## What is CSS?

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work
* External Style Sheets are stored in **CSS files**

# CSS Syntax

## CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

## CSS Example

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

## Example

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

Hello World!

This paragraph is styled with CSS.

### 

### The Syntax

A rule in CSS has two parts: the **selector** and the **declaration**:

There are three types of elements that a CSS selector can be applied to:

* **HTML** Elements, such as <P> tags, <DIV> tags etc.
* Elements with a specific **CLASS**, for example an element with the class "content" (<P CLASS="content">)
* Elements with a specific **ID**, for example an element with the ID "warning1" (<P ID="warning1">)

## 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 begins with "/\*", and ends with "\*/", like this:

/\*This is a comment\*/  
p  
{  
text-align:center;  
/\*This is another comment\*/  
color:black;  
font-family:arial;  
}

### <DIV> and <SPAN>

<div> is short for division [within a document], that means it is a container, a block level element that has a wide variety of uses.

<div> (and <span>) have no inherent properties of their own, and that makes them so well suited for applying classes to block level elements (the <div>) or inline elements (the <span>), even if there is no existing HTML tag to apply the classes to.

You can use the <div> to position and style whole selections and these sections can contain other elements.

## The id Selector

The id selector is used to specify a style for a single, unique element.

The id selector uses the id attribute of the HTML element, and is defined with a "#".

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

## Example

<html>

<head>

<style type="text/css">

#para1

{

text-align:center;

color:red;

}

</style>

</head>

<body>

<p id="para1">Hello World!</p>

<p>This paragraph is not affected by the style.</p>

</body>

</htm }

**Reasult**

Hello World!

This paragraph is not affected by the style.

## The class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

This allows you to set a particular style for many HTML elements with the same class.

The class selector uses the HTML class attribute, and is defined with a "."

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

## Example

<html>

<head>

<style type="text/css">

.center

{

text-align:center;

}

</style>

</head>

<body>

<h1 class="center">Center-aligned heading</h1>

<p class="center">Center-aligned paragraph.</p>

</body>

</html>

**Result:**

# Center-aligned heading

Center-aligned paragraph.

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;}

<body>

<h1 class="center">This heading will not be affected</h1>

<p class="center">This paragraph will be center-aligned.</p>

</body>

Reasult

# This heading will not be affected

This paragraph will be center-aligned.

## 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 one file. Each page must link to the style sheet using 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. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}

 Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin-left:20px") will work in IE, but not in Firefox or Opera.

## 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, by using the <style> tag, like this:

<head>  
<style type="text/css">  
hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}  
</style>  
</head>

## Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

To use inline styles you use the style attribute in 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 paragraph:

<p style="color:sienna;margin-left:20px">This is a paragraph.</p>

## 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, an external style sheet has these properties for the h3 selector:

h3  
{  
color:red;  
text-align:left;  
font-size:8pt;  
}

And an internal style sheet has these properties for the h3 selector:

h3  
{  
text-align:right;  
font-size:20pt;  
}

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

color:red;  
text-align:right;  
font-size:20pt;

The color is inherited from the external style sheet and the text-alignment and the font-size 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)

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).

CSS Background

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 defined in the body selector:

## Example

body {background-color:#b0c4de;}

<html>

<head>

<style type="text/css">

body

{

background-color:#b0c4de;

}

</style>

</head>

<body>

<h1>My CSS web page!</h1>

<p>Hello world! This is a W3Schools.com example.</p>

</body>

</html>

**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"**

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;}

## 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');}

## 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('gradient2.png');  
}

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

## Example

body  
{  
background-image:url('gradient2.png');  
background-repeat:repeat-x;  
}

## Background Image - Set position and no-repeat

 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;  
}

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;  
}

## 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;}

[How to set a fixed background image](http://www.w3schools.com/css/tryit.asp?filename=trycss_background-attachment)  
This example demonstrates how to set a fixed background image. The image will not scroll with the rest of the page.

<html>

<head>

<style type="text/css">

body

{

background-image:url('smiley.gif');

background-repeat:no-repeat;

background-attachment:fixed;

}

</style>

</head>

<body>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

</body>

</html>

CSS Text

## 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](http://www.w3schools.com/cssref/css_colors_legal.asp) 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:#00ff00;}  
h2 {color:rgb(255,0,0);}

## 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;}

## 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;}

It can also be used to decorate text:

## Example

h1 {text-decoration:overline;}  
h2 {text-decoration:line-through;}  
h3 {text-decoration:underline;}  
h4 {text-decoration:blink;}

## 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;}

## Text Indentation

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

## Example

p {text-indent:50px;}

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

<html>

<head>

<style type="text/css">

h1 {letter-spacing:2px;}

h2 {letter-spacing:-3px;}

</style>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

</body>

</html>

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

<html>

<head>

<style type="text/css">

p.small {line-height:70%;}

p.big {line-height:200%;}

</style>

</head>

<body>

<p>

This is a paragraph with a standard line-height.<br />

This is a paragraph with a standard line-height.<br />

The default line height in most browsers is about 110% to 120%.<br />

</p>

<p class="small">

This is a paragraph with a smaller line-height.<br />

This is a paragraph with a smaller line-height.<br />

</p>

<p class="big">

This is a paragraph with a bigger line-height.<br />

This is a paragraph with a bigger line-height.<br />

</p>

</body>

</html>

[Set the text direction of an element](http://www.w3schools.com/css/tryit.asp?filename=trycss_text_direction)  
This example demonstrates how to change the text direction of an element.

<html>

<head>

<style type="text/css">

div.ex1 {direction:rtl;}

</style>

</head>

<body>

<div>Some text. Default writing direction.</div>

<div class="ex1">Some text. Right-to-left direction.</div>

</body>

</html>

[Increase the white space between words](http://www.w3schools.com/css/tryit.asp?filename=trycss_text_word-spacing)  
This example demonstrates how to increase the white space between words in a paragraph.

<html>

<head>

<style type="text/css">

p

{

word-spacing:30px;

}

</style>

</head>

<body>

<p>

This is some text. This is some text.

</p>

</body>

</html>

[Disable text wrapping inside an element](http://www.w3schools.com/css/tryit.asp?filename=trycss_text_white-space)  
This example demonstrates how to disable text wrapping inside an element.

<html>

<head>

<style type="text/css">

p

{

white-space:nowrap;

}

</style>

</head>

<body>

<p>

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

</p>

</body>

</html>

[Vertical alignment of an image](http://www.w3schools.com/css/tryit.asp?filename=trycss_vertical-align)  
This example demonstrates how to set the vertical align of an image in a text.

<html>

<head>

<style type="text/css">

img.top {vertical-align:text-top;}

img.bottom {vertical-align:text-bottom;}

</style>

</head>

<body>

<p>An <img src="img.gif" alt="Pro Creations " width="270" height="50" /> image with a default alignment.</p>

<p>An <img class="top" src=" img.gif " alt="Pro Creations" width="270" height="50" /> image with a text-top alignment.</p>

<p>An <img class="bottom" src=" img.gif " alt="Pro Creations" width="270" height="50" /> image with a text-bottom alignment.</p>

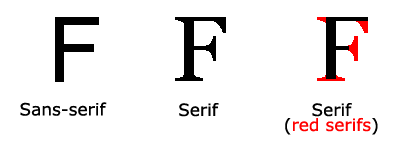
</body>

</html>

CSS Font

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 Lucida Console | All monospace characters have the same width |

## 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 font-family: "Times New Roman".

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

<html><head>

<style type="text/css">

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

p.sansserif{font-family:Arial,Helvetica,sans-serif;}

</style></head>

<body>

<h1>CSS font-family</h1>

<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>

<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

</body></html>

## 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)

<html>

<head>

<style type="text/css">

p.normal {font-style:normal;}

p.italic {font-style:italic;}

p.oblique {font-style:oblique;}

</style>

</head>

<body>

<p class="normal">This is a paragraph, normal.</p>

<p class="italic">This is a paragraph, italic.</p>

<p class="oblique">This is a paragraph, oblique.</p>

</body>

</html>

This is a paragraph, normal.

*This is a paragraph, italic.*

*This is a paragraph, oblique.*

## 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

<html>

<head>

<style>

h1 {font-size:40px;}

h2 {font-size:30px;}

p {font-size:14px;}

</style>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

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

<p>Specifying the font-size in px allows allows Internet Explorer 9, Firefox, Chrome, Opera, and Safari to resize the text.</p>

<p><b>Note:</b> This example does not work in IE, prior version 9.</p></body></html>

## Set Font Size With Em

To avoid the resizing problem with older versions of Internet Explorer, 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*

<html>

<head>

<style>

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 \*/

</style>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

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

<p>Specifying the font-size in em allows all major browsers to resize the text.

Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.

</p>

</body>

</html>

## 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:

<html>

<head>

<style>

body {font-size:100%;}

h1 {font-size:2.5em;}

h2 {font-size:1.875em;}

p {font-size:0.875em;}

</style>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

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

<p>Specifying the font-size in percent and em displays the same size in all

major browsers, and allows all browsers to resize the text!</p>

</body>

</html>

[Set the boldness of the font](http://www.w3schools.com/css/tryit.asp?filename=trycss_font-weight)  
This example demonstrates how to set the boldness of a font.

<html>

<head>

<style type="text/css">

p.normal {font-weight:normal;}

p.light {font-weight:lighter;}

p.thick {font-weight:bold;}

p.thicker {font-weight:900;}

</style>

</head>

<body>

<p class="normal">This is a paragraph.</p>

<p class="light">This is a paragraph.</p>

<p class="thick">This is a paragraph.</p>

<p class="thicker">This is a paragraph.</p>

</body>

</html>

[Set the variant of the font](http://www.w3schools.com/css/tryit.asp?filename=trycss_font-variant)  
This example demonstrates how to set the variant of a font.

<html>

<head>

<style type="text/css">

p.normal {font-variant:normal;}

p.small {font-variant:small-caps;}

</style>

</head>

<body>

<p class="normal">My name is Hege Refsnes.</p>

<p class="small">My name is Hege Refsnes.</p>

</body>

</html>

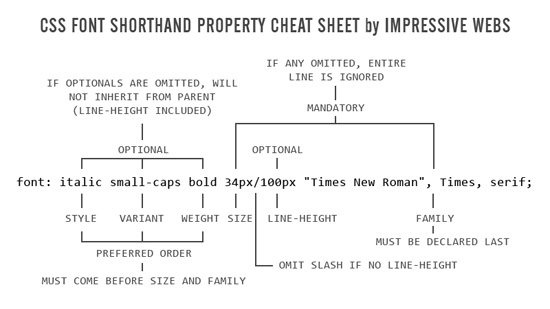
Result :

My name is Hege Refsnes.

My name is Hege Refsnes.

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.

15px/20px = font-size/line-height



<html>

<head>

<style type="text/css">

p.ex1

{

font:15px arial,sans-serif;

}

p.ex2

{

font:italic bold 12px/30px Georgia,serif;

}

</style>

</head>

<body>

<p class="ex1">This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.</p>

<p class="ex2">This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.</p>

</body>

</html>

CSS Links

## Styling Links

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

Special for links are that they 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

<html>

<head>

<style type="text/css">

a:link {color:#FF0000;} /\* unvisited link \*/

a:visited {color:#00FF00;} /\* visited link \*/

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

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

</style>

</head>

<body>

<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>

<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS

definition in order to be effective.</p>

<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order

to be effective.</p>

</body>

</html>

## Text Decoration

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

<html>

<head>

<style type="text/css">

a:link {text-decoration:none;} /\* unvisited link \*/

a:visited {text-decoration:none;} /\* visited link \*/

a:hover {text-decoration:underline;} /\* mouse over link \*/

a:active {text-decoration:underline;} /\* selected link \*/

</style>

</head>

<body>

<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>

<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS

definition in order to be effective.</p>

<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order

to be effective.</p></body></html>

## 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;}

Create link boxes

<html><head>

<style type="text/css">

a:link,a:visited

{

display:block;

font-weight:bold;

color:#FFFFFF;

background-color:#98bf21;

width:120px;

text-align:center;

padding:4px;

text-decoration:none;

}

a:hover,a:active

{

background-color:#7A991A;

}

</style>

</head><body>

<a href="default.asp" target="\_blank">This is a link</a>

</body></html>

CSS Lists

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

<html>

<head>

<style type="text/css">

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;}

</style>

</head>

<body>

<p>Example of unordered lists:</p>

<ul class="a">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>

<ul class="b">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>

<p>Example of ordered lists:</p>

<ol class="c">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ol>

<ol class="d">

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ol>

</body></html>

## An Image as The List Item Marker

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

<html>

<head>

<style type="text/css">

ul

{

list-style-image:url('sqpurple.gif');

}

</style>

</head>

<body>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>

</body>

</html>

## Crossbrowser Solution

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

<html>

<head>

<style type="text/css">

ul

{

list-style-type:none;

padding:0px;

margin:0px;

}

li

{

background-image:url(sqpurple.gif);

background-repeat:no-repeat;

background-position:0px 5px;

padding-left:14px;

}

</style>

</head>

<body>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li></ul></body></html>

## 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:

<html>

<head>

<style type="text/css">

table,th,td

{

border:1px solid black;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

</tr>

</table>

</body>

</html>

## Collapse Borders

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

table

{

border-collapse:collapse;

}

table, td, th

{

border:1px solid black;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

</tr>

</table>

<p><b>Note:</b> If a !DOCTYPE is not specified, the border-collapse property can produce unexpected results

in IE8 and earlier versions.</p>

</body>

</html>

## Table Width and Height

<html>

<head>

<style type="text/css">

table,td,th

{

border:1px solid black;

}

table

{

width:100%;

}

th

{

height:50px;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

## Table Text Alignment

The text in a table is aligned with the text-align and vertical-align properties.

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

<html>

<head>

<style type="text/css">

table,td,th

{

border:1px solid black;

}

td

{

text-align:right;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

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

<html>

<head>

<style type="text/css">

table, td, th

{

border:1px solid black;

}

td

{

height:50px;

vertical-align:bottom;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

## Table Padding

<html>

<head>

<style type="text/css">

table, td, th

{

border:1px solid black;

}

td

{

padding:15px;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

## Table Color

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

<html>

<head>

<style type="text/css">

table, td, th

{

border:1px solid green;

}

th

{

background-color:green;

color:white;

}

</style>

</head>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

Result:

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Savings** |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $300 |
| Cleveland | Brown | $250 |

Make a fancy table

<html>

<head>

<style type="text/css">

#customers

{

font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;

width:100%;

border-collapse:collapse;

}

#customers td, #customers th

{

font-size:1em;

border:1px solid #98bf21;

padding:3px 7px 2px 7px;

}

#customers th

{

font-size:1.1em;

text-align:left;

padding-top:5px;

padding-bottom:4px;

background-color:#A7C942;

color:#ffffff;

}

#customers tr.alt td

{

color:#000000;

background-color:#EAF2D3;

}

</style>

</head>

<body>

<table id="customers">

<tr>

<th>Company</th>

<th>Contact</th>

<th>Country</th>

</tr>

<tr>

<td>Alfreds Futterkiste</td>

<td>Maria Anders</td>

<td>Germany</td>

</tr>

<tr class="alt">

<td>Berglunds snabbköp</td>

<td>Christina Berglund</td>

<td>Sweden</td>

</tr>

<tr>

<td>Centro comercial Moctezuma</td>

<td>Francisco Chang</td>

<td>Mexico</td>

</tr>

<tr class="alt">

<td>Ernst Handel</td>

<td>Roland Mendel</td>

<td>Austria</td>

</tr>

<tr>

<td>Island Trading</td>

<td>Helen Bennett</td>

<td>UK</td>

</tr>

<tr class="alt">

<td>Königlich Essen</td>

<td>Philip Cramer</td>

<td>Germany</td>

</tr>

<tr>

<td>Laughing Bacchus Winecellars</td>

<td>Yoshi Tannamuri</td>

<td>Canada</td>

</tr>

<tr class="alt">

<td>Magazzini Alimentari Riuniti</td>

<td>Giovanni Rovelli</td>

<td>Italy</td>

</tr>

<tr>

<td>North/South</td>

<td>Simon Crowther</td>

<td>UK</td>

</tr>

<tr class="alt">

<td>Paris spécialités</td>

<td>Marie Bertrand</td>

<td>France</td>

</tr>

</table>

</body>

</html>

|  |  |  |
| --- | --- | --- |
| **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 |

Set the position of the table caption  
This example demonstrates how to position the table caption.

|  |  |  |
| --- | --- | --- |
| Table 1.1 Customers | | |
| **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 |
| Magazzini Alimentari Riuniti | Giovanni Rovelli | Italy |
| North/South | Simon Crowther | UK |

**Note:** IE8 supports the caption-side property if a !DOCTYPE is specified.

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 place a border around elements and space elements in relation to other elements.

The image below illustrates the box model:



Explanation of the different parts:

* **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
* **Border** - A border that goes around the padding and content. The border is affected by the background color of the box
* **Padding** - Clears an area around the content. The padding is affected by the background color of the box
* **Content** - The content of the box, where text and images appear

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

## Width and Height of an Element

Remark **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 the padding, borders and margins.

The total width of the element in the example below is 300px:

width:250px;  
padding:10px;  
border:5px solid gray;  
margin:10px;

Let's do the math:  
250px (width)  
+ 20px (left and right padding)  
+ 10px (left and right border)  
+ 20px (left and right margin)  
= 300px

Assume that you had only 250px of space. Let's make an element with a total width of 250px:

<html>

<head>

<style type="text/css">

div.ex

{

width:220px;

padding:10px;

border:5px solid gray;

margin:0px;

}

</style>

</head>

<body>

<img src="250px.gif" width="250" height="1" /><br /><br />

<div class="ex">The line above is 250px wide.<br />

The total width of this element is also 250px.</div>

<p><b>Important:</b> This example will not display correctly in IE8 and earlier

versions!<br />

However, we will solve that problem in the next example.</p>

</body>

</html>

Result:

http://www.w3schools.com/css/250px.gif

The line above is 250px wide.  
The total width of this element is also 250px.

**Important:** This example will not display correctly in IE8 and earlier versions!  
However, we will solve that problem in the next example.

## Browsers Compatibility Issue

The example above does not display properly in IE8 and earlier versions.

IE8 and earlier versions includes padding and border in the width, if a**DOCTYPE is NOT declared**.

To fix this problem, just add a DOCTYPE to the HTML page:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

div.ex

{

width:220px;

padding:10px;

border:5px solid gray;

margin:0px;

}

</style>

</head>

<body>

<img src="250px.gif" width="250" height="1" /><br /><br />

<div class="ex">The line above is 250px wide.<br />

Now the total width of this element is also 250px.</div>

<p><b>Note:</b> In this example we have added a DOCTYPE declaration (above the html element), so it displays correctly in all browsers.</p>

</body>

</html>

CSS Border

## CSS Border Properties

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

## Border Style

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

Remark 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

## 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.

<html>

<head>

<style type="text/css">

p.one

{

border-style:solid;

border-width:5px;

}

p.two

{

border-style:solid;

border-width:medium;

}

p.three

{

border-style:solid;

border-width:1px;

}

</style>

</head>

<body>

<p class="one">Some text.</p>

<p class="two">Some text.</p>

<p class="three">Some text.</p>

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

</body>

</html>

Some text.

Some text.

Some text.

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

## 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".

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

<html>

<head>

<style type="text/css">

p.one

{

border-style:solid;

border-color:red;

}

p.two

{

border-style:solid;

border-color:#98bf21;

}

</style>

</head>

<body>

<p class="one">A solid red border</p>

<p class="two">A solid green border</p>

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

</body></html>

A solid red border

A solid green border

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

## Border - Individual sides

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

<html>

<head>

<style type="text/css">

p

{

border-top-style:dotted;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:solid;

}

</style>

</head>

<body>

<p>2 different border styles.</p>

</body>

</html>

## Example

border-style:dotted solid;

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 border properties in one property. This is called a shorthand property.

The shorthand property for the border properties is "border":

<html>

<head>

<style type="text/css">

p

{

border:5px solid red;

}

</style>

</head>

<body>

<p>This is some text in a paragraph.</p>

</body>

</html>

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

* border-width
* border-style
* border-color

It does not matter if one of the values above are missing (although, border-style is required), as long as the rest are in the specified order.

[All the top border properties in one declaration](http://www.w3schools.com/css/tryit.asp?filename=trycss_border-top)  
This example demonstrates a shorthand property for setting all of the properties for the top border in one declaration.

<html>

<head>

<style type="text/css">

p

{

border-style:solid;

border-top:thick double #ff0000;

}

</style>

</head>

<body>

<p>This is some text in a paragraph.</p>

</body>

</html>

[Set the style of the bottom border](http://www.w3schools.com/css/tryit.asp?filename=trycss_border-bottom-style)  
This example demonstrates how to set the style of the bottom border.

<html>

<head>

<style type="text/css">

p {border-style:solid;}

p.none {border-bottom-style:none;}

p.dotted {border-bottom-style:dotted;}

p.dashed {border-bottom-style:dashed;}

p.solid {border-bottom-style:solid;}

p.double {border-bottom-style:double;}

p.groove {border-bottom-style:groove;}

p.ridge {border-bottom-style:ridge;}

p.inset {border-bottom-style:inset;}

p.outset {border-bottom-style:outset;}

</style>

</head>

<body>

<p class="none">No bottom border.</p>

<p class="dotted">A dotted bottom border.</p>

<p class="dashed">A dashed bottom border.</p>

<p class="solid">A solid bottom border.</p>

<p class="double">A double bottom border.</p>

<p class="groove">A groove bottom border.</p>

<p class="ridge">A ridge bottom border.</p>

<p class="inset">An inset bottom border.</p>

<p class="outset">An outset bottom border.</p>

</body>

</html>

[Set the width of the left border](http://www.w3schools.com/css/tryit.asp?filename=trycss_border-left-width)  
This example demonstrates how to set the width of the left border.

<html>

<head>

<style type="text/css">

p

{

border-style:solid;

border-left-width:15px;

}

</style>

</head>

<body>

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

</body>

</html>

[Set the color of the four borders](http://www.w3schools.com/css/tryit.asp?filename=trycss_border-color)  
This example demonstrates how to set the color of the four borders. It can have from one to four colors.

<html>

<head>

<style type="text/css">

p.one

{

border-style:solid;

border-color:#0000ff;

}

p.two

{

border-style:solid;

border-color:#ff0000 #0000ff;

}

p.three

{

border-style:solid;

border-color:#ff0000 #00ff00 #0000ff;

}

p.four

{

border-style:solid;

border-color:#ff0000 #00ff00 #0000ff rgb(250,0,255);

}

</style>

</head>

<body>

<p class="one">One-colored border!</p>

<p class="two">Two-colored border!</p>

<p class="three">Three-colored border!</p>

<p class="four">Four-colored border!</p>

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

</body>

</html>

[Set the color of the right border](http://www.w3schools.com/css/tryit.asp?filename=trycss_border-right-color)  
This example demonstrates how to set the color of the right border.

<html>

<head>

<style type="text/css">

p

{

border-style:solid;

border-right-color:#ff0000;

}

</style>

</head>

<body>

<p>This is some text in a paragraph.</p>

</body>

</html>

CSS Outlines

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

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

[Draw a line around an element (outline)](http://www.w3schools.com/css/tryit.asp?filename=trycss_outline)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

p

{

border:1px solid red;

outline:green dotted thick;

}

</style>

</head>

<body>

<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is

specified.</p>

</body>

</html>

[Set the style of an outline](http://www.w3schools.com/css/tryit.asp?filename=trycss_outline-style)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

p {border:1px solid red;}

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;}

</style>

</head>

<body>

<p class="dotted">A dotted outline</p>

<p class="dashed">A dashed outline</p>

<p class="solid">A solid outline</p>

<p class="double">A double outline</p>

<p class="groove">A groove outline</p>

<p class="ridge">A ridge outline</p>

<p class="inset">An inset outline</p>

<p class="outset">An outset outline</p>

<b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is

specified.

</body>

</html>

[Set the color of an outline](http://www.w3schools.com/css/tryit.asp?filename=trycss_outline-color)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

p

{

border:1px solid red;

outline-style:dotted;

outline-color:#00ff00;

}

</style>

</head>

<body>

<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>

</body>

</html>

[Set the width of an outline](http://www.w3schools.com/css/tryit.asp?filename=trycss_outline-width)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

p.one

{

border:1px solid red;

outline-style:solid;

outline-width:thin;

}

p.two

{

border:1px solid red;

outline-style:dotted;

outline-width:3px;

}

</style>

</head>

<body>

<p class="one">This is some text in a paragraph.</p>

<p class="two">This is some text in a paragraph.</p>

<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is

specified.</p>

</body>

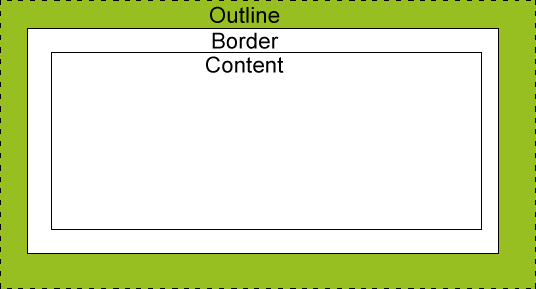
</html>

## 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.



CSS Margin

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 |
| *length* | 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 |

Remark It is possible to use negative values, to overlap content.

## Margin - Individual sides

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

<html>

<head>

<style type="text/css">

p

{

background-color:yellow;

}

p.margin

{

margin-top:100px;

margin-bottom:100px;

margin-right:50px;

margin-left:50px;

}

</style>

</head>

<body>

<p>This is a paragraph with no specified margins.</p>

<p class="margin">This is a paragraph with specified margins.</p>

</body>

</html>

## 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":

<html>

<head>

<style type="text/css">

p

{

background-color:yellow;

}

p.margin

{

margin:100px 50px;

}

</style>

</head>

<body>

<p>This is a paragraph with no specified margins.</p>

<p class="margin">This is a paragraph with specified margins.</p>

</body></html>

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

<html>

<head>

<style type="text/css">

p.ex1 {margin-top:2cm;}

</style>

</head>

<body>

<p>A paragraph with no margins specified.</p>

<p class="ex1">A paragraph with a 2cm top margin.</p>

<p>A paragraph with no margins specified.</p>

</body>

</html>

<html>

<head>

<style type="text/css">

p.bottommargin {margin-bottom:25%;}

</style>

</head>

<body>

<p>This is a paragraph with no margin specified.</p>

<p class="bottommargin">This is a paragraph with a specified bottom margin.</p>

<p>This is a paragraph with no margin specified.</p>

</body>

</html>

CSS Padding

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** |
| *length* | 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:

<html>

<head>

<style type="text/css">

p

{

background-color:yellow;

}

p.padding

{

padding-top:25px;

padding-bottom:25px;

padding-right:50px;

padding-left:50px;

}

</style>

</head>

<body>

<p>This is a paragraph with no specified padding.</p>

<p class="padding">This is a paragraph with specified paddings.</p>

</body>

</html>

## 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":

<html>

<head>

<style type="text/css">

p

{

background-color:yellow;

}

p.padding

{

padding:25px 50px;

}

</style>

</head>

<body>

<p>This is a paragraph with no specified padding.</p>

<p class="padding">This is a paragraph with specified paddings.</p>

</body>

</html>

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

[All the padding properties in one declaration](http://www.w3schools.com/css/tryit.asp?filename=trycss_padding)

<html>

<head>

<style type="text/css">

p.ex1 {padding:2cm;}

p.ex2 {padding:0.5cm 3cm;}

</style>

</head>

<body>

<p class="ex1">This text has equal padding on each side. The padding on each side is 2cm.</p>

<p class="ex2">This text has a top and bottom padding of 0.5cm and a left and right padding of 3cm.</p>

</body>

</html>

CSS Grouping and Nesting Selectors

## Grouping Selectors

To minimize the code, you can group selectors.

Separate each selector with a comma.

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

<html>

<head>

<style type="text/css">

h1,h2,p

{

color:green;

}

</style>

</head>

<body>

<h1>Hello World!</h1>

<h2>Smaller heading!</h2>

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

</body>

</html>

## Nesting Selectors

It is possible to apply a style for a selector within a selector.

In the example below, one style is specified for all p elements, one style is specified for all elements with class="marked", and a third style is specified only for p elements within elements with class="marked":

<html>

<head>

<style type="text/css">

p

{

color:blue;

text-align:center;

}

.marked

{

background-color:red;

}

.marked p

{

color:white;

}

</style>

</head>

<body>

<p>This is a blue, center-aligned paragraph.</p>

<div class="marked">

<p>This p element should not be blue.</p>

</div>

<p>p elements inside a "marked" classed element keeps the alignment style, but has a different text color.</p>

</body>

</html>

CSS Display and Visibility

## 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.

<html>

<head>

<style type="text/css">

h1.hidden {visibility:hidden;}

</style>

</head>

<body>

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

<h1 class="hidden">This is a hidden heading</h1>

<p>Notice that the hidden heading still takes up space.</p>

</body>

</html>

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 the element is not there:

<html>

<head>

<style type="text/css">

h1.hidden {display:none;}

</style>

</head>

<body>

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

<h1 class="hidden">This is a hidden heading</h1>

<p>Notice that the hidden heading does not take up space.</p></body></html>

## 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:

* <span>
* <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 list items as inline elements:

<html>

<head>

<style type="text/css">

li{display:inline;}

</style>

</head>

<body>

<p>Display this link list as a horizontal menu:</p>

<ul>

<li><a href="/html/default.asp" target="\_blank">HTML</a></li>

<li><a href="/css/default.asp" target="\_blank">CSS</a></li>

<li><a href="/js/default.asp" target="\_blank">JavaScript</a></li>

<li><a href="/xml/default.asp" target="\_blank">XML</a></li>

</ul></body></html>

Result:

Display this link list as a horizontal menu:

* [HTML](http://www.w3schools.com/html/default.asp)

* [CSS](http://www.w3schools.com/css/default.asp)

* [JavaScript](http://www.w3schools.com/js/default.asp)

* [XML](http://www.w3schools.com/xml/default.asp)

The following example displays span elements as block elements:

<html>

<head>

<style type="text/css">

span

{

display:block;

}

</style>

</head>

<body>

<h2>Nirvana</h2>

<span>Record: MTV Unplugged in New York</span>

<span>Year: 1993</span>

<h2>Radiohead</h2>

<span>Record: OK Computer</span><span>Year: 1997</span></body></html>

## Nirvana

Record: MTV Unplugged in New YorkYear: 1993

## Radiohead

Record: OK ComputerYear: 1997

**Note:** Changing the display type of an element changes only how the element is displayed, NOT what kind of element it is. For example: An inline element set to display:block is not allowed to have a block element nested inside of it.

<html>

<head>

<style type="text/css">

p {display:inline;}

</style>

</head>

[How to display an element as an inline element.](http://www.w3schools.com/css/tryit.asp?filename=trycss_display)

<body>

<p>A display property with a value of "inline" results in</p>

<p>no distance between two elements.</p>

</body>

</html>

A display property with a value of "inline" results in no distance between two elements.

How to display an element as a block element  
This example demonstrates how to display an element as a block element.

 <html>

<head>

<style type="text/css">

span

{

display:block;

}

</style>

</head>

<body>

<span>A display property with a value of "block" results in</span> <span>a line break between the two elements.</span>

</body>

</html>

A display property with a value of "block" results ina line break between the two elements.

How to make a table element collapse  
This example demonstrates how to make a table element collapse.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head>

<style type="text/css">

tr.collapse {visibility:collapse;}

</style>

</head>

<body>

<table border="1">

<tr>

<td>Peter</td>

<td>Griffin</td>

</tr>

<tr class="collapse">

<td>Lois</td>

<td>Griffin</td>

</tr>

</table>

<p><b>Note:</b> IE8 and earlier support visibility:collapse only if a !DOCTYPE is specified.</p>

</body>

</html>

CSS Positioning

## 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 five different positioning methods.

div {

position: static; /\* Default, no need to set unless forcing back into this state. \*/

}

div {

position: relative;

}

div {

position: absolute;

}

div {

position: fixed;

}

div {

position: inherit; /\* Take value from parent \*/

}

The position value in CSS deals with layout and manipulating elements to be in your desired visual place. There are only the five values shown above, and really only three since static and inherit are fairly rarely needed for this property.

## 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.

#### Static

Every element is static positioned by default. The element resides in the normal page flow. [left](http://css-tricks.com/almanac/properties/l/left/)/right/top/bottom/[z-index](http://css-tricks.com/almanac/properties/z/z-index/) have no effect on a static positioned element. There is rarely a need to set this value unless the element has been set with another matching selector to a different value and you need to set it back.

## Fixed Positioning

An element with fixed position is positioned relative to the browser window.

It will not move even if the window is scrolled:

#### Fixed

Element is removed from the flow of the document like absolute positioned elements. In fact they behave almost the same, only fixed positioned elements are always relative to the document, not any particular parent, and are unaffected by scrolling. It's usually used to provide a persistant visual element, like navigation bar that is always visible.

<html>

<head>

<style type="text/css">

p.pos\_fixed

{

position:fixed;

top:30px;

right:5px;

}

</style>

</head>

<body>

<p class="pos\_fixed">Some more text</p>

<p><b>Note:</b> IE7 and IE8 supports the fixed value only if a

!DOCTYPE is specified.</p>

<p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p>

</body>

</html>

**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.

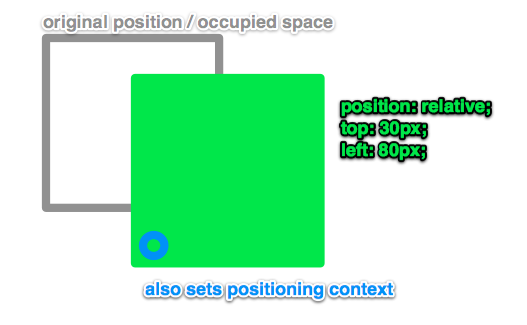
Fixed positioned elements can overlap other elements.

## Relative Positioning

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

#### Relative

Element's original position (as if it were static) remains in the flow of the document. But now [left](http://css-tricks.com/almanac/properties/l/left/)/right/top/bottom/[z-index](http://css-tricks.com/almanac/properties/z/z-index/) do work. The positional properties "nudge" the element from the original position in that direction.



<html>

<head>

<style type="text/css">

h2.pos\_left

{

position:relative;

left:-20px;

}

h2.pos\_right

{

position:relative;

left:20px;

}

</style>

</head>

<body>

<h2>This is a heading with no position</h2>

<h2 class="pos\_left">This heading is moved left according to its normal position</h2>

<h2 class="pos\_right">This heading is moved right according to its normal position</h2>

<p>Relative positioning moves an element RELATIVE to its original position.</p>

<p>The style "left:-20px" subtracts 20 pixels from the element's original left position.</p>

<p>The style "left:20px" adds 20 pixels to the element's original left position.</p>

</body>

</html>

## This is a heading with no position

## This heading is moved left according to its normal position

## This heading is moved right according to its normal position

Relative positioning moves an element RELATIVE to its original position.

The style "left:-20px" subtracts 20 pixels from the element's original left position.

The style "left:20px" adds 20 pixels to the element's original left position.

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.

<html>

<head>

<style type="text/css">

h2.pos\_top

{

position:relative;

top:-50px;

}

</style>

</head>

<body>

<h2>This is a heading with no position</h2>

<h2 class="pos\_top">This heading is moved upwards according to its normal position</h2>

<p><b>Note:</b> Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.</p>

</body>

</html>

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>:

#### positionabsolute.pngAbsolute

Element is removed from the flow of the document (other elements will behave as if it's not even there). All the other positional properties work on it. Essentially you are able to declare the exact position you want the element to appear.

Note that 1) without a width set, the element will stretch only as wide as the content it contains and 2) you can set, for instance, both a left and right value and the element will stretch to touch both points. So you can fill a screen by setting top: 0; left: 0; bottom: 0; right: 0;

<html>

<head>

<style type="text/css">

h2

{

position:absolute;

left:100px;

top:150px;

}

</style>

</head>

<body>

<h2>This is a heading with an absolute position</h2>

<p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>

</body>

</html>

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.

#### Inherit

The position value doesn't cascade, so this can be used to specifically force it to, and inherit the positioning value from its parent.

## 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:

<html>

<head>

<style type="text/css">

img

{

position:absolute;

left:0px;

top:0px;

z-index:-1;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<img src="pro.gif" width="100" height="140" />

<p>Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>

</html>

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.

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.

<html>

<head>

<style type="text/css">

img

{

position:absolute;

clip:rect(0px,60px,200px,0px);

}

</style>

</head>

<body>

<img src="w3css.gif" width="100" height="140" />

</body>

</html>

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.

<html>

<head>

<style type="text/css">

div.scroll

{

background-color:#00FFFF;

width:100px;

height:100px;

overflow:scroll;

}

div.hidden

{

background-color:#00FF00;

width:100px;

height:100px;

overflow:hidden;

}

</style>

</head>

<body>

<p>The overflow property specifies what to do if the content of an element exceeds the size of the element's box.</p>

<p>overflow:scroll</p>

<div class="scroll">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>

<p>overflow:hidden</p>

<div class="hidden">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>

</body>

</html>

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

<html>

<head>

<style type="text/css">

div

{

background-color:#00FFFF;

width:150px;

height:150px;

overflow:auto;

}

</style>

</head>

<body>

<p>The overflow property decides what to do if the content inside an element exceeds the given width and height properties.</p>

<div>

You can use the overflow property when you want to have better control of the layout. Try to change the overflow property to: visible, hidden, scroll, or inherit and see what happens. The default value is visible.

</div>

</body>

</html>

Change the cursor  
This example demonstrates how to change the cursor.

<html>

<body>

<p>Mouse over the words to change the cursor.</p>

<span style="cursor:auto">auto</span><br />

<span style="cursor:crosshair">crosshair</span><br />

<span style="cursor:default">default</span><br />

<span style="cursor:e-resize">e-resize</span><br />

<span style="cursor:help">help</span><br />

<span style="cursor:move">move</span><br />

<span style="cursor:n-resize">n-resize</span><br />

<span style="cursor:ne-resize">ne-resize</span><br />

<span style="cursor:nw-resize">nw-resize</span><br />

<span style="cursor:pointer">pointer</span><br />

<span style="cursor:progress">progress</span><br />

<span style="cursor:s-resize">s-resize</span><br />

<span style="cursor:se-resize">se-resize</span><br />

<span style="cursor:sw-resize">sw-resize</span><br />

<span style="cursor:text">text</span><br />

<span style="cursor:w-resize">w-resize</span><br />

<span style="cursor:wait">wait</span><br />

</body>

</html>

CSS Float

## 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:

## 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:

<!DOCTYPE html>

<html>

<head>

<style>

.thumbnail

{

float:left;

width:110px;

height:90px;

margin:5px;

}

.text\_line

{

clear:both;

margin-bottom:2px;

}

</style>

</head>

<body>

<h3>Image Gallery</h3>

<p>Try resizing the window to see what happens when the images does not have enough room.</p>

<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">

<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">

<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">

<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">

<h3 class="text\_line">Second row</h3>

<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">

<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">

<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">

<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">

</body>

</html>

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

<!DOCTYPE html>

<html>

<head>

<style>

span

{

float:left;

width:0.7em;

font-size:400%;

font-family:algerian,courier;

line-height:80%;

}

</style>

</head>

<body>

<p>

<span>T</span>his is some text.

This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

</p>

<p>

In the paragraph above, the first letter of the text is embedded in a span element.

The span element has a width that is 0.7 times the size of the current font.

The font-size of the span element is 400% (quite large) and the line-height is 80%.

The font of the letter in the span will be in "Algerian".

</p>

</body>

</html>

## Center Aligning Using the margin Property

Block elements can be 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

<!DOCTYPE html>

<html>

<head>

<style>

.center

{

margin:auto;

width:70%;

background-color:#b0e0e6;

}

</style>

</head>

<body>

<div class="center">

<p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>

<p>'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>

</div>

<p><b>Note: </b>Using margin:auto will not work in IE8, unless a !DOCTYPE is declared.</p>

</body>

</html>

## Left and Right Aligning Using the position Property

One method of aligning elements is to use absolute positioning:

## Creating a Transparent Image

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

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

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

## Image Transparency - Hover Effect

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 \*/  
}