum number of colors the device can display
max-device-aspect-ratio	Specifies the minimum ratio between the width and the height of the device
max-device-height	Specifies the maximum height of the device, such as a computer screen
max-device-width	Specifies the maximum width of the device, such as a computer screen
max-height	Specifies the maximum height of the display area, such as a browser window
max-monochrome	Specifies the maximum number of bits per "color" on a monochrome (greyscale) device
max-resolution	Specifies the maximum resolution of the device, using dpi or dpcm
max-width	Specifies the maximum width of the display area, such as a browser window

min-aspect-ratio	Specifies the minimum ratio between the width and the height of the display area
min-color	Specifies the minimum number of bits per color component for the output device
min-color-index	Specifies the minimum number of colors the device can display
min-device-aspect-ratio	Specifies the minimum ratio between the width and the height of the device
min-device-width	Specifies the minimum width of the device, such as a computer screen
min-device-height	Specifies the minimum height of the device, such as a computer screen
min-height	Specifies the minimum height of the display area, such as a browser window
min-monochrome	Specifies the minimum number of bits per "color" on a monochrome (greyscale) device
min-resolution	Specifies the minimum resolution of the device, using dpi or dpcm
min-width	Specifies the minimum width of the display area, such as a browser window
monochrome	Specifies the number of bits per "color" on a monochrome (greyscale) device
orientation	Specifies the whether the display is in landscape mode or portrait mode
resolution	Specifies the resolution of the device, using dpi or dpcm
scan	Specifies progressive or interlaced scanning of a television
width	Specifies the width of the display area, such as a browser window



More Examples

Example

Use the @media rule to make responsive design:

```
@media only screen and (max-width: 500px) {  
    .gridmenu {  
        width:100%;  
    }  
  
    .gridmain {  
        width:100%;  
    }  
  
    .gridright {  
        width:100%;  
    }  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Media Types](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#) 

CSS min-height Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the minimum height of a <p> element:

```
p {  
    min-height: 100px;  
}
```

[Try it yourself »](#)

Definition and Usage

The min-height property is used to set the minimum height of an element.

This prevents the value of the height property from becoming smaller than min-height.

Note: The value of the min-height property overrides both max-height and height.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.minHeight="400px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
min-height	1.0	7.0	3.0	2.0.2	4.0

CSS Syntax

```
min-height: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Default value is 0. Defines the minimum height in px, cm, etc.	Play it »
%	Defines the minimum height in percent of the containing block	
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS reference: [max-height property](#)

HTML DOM reference: [minHeight property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS min-width Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the minimum width of a <p> element:

```
p {  
    min-width: 150px;  
}
```

[Try it yourself »](#)

Definition and Usage

The min-width property is used to set the minimum width of an element.

This prevents the value of the width property from becoming smaller than min-width.

Note: The value of the min-width property overrides both max-width and width.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.minWidth= "400px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
min-width	1.0	7.0	1.0	2.0.2	4.0

CSS Syntax

```
min-width: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Default value is 0. Defines the minimum width in px, cm, etc.	Play it »
%	Defines the minimum width in percent of the containing block	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS reference: [max-width property](#)

HTML DOM reference: [minWidth property](#)

[◀ Previous](#)

Complete [CSS Reference](#)

[Next ▶](#)

CSS3 nav-down Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify where to navigate when using the arrow keys:

```
button#b1 {  
    top: 20%; left: 25%;  
    nav-index: 1;  
    nav-right: #b2; nav-left: #b4;  
    nav-down: #b2; nav-up: #b4;  
}  
  
button#b2 {  
    top: 40%; left: 50%;  
    nav-index: 2;  
    nav-right: #b3; nav-left: #b1;  
    nav-down: #b3; nav-up: #b1;  
}  
  
button#b3 {  
    top: 70%; left: 25%;  
    nav-index: 3;  
    nav-right: #b4; nav-left: #b2;  
    nav-down: #b4; nav-up: #b2;  
}  
  
button#b4 {  
    top: 40%; left: 0%;  
    nav-index: 4;  
    nav-right: #b1; nav-left: #b3;  
    nav-down: #b1; nav-up: #b3;  
}
```

[Try it yourself »](#)

Definition and Usage

The nav-down property specifies where to navigate when using the arrow-down navigation key.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.navDown="#div2"`

Browser Support

The nav-down property is only supported in Opera 12 and some earlier versions. This property became deprecated and obsolete in Opera 15.

Property					
nav-down	Not supported	Not supported	Not supported	Not supported	Supported in Opera 11.5-12. Not supported in newer versions

CSS Syntax

```
nav-down: auto|id|target-name|initial|inherit;
```

Property Values

Value	Description
auto	Default value. The browser determines which element to navigate to
<i>id</i>	Specifies the id of the element to navigate to
<i>target-name</i>	Specifies the target frame to navigate to
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

 Previous

Complete CSS Reference

Next 

CSS3 nav-index Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify where to navigate when using the arrow keys:

```
button#b1 {  
    top: 20%; left: 25%;  
    nav-index: 1;  
    nav-right: #b2; nav-left: #b4;  
    nav-down: #b2; nav-up: #b4;  
}  
  
button#b2 {  
    top: 40%; left: 50%;  
    nav-index: 2;  
    nav-right: #b3; nav-left: #b1;  
    nav-down: #b3; nav-up: #b1;  
}  
  
button#b3 {  
    top: 70%; left: 25%;  
    nav-index: 3;  
    nav-right: #b4; nav-left: #b2;  
    nav-down: #b4; nav-up: #b2;  
}  
  
button#b4 {  
    top: 40%; left: 0%;  
    nav-index: 4;  
    nav-right: #b1; nav-left: #b3;  
    nav-down: #b1; nav-up: #b3;  
}
```

[Try it yourself »](#)

Definition and Usage

The `nav-index` property specifies the sequential navigation order ("tabbing order") for an element.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.navIndex=2`

Browser Support

The `nav-index` property is only supported in Opera 12 and some earlier versions. This property became deprecated and obsolete in Opera 15.

Property					
<code>nav-index</code>	Not supported	Not supported	Not supported	Not supported	Supported in Opera 11.5-12. Not supported in newer versions

CSS Syntax

```
nav-index: auto|number|initial|inherit;
```

Property Values

Value	Description
<code>auto</code>	Default value. The element's tabbing order is assigned by the browser
<code>number</code>	Indicates the tabbing order for the element. 1 means first
<code>initial</code>	Sets this property to its default value. Read about initial
<code>inherit</code>	Inherits this property from its parent element. Read about inherit

 Previous

Complete  Reference

Next 

CSS3 nav-left Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify where to navigate when using the arrow keys:

```
button#b1 {  
    top: 20%; left: 25%;  
    nav-index: 1;  
    nav-right: #b2; nav-left: #b4;  
    nav-down: #b2; nav-up: #b4;  
}  
  
button#b2 {  
    top: 40%; left: 50%;  
    nav-index: 2;  
    nav-right: #b3; nav-left: #b1;  
    nav-down: #b3; nav-up: #b1;  
}  
  
button#b3 {  
    top: 70%; left: 25%;  
    nav-index: 3;  
    nav-right: #b4; nav-left: #b2;  
    nav-down: #b4; nav-up: #b2;  
}  
  
button#b4 {  
    top: 40%; left: 0%;  
    nav-index: 4;  
    nav-right: #b1; nav-left: #b3;  
    nav-down: #b1; nav-up: #b3;  
}
```

[Try it yourself »](#)

Definition and Usage

The nav-left property specifies where to navigate when using the arrow-left navigation key.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.navLeft="#div2"`

Browser Support

The nav-left property is only supported in Opera 12 and some earlier versions. This property became deprecated and obsolete in Opera 15.

Property					
nav-left	Not supported	Not supported	Not supported	Not supported	Supported in Opera 11.5-12. Not supported in newer versions

CSS Syntax

```
nav-left: auto|id|target-name|initial|inherit;
```

Property Values

Value	Description
auto	Default value. The browser determines which element to navigate to
<i>id</i>	Specifies the id of the element to navigate to
<i>target-name</i>	Specifies the target frame to navigate to
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

 Previous

Complete  Reference

Next 

CSS3 nav-right Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify where to navigate when using the arrow keys:

```
button#b1 {  
    top: 20%; left: 25%;  
    nav-index: 1;  
    nav-right: #b2; nav-left: #b4;  
    nav-down: #b2; nav-up: #b4;  
}  
  
button#b2 {  
    top: 40%; left: 50%;  
    nav-index: 2;  
    nav-right: #b3; nav-left: #b1;  
    nav-down: #b3; nav-up: #b1;  
}  
  
button#b3 {  
    top: 70%; left: 25%;  
    nav-index: 3;  
    nav-right: #b4; nav-left: #b2;  
    nav-down: #b4; nav-up: #b2;  
}  
  
button#b4 {  
    top: 40%; left: 0%;  
    nav-index: 4;  
    nav-right: #b1; nav-left: #b3;  
    nav-down: #b1; nav-up: #b3;  
}
```

[Try it yourself »](#)

Definition and Usage

The nav-right property specifies where to navigate when using the arrow-right navigation key.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.navRight="#div2"`

Browser Support

The nav-right property is only supported in Opera 12 and some earlier versions. This property became deprecated and obsolete in Opera 15.

Property					
nav-right	Not supported	Not supported	Not supported	Not supported	Supported in Opera 11.5-12. Not supported in newer versions

CSS Syntax

```
nav-right: auto|id|target-name|initial|inherit;
```

Property Values

Value	Description
auto	Default value. The browser determines which element to navigate to
<i>id</i>	Specifies the id of the element to navigate to
<i>target-name</i>	Specifies the target frame to navigate to
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

 Previous

Complete  Reference

Next 

CSS3 nav-up Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Specify where to navigate when using the arrow keys:

```
button#b1 {  
    top: 20%; left: 25%;  
    nav-index: 1;  
    nav-right: #b2; nav-left: #b4;  
    nav-down: #b2; nav-up: #b4;  
}  
  
button#b2 {  
    top: 40%; left: 50%;  
    nav-index: 2;  
    nav-right: #b3; nav-left: #b1;  
    nav-down: #b3; nav-up: #b1;  
}  
  
button#b3 {  
    top: 70%; left: 25%;  
    nav-index: 3;  
    nav-right: #b4; nav-left: #b2;  
    nav-down: #b4; nav-up: #b2;  
}  
  
button#b4 {  
    top: 40%; left: 0%;  
    nav-index: 4;  
    nav-right: #b1; nav-left: #b3;  
    nav-down: #b1; nav-up: #b3;  
}
```

[Try it yourself »](#)

Definition and Usage

The nav-up property specifies where to navigate when using the arrow-up navigation key.

Default value: auto

Inherited: no

Version: CSS3

JavaScript syntax: `object.style.navUp="#div2"`

Browser Support

The nav-up property is only supported in Opera 12 and some earlier versions. This property became deprecated and obsolete in Opera 15.

Property						
nav-up	Not supported	Not supported	Not supported	Not supported	Supported in Opera 11.5-12. Not supported in newer versions	

CSS Syntax

```
nav-up: auto|id|target-name|initial|inherit;
```

Property Values

Value	Description
auto	Default value. The browser determines which element to navigate to
<i>id</i>	Specifies the id of the element to navigate to
<i>target-name</i>	Specifies the target frame to navigate to
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

 Previous

Complete  Reference

Next 

CSS3 opacity Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the opacity level for a <div> element:

```
div {  
    opacity: 0.5;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The opacity property sets the opacity level for an element.

The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.

Default value: 1

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.opacity="0.5"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
opacity	4.0	9.0	2.0	3.1	9.0

Note: IE8 and earlier versions supports an alternative, the filter property. Like:
filter: Alpha(opacity=50).

CSS Syntax

```
opacity: number|initial|inherit;
```

Property Values

Value	Description	Play it
number	Specifies the opacity. From 0.0 (fully transparent) to 1.0 (fully opaque)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

How to use JavaScript to change the opacity for an element:

```
function myFunction(x) {
// Return the text of the selected option
  var opacity = x.options[x.selectedIndex].text;
  var el = document.getElementById("p1");
  if (el.style.opacity !== undefined) {
    el.style.opacity = opacity;
  } else {
    alert("Your browser doesn't support this example!");
  }
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Image Opacity / Transparency](#)

HTML DOM reference: [opacity property](#)

 Previous

Complete  Reference

Next 

CSS order Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the order of the flexible items:

```
/* Code for Safari 6.1 and newer */
div#myRedDIV {-webkit-order: 2;}
div#myBlueDIV {-webkit-order: 4;}
div#myGreenDIV {-webkit-order: 3;}
div#myPinkDIV {-webkit-order: 1;}

/* Standard syntax */
div#myRedDIV {order: 2;}
div#myBlueDIV {order: 4;}
div#myGreenDIV {order: 3;}
div#myPinkDIV {order: 1;}
```

[Try it yourself »](#)

Definition and Usage

The order property specifies the order of a flexible item relative to the rest of the flexible items inside the same container.

Note: If the element is not a flexible item, the order property has no effect.

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.order="2"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
order	29.0 21.0 -webkit-	11.0	28.0 18.0 -moz-	6.1 -webkit-	12.10

CSS Syntax

```
order: number|initial|inherit;
```

Property Values

Value	Description
number	Default value 0. Specifies the order for the flexible item
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS Reference: [flex property](#)

CSS Reference: [flex-basis property](#)

CSS Reference: [flex-direction property](#)

CSS Reference: [flex-flow property](#)

CSS Reference: [flex-grow property](#)

CSS Reference: [flex-shrink property](#)

CSS Reference: [flex-wrap property](#)

CSS Reference: [align-content property](#)

CSS Reference: [align-items property](#)

CSS Reference: [align-self property](#)

HTML DOM reference: [order property](#)

 Previous

Complete  Reference

Next 

CSS outline Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the outline around a <p> element:

```
p {
    outline: #00FF00 dotted thick;
}
```

[Try it yourself »](#)

Definition and Usage

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

The outline shorthand property sets all the outline properties in one declaration.

The properties that can be set, are (in order): outline-color, outline-style, outline-width.

If one of the values above are missing, e.g. "outline:solid #ff0000;", the default value for the missing property will be inserted, if any.

Note: The outline is not a part of the element's dimensions, therefore the element's width and height properties do not contain the width of the outline.

Default value: invert none medium

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.outline="#0000FF dotted 5px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
outline	1.0	8.0	1.5	1.2	7.0

Note: IE8 supports the outline property only if a !DOCTYPE is specified.

CSS Syntax

```
outline: outline-color outline-style outline-width|initial|inherit;
```

Property Values

Value	Description
<i>outline-color</i>	Specifies the color of the outline
<i>outline-style</i>	Specifies the style of the outline
<i>outline-width</i>	Specifies the width of outline
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS Outline](#)

HTML DOM reference: [outline property](#)

◀ Previous

Complete Reference

Next ▶

CSS outline-color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the color of a dotted outline:

```
p {
    outline-style: dotted;
    outline-color: #00ff00;
}
```

[Try it yourself »](#)

Definition and Usage

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

The outline-color property specifies the color of an outline.

Note: Always declare the outline-style property before the outline-color property. An element must have an outline before you change the color of it.

Default value: invert

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.outlineColor="#FF0000"`

[Try it](#)

Tips and Notes

An outline is a line around an element. It is displayed around the margin of the element. However, it is different from the border property.

The outline is not a part of the element's dimensions, therefore the element's width and height properties do not contain the width of the outline.

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
outline-color	1.0	8.0	1.5	1.2	7.0

Note: IE8 supports the outline-color property only if a !DOCTYPE is specified.

CSS Syntax

```
outline-color: invert|color|initial|inherit;
```

Property Values

Value	Description	Play it
invert	Performs a color inversion. This ensures that the outline is visible, regardless of color background. This is default	Play it »
color	Specifies the color of the outline. Look at CSS Color Values for a complete list of possible color values.	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Outline](#)

CSS reference: [outline property](#)

[« Previous](#)

[Complete CSS Reference](#)

[Next »](#)

CSS3 outline-offset Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify an outline 15px outside the border edge:

```
div {
    border: 2px solid black;
    outline: 2px solid red;
    outline-offset: 15px;
}
```

[Try it yourself »](#)

Definition and Usage

The outline-offset property offsets an outline, and draws it beyond the border edge.

Outlines differ from borders in two ways:

- Outlines do not take up space
- Outlines may be non-rectangular

Default value: 0

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.outlineOffset="15px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
outline-offset	4.0	Not supported	3.5	3.1	10.5

CSS Syntax

```
outline-offset: length|initial|inherit;
```

Property Values

Value	Description
<i>length</i>	The distance the outline is outset from the border edge. Default value is 0
<i>initial</i>	Sets this property to its default value. Read about <i>initial</i>
<i>inherit</i>	Inherits this property from its parent element. Read about <i>inherit</i>

Related Pages

CSS3 tutorial: [CSS3 User Interface](#)

HTML DOM reference: [outlineOffset property](#)

Previous

Complete Reference

Next

CSS outline-style Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the style of an outline:

```
p {  
    outline-style: dotted;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

The outline-style property specifies the style of an outline.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.outlineStyle="dashed"`

[Try it](#)

Tips and Notes

An outline is a line around an element. It is displayed around the margin of the element. However, it is different from the border property.

The outline is not a part of the element's dimensions, therefore the element's width and height properties do not contain the width of the outline.

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
outline-style	1.0	8.0	1.5	1.2	7.0

Note: IE8 supports the outline-style property only if a !DOCTYPE is specified.

CSS Syntax

```
outline-style:  
none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit;
```

Property Values

Value	Description	Play it
none	Specifies no outline. This is default	Play it »
hidden	Specifies a hidden outline	Play it »
dotted	Specifies a dotted outline	Play it »
dashed	Specifies a dashed outline	Play it »
solid	Specifies a solid outline	Play it »
double	Specifies a double outline	Play it »
groove	Specifies a 3D grooved outline. The effect depends on the outline-color value	Play it »
ridge	Specifies a 3D ridged outline. The effect depends on the outline-color value	Play it »
inset	Specifies a 3D inset outline. The effect depends on the outline-color value	Play it »

outset	Specifies a 3D outset outline. The effect depends on the outline-color value	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

More Examples

Example

Set the style of an outline using different values:

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Outline](#)

CSS reference: [outline property](#)

HTML DOM reference: [outlineStyle property](#)

[◀ Previous](#)

Complete  Reference

[Next ▶](#)

CSS outline-width Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the width of an outline:

```
p {  
    outline-style: dotted;  
    outline-width: 5px;  
}
```

[Try it yourself »](#)

Definition and Usage

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".

The outline-width specifies the width of an outline.

Note: Always declare the outline-style property before the outline-width property. An element must have an outline before you change the width of it.

Default value: medium

Inherited: no

Animatable: yes, see *individual properties*. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.outlineWidth="2px"`

[Try it](#)

Tips and Notes

An outline is a line around an element. It is displayed around the margin of the element. However, it is different from the border property.

The outline is not a part of the element's dimensions, therefore the element's width and height properties do not contain the width of the outline.

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
outline-width	1.0	8.0	1.5	1.2	7.0

Note: IE8 supports the outline-width property only if a !DOCTYPE is specified.

CSS Syntax

```
outline-width: medium|thin|thick|Length|initial|inherit;
```

Property Values

Value	Description	Play it
medium	Specifies a medium outline. This is default	Play it »
thin	Specifies a thin outline	Play it »
thick	Specifies a thick outline	Play it »
length	Allows you to define the thickness of the outline	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Outline](#)

CSS reference: [outline property](#)

HTML DOM reference: [outlineWidth property](#)

 Previous

Complete  Reference

Next 

CSS overflow Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the overflow property to scroll:

```
div {
    width: 150px;
    height: 150px;
    overflow: scroll;
}
```

[Try it yourself »](#)

Definition and Usage

The overflow property specifies what happens if content overflows an element's box.

Default value: visible

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.overflow="scroll"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



overflow	1.0	4.0	1.0	1.0	7.0
----------	-----	-----	-----	-----	-----

CSS Syntax

```
overflow: visible|hidden|scroll|auto|initial|inherit;
```

Property Values

Value	Description	Play it
visible	The overflow is not clipped. It renders outside the element's box. This is default	Play it »
hidden	The overflow is clipped, and the rest of the content will be invisible	Play it »
scroll	The overflow is clipped, but a scroll-bar is added to see the rest of the content	Play it »
auto	If overflow is clipped, a scroll-bar should be added to see the rest of the content	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Positioning](#)

HTML DOM reference: [overflow property](#)

 [Previous](#)

[Complete CSS Reference](#)

[Next](#) 

CSS3 overflow-x Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Clip the left/right edges of the content inside the <div> element - if it overflows the element's content area:

```
div {
  overflow-x: hidden;
}
```

[Try it yourself »](#)

Definition and Usage

The overflow-x property specifies what to do with the left/right edges of the content - if it overflows the element's content area.

Tip: Use the [overflow-y](#) property to determine clipping at the top and bottom edges.

Default value: visible

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.overflowX="scroll"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
overflow-x	4.0	9.0	3.5	3.0	9.5

CSS Syntax

```
overflow-x: visible|hidden|scroll|auto|initial|inherit;
```

Property Values

Value	Description	Play it
visible	The content is not clipped, and it may be rendered outside the content box	Play it »
hidden	The content is clipped - and no scrolling mechanism is provided	Play it »
scroll	The content is clipped and a scrolling mechanism is provided	Play it »
auto	Should cause a scrolling mechanism to be provided for overflowing boxes	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

HTML DOM reference: [overflowX property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

CSS3 overflow-y Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Clip the top/bottom edges of the content inside the <div> element - if it overflows the element's content area:

```
div {
  overflow-y: hidden;
}
```

[Try it yourself »](#)

Definition and Usage

The overflow-y property specifies what to do with the top/bottom edges of the content - if it overflows the element's content area.

Tip: Use the [overflow-x](#) property to determine clipping at the left and right edges.

Default value: visible

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.overflowY="scroll"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
overflow-y	4.0	9.0	3.5	3.0	9.5

CSS Syntax

```
overflow-y: visible|hidden|scroll|auto|initial|inherit;
```

Property Values

Value	Description	Play it
visible	Default value. The content is not clipped, and it may be rendered outside the content box	Play it »
hidden	The content is clipped - and no scrolling mechanism is provided	Play it »
scroll	The content is clipped and a scrolling mechanism is provided	Play it »
auto	Should cause a scrolling mechanism to be provided for overflowing boxes	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

HTML DOM reference: [overflowY property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

CSS padding Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the padding of a <p> element:

```
p {  
    padding: 2cm 4cm 3cm 4cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The padding shorthand property sets all the padding properties in one declaration. This property can have from one to four values.

Examples:

- **padding:10px 5px 15px 20px;**
 - top padding is 10px
 - right padding is 5px
 - bottom padding is 15px
 - left padding is 20px
- **padding:10px 5px 15px;**
 - top padding is 10px
 - right and left padding are 5px
 - bottom padding is 15px
- **padding:10px 5px;**
 - top and bottom padding are 10px
 - right and left padding are 5px

padding:10px;

- all four paddings are 10px

Note: Negative values are not allowed.**Default value:** 0**Inherited:** no**Animatable:** yes, see *individual properties*. [Read about animatable](#)**Try it****Version:** CSS1**JavaScript syntax:** `object.style.padding="100px 20px"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
padding	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
padding: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies the padding in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies the padding in percent of the width of the containing element	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Padding](#)

CSS tutorial: [CSS Box Model](#)

HTML DOM reference: [padding property](#)

 Previous

Complete  Reference

Next 

CSS padding-bottom Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the bottom padding for a <p> element:

```
p {  
    padding-bottom: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The padding-bottom property sets the bottom padding (space) of an element.

Note: Negative values are not allowed.

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.paddingBottom="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



padding-bottom	1.0	4.0	1.0	1.0	3.5
----------------	-----	-----	-----	-----	-----

CSS Syntax

```
padding-bottom: length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed bottom padding in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a bottom padding in percent of the width of the containing element	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Padding](#)

HTML DOM reference: [paddingBottom property](#)

 Previous

Complete  Reference

Next 

CSS padding-left Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the left padding for a <p> element:

```
p {  
    padding-left: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The padding-left property sets the left padding (space) of an element.

Note: Negative values are not allowed.

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.paddingLeft="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



padding-left	1.0	4.0	1.0	1.0	3.5
--------------	-----	-----	-----	-----	-----

CSS Syntax

```
padding-left: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed left padding in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a left padding in percent of the width of the containing element	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Padding](#)

HTML DOM reference: [paddingLeft property](#)

 Previous

Complete  Reference

Next 

CSS padding-right Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the right padding for a <p> element:

```
p {  
    padding-right: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The padding-right property sets the right padding (space) of an element.

Note: Negative values are not allowed.

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.paddingRight="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



padding-right	1.0	4.0	1.0	1.0	3.5
---------------	-----	-----	-----	-----	-----

CSS Syntax

```
padding-right: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed right padding in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a right padding in percent of the width of the containing element	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Padding](#)

HTML DOM reference: [paddingRight property](#)

 [Previous](#)

[Complete CSS Reference](#)

[Next](#) 

CSS padding-top Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the top padding for a <p> element:

```
p {  
    padding-top: 2cm;  
}
```

[Try it yourself »](#)

Definition and Usage

The padding-top property sets the top padding (space) of an element.

Note: Negative values are not allowed.

Default value: 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.paddingTop="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	1	1	1	1	1



padding-top	1.0	4.0	1.0	1.0	3.5
-------------	-----	-----	-----	-----	-----

CSS Syntax

```
padding-top: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Specifies a fixed top padding in px, pt, cm, etc. Default value is 0	Play it »
%	Specifies a top padding in percent of the width of the containing element	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Padding](#)

HTML DOM reference: [paddingTop property](#)

 Previous

Complete  Reference

Next 

CSS page-break-after Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Always insert a page break after each <footer> element (when printing):

```
@media print {
    footer {page-break-after: always;}
}
```

Definition and Usage

The page-break-after property sets whether a page break should occur AFTER a specified element.

Note: You cannot use this property on an empty <div> or on absolutely positioned elements.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.pageBreakAfter="always"`

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
----------	--	--	--	--	--

page-break-after	1.0	4.0	1.0	1.2	7.0
------------------	-----	-----	-----	-----	-----

Note: Internet Explorer and Firefox do not support the property values "left" or "right".

CSS Syntax

```
page-break-after: auto|always|avoid|left|right|initial|inherit;
```

Property Values

Value	Description
auto	Default. Automatic page breaks
always	Always insert a page break after the element
avoid	Avoid page break after the element (if possible)
left	Insert page breaks after the element so that the next page is formatted as a left page
right	Insert page breaks after the element so that the next page is formatted as a right page
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

HTML DOM reference: [pageBreakAfter property](#)

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS page-break-before Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Always insert a page break before each <h1> element (when printing):

```
@media print {
    h1 {page-break-before: always;}
}
```

Definition and Usage

The page-break-before property sets whether a page break should occur BEFORE a specified element.

Note: You cannot use this property on an empty <div> or on absolutely positioned elements.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.pageBreakBefore="always"`

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
----------	--	--	--	--	--

page-break-before	1.0	4.0	1.0	1.2	7.0
-------------------	-----	-----	-----	-----	-----

Note: Internet Explorer and Firefox do not support the property values "left" or "right".

CSS Syntax

```
page-break-before: auto|always|avoid|left|right|initial|inherit;
```

Property Values

Value	Description
auto	Default. Automatic page breaks
always	Always insert a page break before the element
avoid	Avoid page break before the element (if possible)
left	Insert page breaks before the element so that the next page is formatted as a left page
right	Insert page breaks before the element so that the next page is formatted as a right page
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

HTML DOM reference: [pageBreakBefore property](#)

◀ Previous

Complete CSS Reference

Next ▶

CSS page-break-inside Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Avoid page break inside paragraphs, across pages (when printing):

```
@media print {
    p {page-break-inside: avoid;}
}
```

Definition and Usage

The page-break-inside property sets whether a page break is allowed inside a specified element.

Note: You cannot use this property on absolutely positioned elements.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.pageBreakInside="avoid"`

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
page-break-	1.0	8.0	19.0	1.3	7.0

inside

CSS Syntax

```
page-break-inside: auto|avoid|initial|inherit;
```

Property Values

Value	Description
auto	Default. Automatic page breaks
avoid	Avoid page break inside the element (if possible)
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

HTML DOM reference: [pageBreakInside property](#)

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS3 perspective Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the perspective from where an element is viewed:

```
div {
    -webkit-perspective: 500px; /* Chrome, Safari, Opera */
    perspective: 500px;
}
```

[Try it yourself »](#)

Definition and Usage

The perspective property defines how many pixels a 3D element is placed from the view. This property allows you to change the perspective on how 3D elements are viewed.

When defining the perspective property for an element, it is the CHILD elements that get the perspective view, NOT the element itself.

Note: The perspective property only affects 3D transformed elements!

Tip: Use this property together with the [perspective-origin](#) property, which allows you to change the bottom position of 3D elements.

To better understand the perspective property, [view a demo](#).

Default value: none

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3



JavaScript syntax: `object.style.perspective="50px"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
perspective	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	4.0.3 -webkit-	23.0 15.0 -webkit-

CSS Syntax

```
perspective: Length | none;
```

Property Values

Property Value	Description
<code>length</code>	How far the element is placed from the view
<code>none</code>	Default value. Same as 0. The perspective is not set
<code>initial</code>	Sets this property to its default value. Read about initial
<code>inherit</code>	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS 3D Transforms](#)

HTML DOM reference: [perspective property](#)

Previous

Complete Reference

Next

CSS3 perspective property

CSS3 perspective-origin Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set a 3D element's base placement:

```
div {
    -webkit-perspective: 150px; /* Chrome, Safari, Opera */
    -webkit-perspective-origin: 10% 10%; /* Chrome, Safari, Opera */
    perspective: 150px;
    perspective-origin: 10% 10%;
}
```

[Try it yourself »](#)

Definition and Usage

The perspective-origin property defines where a 3D element is based in the x- and the y-axis. This property allows you to change the bottom position of 3D elements.

When defining the perspective-origin property for an element, it is the CHILD elements that are positioned, NOT the element itself.

Note: This property must be used together with the [perspective](#) property, and only affects 3D transformed elements!

To better understand the perspective-origin property, [view a demo](#).

Default value: 50% 50%

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3



JavaScript syntax: object.style.perspectiveOrigin="10px 50%"**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
perspective-origin	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	4.0.3 -webkit-	23.0 15.0 -webkit-

CSS Syntax

```
perspective-origin: x-axis y-axis|initial|inherit;
```

Property Values

Property Value	Description
<i>x-axis</i>	<p>Defining where the view is placed at the x-axis</p> <p>Possible values:</p> <ul style="list-style-type: none"> • left • center • right • <i>length</i> • % <p>Default value: 50%</p>
<i>y-axis</i>	<p>Defining where the view is placed at the y-axis</p> <p>Possible values:</p> <ul style="list-style-type: none"> • top • center • bottom • <i>length</i> • % <p>Default value: 50%</p>

initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

Related Pages

CSS tutorial: [CSS 3D Transforms](#)

HTML DOM reference: [perspectiveOrigin property](#)

 Previous

Complete  Reference

Next 

CSS position Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Position an `<h2>` element:

```
h2 {  
    position: absolute;  
    left: 100px;  
    top: 150px;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The position property specifies the type of positioning method used for an element (static, relative, absolute or fixed).

Default value: static

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.position="absolute"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
position	1.0	7.0	1.0	1.0	4.0

CSS Syntax

```
position: static|absolute|fixed|relative|initial|inherit;
```

Property Values

Value	Description	Play it
static	Default value. Elements render in order, as they appear in the document flow	Play it »
absolute	The element is positioned relative to its first positioned (not static) ancestor element	Play it »
fixed	The element is positioned relative to the browser window	Play it »
relative	The element is positioned relative to its normal position, so "left:20" adds 20 pixels to the element's LEFT position	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

How to position an element relative to its normal position:

```

h2.pos_left {
    position: relative;
    left: -20px;
}

h2.pos_right {
    position: relative;
    left: 20px;
}

```

}

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Positioning](#)

HTML DOM reference: [position property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS quotes Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify the quotation marks for quotations:

```
q {  
    quotes: "«" "»";  
}
```

[Try it yourself »](#)

Definition and Usage

The quotes property sets the type of quotation marks for quotations.

Default value: *not specified*

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.quotes=""«'»""`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
quotes	11.0	8.0	1.5	5.1	4.0

Note: IE8 supports the quotes property only if a !DOCTYPE is specified.

CSS Syntax

```
quotes: none|string|initial|inherit;
```

Property Values

Value	Description	Play it
none	Specifies that the "open-quote" and "close-quote" values of the "content" property will not produce any quotation marks	Play it »
<i>string string string string</i>	Specifies which quotation marks to use. The first two values specifies the first level of quotation embedding, the next two values specifies the next level of quote embedding, etc	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Quotation Mark Characters

Result	Description	Entity Number
"	double quote	"
'	single quote	'
<	single, left angle quote	‹
>	single, right angle quote	›
«	double, left angle quote	«
»	double, right angle quote	»
'	left quote (single high-6)	‘

'	right quote (single high-9)	’
"	left quote (double high-6)	“
"	right quote (double high-9)	”
"	double quote (double low-9)	„

Related Pages

HTML DOM reference: [quotes property](#)

 [Previous](#)

[Complete CSS Reference](#)

[Next](#) 

CSS3 resize Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify that a <div> element should be resizable by the user:

```
div {
    resize: both;
    overflow: auto;
}
```

[Try it yourself »](#)

Definition and Usage

The resize property specifies whether or not an element is resizable by the user.

Note: The resize property applies to elements whose computed overflow value is something other than "visible".

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.resize="both"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.

Property					
resize	4.0	Not supported	5.0 4.0 -moz-	4.0	15.0

CSS Syntax

```
resize: none|both|horizontal|vertical|initial|inherit;
```

Property Values

Value	Description	Play it
none	Default value. The user cannot resize the element	Play it »
both	The user can adjust both the height and the width of the element	Play it »
horizontal	The user can adjust the width of the element	Play it »
vertical	The user can adjust the height of the element	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS3 tutorial: [CSS3 User Interface](#)

HTML DOM reference: [resize property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

CSS right Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the right edge of the image 5px to the left of the right edge of its containing element:

```
img {  
    position: absolute;  
    right: 5px;  
}
```

[Try it yourself »](#)

Definition and Usage

For absolutely positioned elements, the right property sets the right edge of an element to a unit to the left/right of the right edge of its containing element.

For relatively positioned elements, the right property sets the right edge of an element to a unit to the left/right to its normal position.

Note: If "position:static", the right property has no effect.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.right="200px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
right	1.0	5.5	1.0	1.0	5.0

CSS Syntax

```
right: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Lets the browser calculate the right edge position. This is default	Play it »
length	Sets the right edge position in px, cm, etc. Negative values are allowed	Play it »
%	Sets the right edge position in % of containing element. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

CSS reference: [left property](#)

CSS reference: [bottom property](#)

CSS reference: [top property](#)

HTML DOM reference: [right property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS right property

CSS tab-size Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the tab-size of a <pre> element:

```
pre {-moz-tab-size: 16;} /* Code for Firefox */
pre {-o-tab-size: 16;} /* Code for Opera 10.6-12.1 */
pre {tab-size: 16;}
```

[Try it yourself »](#)

Definition and Usage

The tab-size property specifies the length of the space used for the tab character.

In HTML, the tab character is usually displayed as a single space-character, except for some elements, like <textarea> and <pre>, and the result of the tab-size property will only be visible for these elements.

Default value: 8

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.tabSize="16"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- or -o- specify the first version that worked with a prefix.

Property					
tab-size	21.0	Not supported	4.0 -moz-	6.1	15.0 10.6 -o-

Note: No browser is currently supporting the value as a length unit.

CSS Syntax

```
tab-size: number|Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>number</i>	Default value 8. Specifies the number of space-characters that should be displayed for each tab-character	Play it »
<i>length</i>	Specifies the length of a tab-character. This property value is not supported in any of the major browsers	
<i>initial</i>	Sets this property to its default value. Read about initial	Play it »
<i>inherit</i>	Inherits this property from its parent element. Read about inherit	

[◀ Previous](#)

[Complete CSS Reference](#)

[Next ▶](#)

CSS table-layout Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the table layout algorithm:

```
table {
    table-layout: fixed;
}
```

[Try it yourself »](#)

Definition and Usage

The table-layout property sets the table layout algorithm to be used for a table.

Default value: auto

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.tableLayout= "fixed"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
table-layout	14.0	5.0	1.0	1.0	7.0

CSS Syntax

```
table-layout: auto|fixed|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Automatic table layout algorithm (this is default): <ul style="list-style-type: none"> The column width is set by the widest unbreakable content in the cells Can be slow, since it needs to read through all the content in the table, before determining the final layout 	Play it »
fixed	Fixed table layout algorithm: <ul style="list-style-type: none"> The horizontal layout only depends on the table's width and the width of the columns, not the contents of the cells Allows a browser to lay out the table faster than the automatic table layout The browser can begin to display the table once the first row has been received 	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Table](#)

HTML DOM reference: [tableLayout property](#)

◀ Previous

Complete  Reference

Next ▶

CSS text-align Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the text alignment for `<h1>`, `<h2>`, and `<h3>` elements:

```
h1 {  
    text-align: center;  
}  
  
h2 {  
    text-align: left;  
}  
  
h3 {  
    text-align: right;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The `text-align` property specifies the horizontal alignment of text in an element.

Default value: left if direction is ltr, and right if direction is rtl

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.textAlign="right"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-align	1.0	3.0	1.0	1.0	3.5

CSS Syntax

```
text-align: left|right|center|justify|initial|inherit;
```

Property Values

Value	Description	Play it
left	Aligns the text to the left	Play it »
right	Aligns the text to the right	Play it »
center	Centers the text	Play it »
justify	Stretches the lines so that each line has equal width (like in newspapers and magazines)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

This example demonstrates a more advanced text-align example:

```

h1 {
    text-align: center;
}

p.date {
    text-align: right;
}

```

```
}
```

```
p.main {  
    text-align: justify;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textAlign property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS text-align-last Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Align the last line of a <p> element to the right:

```
p {  
    text-align: justify;  
    -moz-text-align-last: right; /* Code for Firefox */  
    text-align-last: right;  
}
```

[Try it yourself »](#)

Definition and Usage

The text-align-last property specifies how to align the last line of a text.

Note: The text-align-last property will only work for elements with the text-align property set to "justify".

Default value: auto

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.textAlignLast="right"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.

Property					
text-align-last	Not supported	5.5	12.0 -moz-	Not supported	Not supported

Note: In Internet Explorer, the "start" and "end" values are not supported.

CSS Syntax

```
text-align-last: auto|left|right|center|justify|start|end|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Default value. The last line is justified and aligned left	Play it »
left	The last line is aligned to the left	Play it »
right	The last line is aligned to the right	Play it »
center	The last line is center-aligned	Play it »
justify	The last line is justified as the rest of the lines	Play it »
start	The last line is aligned at the beginning of the line (left if the text-direction is left-to-right, and right is the text-direction is right-to-left)	Play it »
end	The last line is aligned at the end of the line (right if the text-direction is left-to-right, and left is the text-direction is right-to-left)	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS text-align-last property

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textAlignLast property](#)

 Previous

Complete  Reference

Next 

CSS text-decoration Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the text decoration for `<h1>`, `<h2>`, and `<h3>` elements:

```
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}
```

[Try it yourself »](#)

Definition and Usage

The `text-decoration` property specifies the decoration added to text.

Note: In CSS3, the `text-decoration` property is a shorthand property for [text-decoration-line](#), [text-decoration-color](#), and [text-decoration-style](#), but this is currently not supported in any of the major browsers.

Note: In CSS3 you can use the [text-decoration-color](#) property to change the color of the decoration, otherwise the color is the same as the color of the text.

Default value: none

Inherited: no

Animatable: no, see *individual properties*. [Read about animatable](#)

Version: CSS1, renewed in CSS3**JavaScript syntax:** `object.style.textDecoration="underline"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-decoration	1.0	3.0	1.0	1.0	3.5

CSS Syntax

```
text-decoration: none|underline|overline|line-through|initial|inherit;
```

Property Values

Value	Description	Play it
none	Defines a normal text. This is default	Play it »
underline	Defines a line below the text	Play it »
overline	Defines a line above the text	Play it »
line-through	Defines a line through the text	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textDecoration property](#)

[Previous](#)

Complete [CSS](#) Reference

[Next](#)

CSS text-decoration-color Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Change the color of the line, in an underlined text:

```
p {
    text-decoration: underline;
    -moz-text-decoration-color: red; /* Code for Firefox */
    text-decoration-color: red;
}
```

[Try it yourself »](#)

Definition and Usage

The text-decoration-color property specifies the color of the text-decoration (underlines, overlines, linethroughs).

Note: The text-decoration-color property will only have an effect on elements with a visible text-decoration.

Default value: *currentColor*

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: *object.style.textDecorationColor="red"*

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.

Property					
text-decoration-color	Not supported	Not supported	6.0 -moz-	Not supported	Not supported

CSS Syntax

```
text-decoration-color: color|initial|inherit;
```

Property Values

Value	Description	Play it
color	Specifies the color of the text-decoration	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textDecorationColor property](#)

[Previous](#)

Complete Reference

[Next](#)

CSS text-decoration-line Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Display <p> elements with a line on top:

```
p {  
    -moz-text-decoration-line: overline; /* Code for Firefox */  
    text-decoration-line: overline;  
}
```

[Try it yourself »](#)

Definition and Usage

The text-decoration-line property specifies what type of line, if any, the decoration will have.

Note: You can also set the text-decoration-line using the text-decoration property, which is a short-hand property for the text-decoration-line, text-decoration-style, and the text-decoration-color properties.

Note: You can also combine more than one value, like underline *and* overline to display lines both under and over the text.

Default value: none

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.textDecorationLine="overline"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.

Property					
text-decoration-line	Not supported	Not supported	6.0 -moz-	Not supported	Not supported

CSS Syntax

```
text-decoration-line: none|underline|overline|line-through|initial|inherit;
```

Property Values

Value	Description	Play it
none	Default value. Specifies no line for the text-decoration	Play it »
underline	Specifies that a line will be displayed under the text	Play it »
overline	Specifies that a line will be displayed over the text	Play it »
line-through	Specifies that a line will be displayed through the text	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textDecorationLine property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS text-decoration-style Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Display a wavy line under a <p> element:

```
p {
    -moz-text-decoration-style: wavy; /* Code for Firefox */
    text-decoration: underline;
}
```

[Try it yourself »](#)

Definition and Usage

The text-decoration-style property specifies how the line, if any, will display.

Default value: solid

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.textDecorationStyle="wavy"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -moz- specify the first version that worked with a prefix.



Property					
text-decoration-style	Not supported	Not supported	6.0 -moz-	Not supported	Not supported

CSS Syntax

```
text-decoration-style: solid|double|dotted|dashed|wavy|initial|inherit;
```

Property Values

Value	Description	Play it
solid	Default value. The line will display as a single line	Play it »
double	The line will display as a double line	Play it »
dotted	The line will display as a dotted line	Play it »
dashed	The line will display as a dashed line	Play it »
wavy	The line will display as a wavy line	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textDecorationStyle property](#)

Previous

Complete Reference

Next

CSS text-indent Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Indent the first line of all `<p>` elements with 50 pixels:

```
p {  
    text-indent: 50px;  
}
```

[Try it yourself »](#)

Definition and Usage

The `text-indent` property specifies the indentation of the first line in a text-block.

Note: Negative values are allowed. The first line will be indented to the left if the value is negative.

Default value: 0

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.textIndent="50px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.



Property					
text-indent	1.0	3.0	1.0	1.0	3.5

CSS Syntax

```
text-indent: Length|initial|inherit;
```

Property Values

Value	Description	Play it
<i>length</i>	Defines a fixed indentation in px, pt, cm, em, etc. Default value is 0	Play it »
%	Defines the indentation in % of the width of the parent element	Play it »
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textIndent property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS3 text-justify Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Justification changes spacing between words:

```
div {  
    text-align: justify;  
    text-justify: inter-word;  
}
```

[Try it yourself »](#)

Definition and Usage

The text-justify property specifies the justification method to use when text-align is set to "justify".

This property specifies how justified text should be aligned and spaced.

Default value: auto

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.textJustify="inter-word"`

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-justify	Not supported	5.5	Not supported	Not supported	Not supported

CSS Syntax

```
text-justify: auto|inter-word|inter-ideograph|inter-cluster|distribute|kashida|trim|initial|inherit;
```

Property Values

Value	Description	Play it
auto	The browser determines the justification algorithm	Play it »
inter-word	Increases/Decreases the space between words	Play it »
inter-ideograph	Justifies content with ideographic text	Play it »
inter-cluster	Only content that does not contain any inter-word spacing (such as Asian languages) is justified	Play it »
distribute	Spacing between letters and words are increased or decreased as necessary	Play it »
kashida	Justifies content by elongating characters	Play it »
trim		Play it »
none	The justification is disabled	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

[◀ Previous](#)
[Complete CSS Reference](#)
[Next ▶](#)

CSS3 text-justify property

CSS3 text-overflow Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Use of the text-overflow property:

```
div {  
    text-overflow: ellipsis;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The text-overflow property specifies what should happen when text overflows the containing element.

Default value: clip

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.textOverflow="ellipsis"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -o- specify the first version that worked with a prefix.

Property					
text-overflow	4.0	6.0	7.0	3.1	11.0 9.0 -o-

CSS Syntax

```
text-overflow: clip|ellipsis|string|initial|inherit;
```

Property Values

Value	Description	Play it
clip	Default value. Clips the text	Play it »
ellipsis	Render an ellipsis ("...") to represent clipped text	Play it »
<i>string</i>	Render the given string to represent clipped text	
initial	Sets this property to its default value. Read about <i>initial</i>	Play it »
inherit	Inherits this property from its parent element. Read about <i>inherit</i>	



More Examples

Text-overflow with a hover effect

This example demonstrates how to display the entire text when hover over the element.

Related Pages

CSS3 tutorial: [CSS3 Text Effects](#)

HTML DOM reference: [textOverflow property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS3 text-shadow Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Basic text-shadow:

```
h1 {  
    text-shadow: 2px 2px #ff0000;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The text-shadow property adds shadow to text.

This property accepts a comma-separated list of shadows to be applied to the text.

Default value: *none*

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: *object.style.textShadow="2px 5px 5px red"*

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-shadow	4.0	10.0	3.5	4.0	9.6

CSS Syntax

```
text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;
```

Note: To add more than one shadow to the text, add a comma-separated list of shadows.

Property Values

Value	Description	Play it
<i>h-shadow</i>	Required. The position of the horizontal shadow. Negative values are allowed	Play it »
<i>v-shadow</i>	Required. The position of the vertical shadow. Negative values are allowed	Play it »
<i>blur-radius</i>	Optional. The blur radius. Default value is 0	Play it »
<i>color</i>	Optional. The color of the shadow. Look at CSS Color Values for a complete list of possible color values	Play it »
<i>none</i>	Default value. No shadow	Play it »
<i>initial</i>	Sets this property to its default value. Read about initial	Play it »
<i>inherit</i>	Inherits this property from its parent element. Read about inherit	



More Examples

Example

This example demonstrates a text-shadow with a blur effect:

```
h1 {
    text-shadow: 2px 2px 8px #FF0000;
}
```

[Try it yourself »](#)

Example

This example demonstrates text-shadow on a white text:

```
h1 {  
    color: white;  
    text-shadow: 2px 2px 4px #000000;  
}
```

[Try it yourself »](#)

Example

This example demonstrates text-shadow with a red neon glow:

```
h1 {  
    text-shadow: 0 0 3px #FF0000;  
}
```

[Try it yourself »](#)

Example

This example demonstrates text-shadow with a red and blue neon glow:

```
h1 {  
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;  
}
```

[Try it yourself »](#)

Related Pages

CSS3 tutorial: [CSS3 Text Effects](#)

HTML DOM reference: [textShadow property](#)

[Previous](#)

Complete CSS Reference

[Next](#)

CSS text-transform Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Transform text in different <p> elements:

```
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}
```

[Try it yourself »](#)

Definition and Usage

The text-transform property controls the capitalization of text.

Default value: none

Inherited: yes

Version: CSS1

JavaScript syntax: `object.style.textTransform="uppercase"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
text-transform	1.0	4.0	1.0	1.0	7.0

CSS Syntax

```
text-transform: none|capitalize|uppercase|lowercase|initial|inherit;
```

Property Values

Value	Description	Play it
none	No capitalization. The text renders as it is. This is default	Play it »
capitalize	Transforms the first character of each word to uppercase	Play it »
uppercase	Transforms all characters to uppercase	Play it »
lowercase	Transforms all characters to lowercase	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [textTransform property](#)

[Previous](#)

[Complete CSS Reference](#)

[Next](#)

CSS top Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the top edge of the image to 15px below the top edge of its containing element:

```
img {  
    position: absolute;  
    top: 15px;  
}
```

[Try it yourself »](#)

Definition and Usage

For absolutely positioned elements, the top property sets the top edge of an element to a unit above/below the top edge of its containing element.

For relatively positioned elements, the top property sets the top edge of an element to a unit above/below its normal position.

Note: If "position:static", the top property has no effect.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.top="100px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
top	1.0	5.0	1.0	1.0	6.0

CSS Syntax

```
top: auto|Length|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Lets the browser calculate the top edge position. This is default	Play it »
length	Sets the top edge position in px, cm, etc. Negative values are allowed	Play it »
%	Sets the top edge position in % of containing element. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

CSS reference: [bottom property](#)

CSS reference: [left property](#)

CSS reference: [right property](#)

HTML DOM reference: [top property](#)

[Previous](#)

Complete [CSS Reference](#)

[Next](#)

CSS top property

CSS3 transform Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Rotate a <div> element:

```
div {
    -ms-transform: rotate(7deg); /* IE 9 */
    -webkit-transform: rotate(7deg); /* Chrome, Safari, Opera */
    transform: rotate(7deg);
}
```

[Try it yourself »](#)

Definition and Usage

The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.

To better understand the transform property, [view a demo](#).

Default value: none

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.transform="rotate(7deg)"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
transform (2D)	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	3.2 -webkit-	23.0 15.0 -webkit- 12.1 10.5 -o-
transform (3D)	36.0 12.0 -webkit-	10.0	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-

Syntax

```
transform: none|transform-functions|initial|inherit;
```

Property Values

Value	Description	Play it
none	Defines that there should be no transformation	Play it »
matrix(<i>n,n,n,n,n,n</i>)	Defines a 2D transformation, using a matrix of six values	Play it »
matrix3d (<i>n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n</i>)	Defines a 3D transformation, using a 4x4 matrix of 16 values	
translate(<i>x,y</i>)	Defines a 2D translation	Play it »
translate3d(<i>x,y,z</i>)	Defines a 3D translation	
translateX(<i>x</i>)	Defines a translation, using only the value for the X-axis	Play it »
translateY(<i>y</i>)	Defines a translation, using only the value for the Y-axis	Play it »
translateZ(<i>z</i>)	Defines a 3D translation, using only the value for the Z-axis	

scale(<i>x,y</i>)	Defines a 2D scale transformation	Play it »
scale3d(<i>x,y,z</i>)	Defines a 3D scale transformation	
scaleX(<i>x</i>)	Defines a scale transformation by giving a value for the X-axis	Play it »
scaleY(<i>y</i>)	Defines a scale transformation by giving a value for the Y-axis	Play it »
scaleZ(<i>z</i>)	Defines a 3D scale transformation by giving a value for the Z-axis	
rotate(<i>angle</i>)	Defines a 2D rotation, the angle is specified in the parameter	Play it »
rotate3d(<i>x,y,z,angle</i>)	Defines a 3D rotation	
rotateX(<i>angle</i>)	Defines a 3D rotation along the X-axis	Play it »
rotateY(<i>angle</i>)	Defines a 3D rotation along the Y-axis	Play it »
rotateZ(<i>angle</i>)	Defines a 3D rotation along the Z-axis	Play it »
skew(<i>x-angle,y-angle</i>)	Defines a 2D skew transformation along the X- and the Y-axis	Play it »
skewX(<i>angle</i>)	Defines a 2D skew transformation along the X-axis	Play it »
skewY(<i>angle</i>)	Defines a 2D skew transformation along the Y-axis	Play it »
perspective(<i>n</i>)	Defines a perspective view for a 3D transformed element	
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

[Images thrown on the table](#)

This example demonstrates how to create "polaroid" pictures and rotate the pictures.

Related Pages

CSS tutorial: [CSS 2D Transforms](#)

CSS tutorial: [CSS 3D Transforms](#)

HTML DOM reference: [transform property](#)

 Previous

Complete  Reference

Next 

CSS3 transform-origin Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set a rotated element's base placement:

```
div {
    -ms-transform: rotate(45deg); /* IE 9 */
    -ms-transform-origin: 20% 40%; /* IE 9 */
    -webkit-transform: rotate(45deg); /* Chrome, Safari, Opera */
    -webkit-transform-origin: 20% 40%; /* Chrome, Safari, Opera */
    transform: rotate(45deg);
    transform-origin: 20% 40%;
}
```

[Try it yourself »](#)

Definition and Usage

The transform-origin property allows you to change the position of transformed elements.

2D transformations can change the x- and y-axis of an element. 3D transformations can also change the z-axis of an element.

To better understand the transform-origin property, [view a demo](#).

Note: This property must be used together with the [transform](#) property.

Tip: To better understand this property for **3D transforms**, [view a demo](#).

Default value: 50% 50% 0

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS3

JavaScript syntax: `object.style.transformOrigin="0 0"`**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz-, or -o- specify the first version that worked with a prefix.

Property					
transform-origin (two-value syntax)	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	3.2 -webkit-	23.0 15.0 - webkit- 12.1 10.5 -o-
transform-origin (three-value syntax)	36.0 12.0 - webkit-	10.0	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 - webkit-

CSS Syntax

```
transform-origin: x-axis y-axis z-axis|initial|inherit;
```

Property Values

Property Value	Description
<i>x-axis</i>	Defines where the view is placed at the x-axis. Possible values: <ul style="list-style-type: none"> • left • center • right • <i>length</i> • %
<i>y-axis</i>	Defines where the view is placed at the y-axis. Possible values: <ul style="list-style-type: none"> • top • center • bottom • <i>length</i> • %

<i>z-axis</i>	Defines where the view is placed at the z-axis (for 3D transformations). Possible values:
	<ul style="list-style-type: none">• <i>length</i>
initial	Sets this property to its default value. Read about <i>initial</i>
inherit	Inherits this property from its parent element. Read about <i>inherit</i>

Related Pages

CSS tutorial: [CSS 2D Transforms](#)

CSS tutorial: [CSS 3D Transforms](#)

HTML DOM reference: [transformOrigin property](#)

 Previous

Complete CSS Reference

Next 

CSS3 transform-style Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Let the transformed child elements preserve the 3D transformations:

```
div {
    -webkit-transform: rotateY(60deg); /* Chrome, Safari, Opera */
    -webkit-transform-style: preserve-3d; /* Chrome, Safari, Opera */
    transform: rotateY(60deg);
    transform-style: preserve-3d;
}
```

[Try it yourself »](#)

Definition and Usage

The transform-style property specifies how nested elements are rendered in 3D space.

Note: This property must be used together with the [transform](#) property.

To better understand the transform-style property, [view a demo](#).

Default value: flat

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.transformStyle="preserve-3d"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property					
transform-style	36.0 12.0 -webkit-	11.0	16.0 10.0 -moz-	4.0 -webkit-	23.0 15.0 -webkit-

CSS Syntax

```
transform-style: flat|preserve-3d|initial|inherit;
```

Property Values

Property Value	Description
flat	Specifies that child elements will NOT preserve its 3D position. This is default
preserve-3d	Specifies that child elements will preserve its 3D position
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS 2D Transforms](#)

CSS tutorial: [CSS 3D Transforms](#)

HTML DOM reference: [transformStyle property](#)

Previous

Complete Reference

Next

CSS3 transform-style property

CSS3 transition Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Hover over a <div> element to gradually change the width from 100px to 300px:

```
div {
    width: 100px;
    -webkit-transition: width 2s; /* Safari 3.1 to 6.0 */
    transition: width 2s;
}

div:hover {
    width: 300px;
}
```

[Try it yourself »](#)

Definition and Usage

The transition property is a shorthand property for the four transition properties:

transition-property, transition-duration, transition-timing-function, and transition-delay.

Note: Always specify the transition-duration property, otherwise the duration is 0, and the transition will have no effect.

Default value: all 0 ease 0

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.transition="all 2s"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Property					
transition	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

CSS Syntax

```
transition: property duration timing-function delay|initial|inherit;
```

Property Values

Value	Description
<u>transition-property</u>	Specifies the name of the CSS property the transition effect is for
<u>transition-duration</u>	Specifies how many seconds or milliseconds the transition effect takes to complete
<u>transition-timing-function</u>	Specifies the speed curve of the transition effect
<u>transition-delay</u>	Defines when the transition effect will start
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS3 Transitions](#)

HTML DOM reference: [transition property](#)

[Previous](#)

Complete **CSS** Reference

Next [»](#)

CSS3 transition-delay Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Wait 2 seconds before the transition effect starts:

```
div {
    -webkit-transition-delay: 2s; /* Safari */
    transition-delay: 2s;
}
```

[Try it yourself »](#)

Definition and Usage

The transition-delay property specifies when the transition effect will start.

The transition-delay value is defined in seconds (s) or milliseconds (ms).

Default value: 0

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.transitionDelay="2s"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Property					
transition-delay	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

CSS Syntax

```
transition-delay: time|initial|inherit;
```

Property Values

Value	Description
<i>time</i>	Specifies the number of seconds or milliseconds to wait before the transition effect will start
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS3 Transitions](#)

HTML DOM reference: [transitionDelay property](#)

◀ Previous

Complete Reference

Next ▶



CSS3 transition-duration Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Let the transition effect last for 5 seconds:

```
div {  
    -webkit-transition-duration: 5s; /* Safari */  
    transition-duration: 5s;  
}
```

[Try it yourself »](#)

Definition and Usage

The transition-duration property specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.

Default value: 0

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.transitionDuration="1s"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Property					
transition-duration	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

CSS Syntax

```
transition-duration: time|initial|inherit;
```

Property Values

Value	Description
time	Specifies how many seconds or milliseconds a transition effect takes to complete. Default value is 0, meaning there will be no effect
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS tutorial: [CSS3 Transitions](#)

HTML DOM reference: [transitionDuration property](#)

Previous

Complete CSS Reference

Next

WEB HOSTING

UK Reseller Hosting

WEB BUILDING
FREE Website BUILDER
Free HTML5 Templates

W3SCHOOLS EXAMS

HTML, CSS, JavaScript, PHP, jQuery, and XML Certifications

SHARE THIS PAGE

COLOR PICKER



[REPORT ERROR](#)
[PRINT PAGE](#)
[FORUM](#)
[ABOUT](#)

Top 10 Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [SQL Tutorial](#)
- [PHP Tutorial](#)
- [jQuery Tutorial](#)
- [Bootstrap Tutorial](#)
- [Angular Tutorial](#)
- [ASP.NET Tutorial](#)
- [XML Tutorial](#)

Top 10 References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)
- [Browser Statistics](#)
- [HTML DOM](#)
- [PHP Reference](#)
- [jQuery Reference](#)
- [HTML Colors](#)
- [HTML Character Sets](#)
- [XML DOM](#)

Top 10 Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [HTML DOM Examples](#)
- [PHP Examples](#)
- [jQuery Examples](#)
- [XML Examples](#)
- [XML DOM Examples](#)
- [ASP Examples](#)
- [SVG Examples](#)

Web Certificates

- [HTML Certificate](#)
- [HTML5 Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [jQuery Certificate](#)
- [PHP Certificate](#)
- [XML Certificate](#)

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 1999-2015 by Refsnes Data. All Rights Reserved.



CSS3 transition-property Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Hover over a <div> element, and change the width with a smooth transition effect:

```
div {
    -webkit-transition-property: width; /* Safari */
    transition-property: width;
}

div:hover {
    width: 300px;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The transition-property property specifies the name of the CSS property the transition effect is for (the transition effect will start when the specified CSS property changes).

Tip: A transition effect could typically occur when a user hover over an element.

Note: Always specify the transition-duration property, otherwise the duration is 0, and the transition will have no effect.

Default value: all

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: object.style.transitionProperty="width,height"**Try it**

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Property					
transition-property	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

CSS Syntax

```
transition-property: none|all|property|initial|inherit;
```

Property Values

Value	Description
none	No property will get a transition effect
all	Default value. All properties will get a transition effect
<i>property</i>	Defines a comma separated list of CSS property names the transition effect is for
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit



More Examples

Example

Hover over a <div> element, and change the width AND height with a smooth transition effect:

```
div {
  -webkit-transition-property: width, height; /* Safari */
  transition-property: width, height;
}
```

```
div:hover {  
    width: 300px;  
    height: 300px;  
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS3 Transitions](#)

HTML DOM reference: [transitionProperty property](#)

[◀ Previous](#)

Complete CSS Reference

[Next ▶](#)

CSS3 transition-timing-function Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

A transition effect with the same speed from start to end:

```
div {
    -webkit-transition-timing-function: linear; /* Safari and Chrome */
    transition-timing-function: linear;
}
```

[Try it yourself »](#)

Definition and Usage

The transition-timing-function property specifies the speed curve of the transition effect.

This property allows a transition effect to change speed over its duration.

Default value: ease

Inherited: no

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.transitionTimingFunction="linear"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by -webkit-, -moz- or -o- specify the first version that worked with a prefix.

Property					
transition-timing-function	26.0 4.0 - webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 - webkit-	12.1 10.5 -o-

CSS Syntax

```
transition-timing-function: ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier()|initial|inherit;
```

Property Values

Value	Description
ease	Default value. Specifies a transition effect with a slow start, then fast, then end slowly (equivalent to cubic-bezier(0.25,0.1,0.25,1))
linear	Specifies a transition effect with the same speed from start to end (equivalent to cubic-bezier(0,0,1,1))
ease-in	Specifies a transition effect with a slow start (equivalent to cubic-bezier(0.42,0,1,1))
ease-out	Specifies a transition effect with a slow end (equivalent to cubic-bezier(0,0,0.58,1))
ease-in-out	Specifies a transition effect with a slow start and end (equivalent to cubic-bezier(0.42,0,0.58,1))
cubic-bezier(<i>n,n,n,n</i>)	Define your own values in the cubic-bezier function. Possible values are numeric values from 0 to 1
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Tip: Try the different values in the examples below to understand how it works!



More Examples

Example

To better understand the different function values: Here are five different div elements with five different values:

```
/* For Safari 3.1 to 6.0 */
#div1 {-webkit-transition-timing-function: linear;}
#div2 {-webkit-transition-timing-function: ease;}
#div3 {-webkit-transition-timing-function: ease-in;}
#div4 {-webkit-transition-timing-function: ease-out;}
#div5 {-webkit-transition-timing-function: ease-in-out;}

/* Standard syntax */
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

[Try it yourself »](#)

Example

Same as the example above, but the speed curves are specified with the cubic-bezier function:

```
/* For Safari 3.1 to 6.0 */
#div1 {-webkit-transition-timing-function: cubic-bezier(0,0,1,1);}
#div2 {-webkit-transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {-webkit-transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {-webkit-transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {-webkit-transition-timing-function: cubic-bezier(0.42,0,0.58,1);}

/* Standard syntax */
#div1 {transition-timing-function: cubic-bezier(0,0,1,1);}
#div2 {transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {transition-timing-function: cubic-bezier(0.42,0,1,1);}
#div4 {transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS3 Transitions](#)

HTML DOM reference: [transitionTimingFunction property](#)

[Previous](#)

Complete [CSS](#) Reference

Next [»](#)

CSS unicode-bidi Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Override text:

```
div {
    direction: rtl;
    unicode-bidi: bidi-override;
}
```

[Try it yourself »](#)

Definition and Usage

The `unicode-bidi` property is used together with the `direction` property to set or return whether the text should be overridden to support multiple languages in the same document.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.unicodeBidi="bidi-override"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera	Microsoft Edge
Property	1	1	1	1	1	1



	unicode-bidi	2.0		5.5		1.0		1.3		9.2
--	--------------	-----	--	-----	--	-----	--	-----	--	-----

CSS Syntax

```
unicode-bidi: normal|embed|bidi-override|initial|inherit;
```

Property Values

Value	Description	Play it
normal	Does not use an additional level of embedding. This is default	Play it »
embed	Creates an additional level of embedding	Play it »
bidi-override	Creates an additional level of embedding. Reordering depends on the direction property	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [unicodeBidi property](#)

Previous

Complete Reference

Next

CSS vertical-align Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Vertical align an image:

```
img {
    vertical-align: top;
}
```

[Try it yourself »](#)

Definition and Usage

The vertical-align property sets the vertical alignment of an element.

Default value: baseline

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.verticalAlign="top"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
vertical-align	1.0	4.0	1.0	1.0	4.0

CSS Syntax

```
vertical-align: baseline|Length|sub|super|top|text-top|middle|bottom|text-bottom|initial|inherit;
```

Property Values

Value	Description	Play it
baseline	Align the baseline of the element with the baseline of the parent element. This is default	Play it »
length	Raises or lower an element by the specified length. Negative values are allowed	Play it »
%	Raises or lower an element in a percent of the "line-height" property. Negative values are allowed	Play it »
sub	Aligns the element as if it was subscript	Play it »
super	Aligns the element as if it was superscript	Play it »
top	The top of the element is aligned with the top of the tallest element on the line	Play it »
text-top	The top of the element is aligned with the top of the parent element's font	Play it »
middle	The element is placed in the middle of the parent element	Play it »
bottom	The bottom of the element is aligned with the lowest element on the line	Play it »
text-bottom	The bottom of the element is aligned with the bottom of the parent element's font	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [verticalAlign property](#)

 Previous

Complete  Reference

Next 

CSS visibility Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Make <h2> elements invisible:

```
h2 {  
    visibility: hidden;  
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The visibility property specifies whether or not an element is visible.

Tip: Even invisible elements take up space on the page. Use the display property to create invisible elements that do not take up space!

Default value: visible

Inherited: yes

Animatable: yes. [Read about animatable](#)

Version: CSS2

JavaScript syntax: `object.style.visibility="hidden"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
visibility	1.0	4.0	1.0	1.0	4.0

CSS Syntax

```
visibility: visible|hidden|collapse|initial|inherit;
```

Property Values

Value	Description	Play it
visible	Default value. The element is visible	Play it »
hidden	The element is invisible (but still takes up space)	Play it »
collapse	Only for table elements. collapse removes a row or column, but it does not affect the table layout. The space taken up by the row or column will be available for other content. If collapse is used on other elements, it renders as "hidden"	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

This example demonstrates how to make a table element collapse:

```
tr.collapse {
    visibility: collapse;
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Display and visibility](#)

HTML DOM reference: [visibility property](#)

 Previous

Complete  Reference

Next 

CSS white-space Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify that the text in <p> elements will never wrap:

```
p {  
    white-space: nowrap;  
}
```

[Try it yourself »](#)

Definition and Usage

The white-space property specifies how white-space inside an element is handled.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS1

JavaScript syntax: `object.style.whiteSpace = "nowrap"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
white-space	1.0	8.0	3.5	3.0	9.5

CSS Syntax

```
white-space: normal|nowrap|pre|pre-line|pre-wrap|initial|inherit;
```

Property Values

Value	Description	Play it
normal	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary. This is default	Play it »
nowrap	Sequences of whitespace will collapse into a single whitespace. Text will never wrap to the next line. The text continues on the same line until a tag is encountered	Play it »
pre	Whitespace is preserved by the browser. Text will only wrap on line breaks. Acts like the <pre> tag in HTML	Play it »
pre-line	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary, and on line breaks	Play it »
pre-wrap	Whitespace is preserved by the browser. Text will wrap when necessary, and on line breaks	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [whiteSpace property](#)

 Previous

Complete CSS Reference

Next 

CSS width Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Set the height and width of a <p> element:

```
p.ex {
    height: 100px;
    width: 100px;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

Definition and Usage

The width property sets the width of an element.

Note: The width property does not include padding, borders, or margins; it sets the width of the area inside the padding, border, and margin of the element!

Note: The min-width and max-width properties override width.

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.width="500px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
width	1.0	4.0	1.0	1.0	3.5

CSS Syntax

```
width: auto|value|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Default value. The browser calculates the width	Play it »
<i>length</i>	Defines the width in px, cm, etc.	Play it »
%	Defines the width in percent of the containing block	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	



More Examples

Example

Set the width of an element using a percent value:

```
img {
  width: 50%;
}
```

[Try it yourself »](#)

Related Pages

CSS tutorial: [CSS Dimension](#)

CSS tutorial: [CSS Box model](#)

CSS reference: [height property](#)

HTML DOM reference: [width property](#)

 Previous

Complete  Reference

Next 

CSS3 word-break Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Break words between any two letters:

```
p.test {
    word-break: break-all;
}
```

[Try it yourself »](#)

Definition and Usage

The word-break property specifies line breaking rules for non-CJK scripts.

Tip: CJK scripts are Chinese, Japanese and Korean ("CJK") scripts.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.wordBreak = "break-all"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property	16	10	3.5	5.1	10.5



word-break	4.0	5.5	15.0	3.1	15.0
------------	-----	-----	------	-----	------

CSS Syntax

```
word-break: normal|break-all|keep-all|initial|inherit;
```

Property Values

Value	Description
normal	Default value. Break words according to their usual rules
break-all	Lines may break between any two letters
keep-all	Breaks are prohibited between pairs of letters
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

[◀ Previous](#)[Complete CSS Reference](#)[Next ▶](#)

CSS word-spacing Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Specify that the space between words in <p> elements should be 30 pixels:

```
p {  
    word-spacing: 30px;  
}
```

[Try it yourself »](#)

Definition and Usage

The word-spacing property increases or decreases the white space between words.

Note: Negative values are allowed.

Default value: normal

Inherited: yes

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS1

JavaScript syntax: `object.style.wordSpacing="20px"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property	Chrome	Internet Explorer	Firefox	Safari	Opera
Property					



word-spacing	1.0	6.0	1.0	1.0	3.5
--------------	-----	-----	-----	-----	-----

CSS Syntax

```
word-spacing: normal|Length|initial|inherit;
```

Property Values

Value	Description	Play it
normal	Defines normal space between words . This is default	Play it »
length	Defines an extra space between words in px, pt, cm, em, etc. Negative values are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Text](#)

HTML DOM reference: [wordSpacing property](#)

[◀ Previous](#)

Complete  Reference

[Next ▶](#)

CSS3 word-wrap Property

[Previous](#)
[Complete CSS Reference](#)
[Next](#)

Example

Allow long words to be able to break and wrap onto the next line:

```
p.test {
    word-wrap: break-word;
}
```

[Try it yourself »](#)

Definition and Usage

The word-wrap property allows long words to be able to be broken and wrap onto the next line.

Default value: normal

Inherited: yes

Animatable: no. [Read about animatable](#)

Version: CSS3

JavaScript syntax: `object.style.wordWrap = "break-word"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
word-wrap	4.0	5.5	3.5	3.1	10.5

CSS Syntax

```
word-wrap: normal|break-word|initial|inherit;
```

Property Values

Value	Description
normal	Break words only at allowed break points
break-word	Allows unbreakable words to be broken
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

Related Pages

CSS3 tutorial: [CSS3 Text Effects](#)

◀ Previous

Complete CSS Reference

Next ▶

CSS z-index Property

[Previous](#)[Complete CSS Reference](#)[Next](#)

Example

Set the z-index for an image:

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

[Try it yourself »](#)

Definition and Usage

The z-index property specifies the stack order of an element.

An element with greater stack order is always in front of an element with a lower stack order.

Note: z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).

Default value: auto

Inherited: no

Animatable: yes. [Read about animatable](#)

[Try it](#)

Version: CSS2

JavaScript syntax: `object.style.zIndex="-1"`

[Try it](#)

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Property					
z-index	1.0	4.0	3.0	1.0	4.0

CSS Syntax

```
z-index: auto|number|initial|inherit;
```

Property Values

Value	Description	Play it
auto	Sets the stack order equal to its parents. This is default	Play it »
number	Sets the stack order of the element. Negative numbers are allowed	Play it »
initial	Sets this property to its default value. Read about initial	Play it »
inherit	Inherits this property from its parent element. Read about inherit	

Related Pages

CSS tutorial: [CSS Positioning](#)

HTML DOM reference: [zIndex property](#)

Previous

Complete CSS Reference

Next