CSS Basic

CSS Styling

CSS Box Model

CSS Advanced

CSS Examples

CSS Quiz

CSS References

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.

CSS Example

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

Try it yourself »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Introduction

« Previous Next Chapter »

What You Should Already Know

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

- HTML / XHTML

If you want to study these subjects first, find the tutorials on our Home page.

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 Demo

An HTML document can be displayed with different styles: See how it works

Styles 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 <font>, 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 be removed from the HTML document, and stored in a separate CSS file.

All browsers support CSS today.

CSS Saves a Lot of Work!

CSS defines HOW HTML elements are to be displayed.

Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Syntax

« Previous

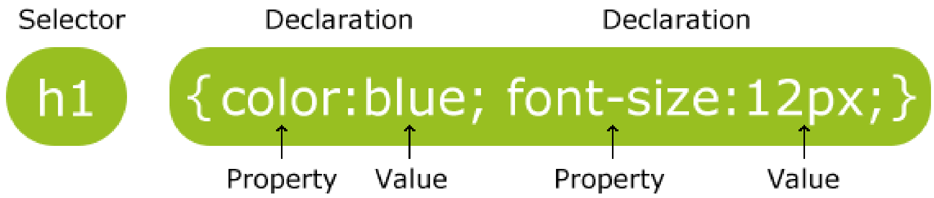
Next Chapter »

Examples

- Look at Example 1
- Look at Example 2

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

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

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

To make the CSS more readable, you can put one declaration on each line, like this:

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

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

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Id and Class

« Previous Next Chapter »

The id and class Selectors

In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

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

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

Try it yourself »

Do **NOT** start an ID name with a number! It will not work in Mozilla/Firefox.

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 any 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

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

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;}
```

Try it yourself »

Do **NOT** start a class name with a number! This is only supported in Internet Explorer.

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS How To...

« Previous   Next Chapter »

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

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).

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!

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Background

« Previous   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 defined in the body selector:

Example

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

Try it yourself »

The background color can be specified by:

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

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('gradient2.png');
}
```

Try it yourself »

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;
}
```

Try it yourself »

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;
}
```

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:top right;
}
```

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 top right;}
```

Try it yourself »

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

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

It does not matter if one of the property values are 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.

All CSS Background Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
background	Sets all the background properties in one declaration	*background-color* *background-image* *background-repeat* *background-attachment* *background-position* inherit	1
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed inherit	1
background-color	Sets the background color of an element	*color-rgb* *color-hex* *color-name* transparent inherit	1
background-image	Sets the background image for an element	url(*URL*) none inherit	1
background-position	Sets the starting position of a background image	top left top center top right center left center center center right bottom left bottom center bottom right *x% y%* *xpos ypos* inherit	1

background-repeat	Sets if/how a background image will be repeated	repeat
		repeat-x
		repeat-y
		no-repeat
		inherit
			1

« Previous   Next Chapter »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Text

« Previous

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. The color can be specified by:

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

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);}
```

Try it yourself »

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;}
h4 {text-decoration:blink;}
```

Try it yourself »

It is not recommended to underline text that is not a link, as this often confuse 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-indentation 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.

All CSS Text Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
color	Sets the color of a text	*color*	1
direction	Sets the text direction	ltr rtl	2
line-height	Sets the distance between lines	normal *number*	1

		length *%*	
letter-spacing	Increase or decrease the space between characters	normal *length*	1
text-align	Aligns the text in an element	left right center justify	1
text-decoration	Adds decoration to text	none underline overline line-through blink	1
text-indent	Indents the first line of text in an element	*length* *%*	1
text-shadow		none *color* *length*	
text-transform	Controls the letters in an element	none capitalize uppercase lowercase	1
unicode-bidi		normal embed bidi-override	2
vertical-align	Sets the vertical alignment of an element	baseline sub super top text-top middle bottom text-bottom *length* *%*	1
white-space	Sets how white space inside an element is handled	normal pre nowrap	1
word-spacing	Increase or decrease the space between words	normal *length*	1

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

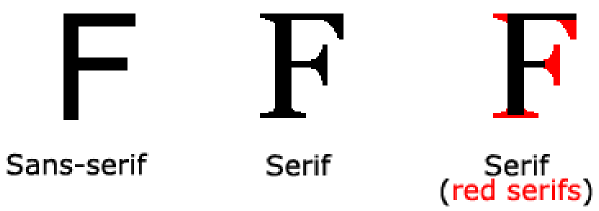
CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Font

« Previous Next Chapter »

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

Difference Between Serif and Sans-serif Fonts



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

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

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 »

The example above allows Firefox, Chrome, and Safari to resize the text, **but not Internet Explorer**.

The text can be resized in all browsers using the zoom tool (however, this resizes the entire page, not just the text).

Set Font Size With Em

To avoid the resizing problem with 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*

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 IE. When resizing the text, it 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.

All CSS Font Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
font	Sets all the font properties in one declaration	*font-style* *font-variant* *font-weight* *font-size/line-height* *font-family* caption icon menu message-box small-caption status-bar inherit	1
font-family	Specifies the font family for text	*family-name* *generic-family* inherit	1
font-size	Specifies the font size of text	xx-small x-small small medium large x-large xx-large smaller larger *length* *%* inherit	1
font-style	Specifies the font style for text	normal italic oblique inherit	1
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal small-caps inherit	1
font-weight	Specifies the weight of a font	normal bold bolder	1

			lighter 100 200 300 400 500 600 700 800 900 inherit	

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Links

« Previous Next Chapter »

Links can be styled in different ways.

Styling Links

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

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

```
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 */
```

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 »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Lists

« Previous   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 property values are for unordered lists, and some for ordered lists.

Values for Unordered Lists

Value	Description
none	No marker
disc	Default. The marker is a filled circle
circle	The marker is a circle
square	The marker is a square

Values for Ordered Lists

Value	Description
armenian	The marker is traditional Armenian numbering
decimal	The marker is a number
decimal-leading-zero	The marker is a number padded by initial zeros (01, 02, 03, etc.)
georgian	The marker is traditional Georgian numbering (an, ban, gan, etc.)
lower-alpha	The marker is lower-alpha (a, b, c, d, e, etc.)
lower-greek	The marker is lower-greek (alpha, beta, gamma, etc.)
lower-latin	The marker is lower-latin (a, b, c, d, e, etc.)
lower-roman	The marker is lower-roman (i, ii, iii, iv, v, etc.)
upper-alpha	The marker is upper-alpha (A, B, C, D, E, etc.)
upper-latin	The marker is upper-latin (A, B, C, D, E, etc.)
upper-roman	The marker is upper-roman (I, II, III, IV, V, etc.)

Note: No versions of Internet Explorer (including IE8) support the property values "decimal-leading-zero", "lower-greek", "lower-latin", "upper-latin", "armenian", or "georgian".

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;
}
li
{
background-image: url(sqpurple.gif);
background-repeat: no-repeat;
background-position: 0px 5px;
padding-left: 14px;
}
```

Try it yourself »

Example explained:

- For ul:
  - 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 li:
  - Set the URL of the image, and show it only once (no-repeat)
  - Position the image where you want it (left 0px and down 5px)
  - Position the text in the list with padding-left

List - Shorthand property

It is also possible to specify all the list properties in one, single property. This is called a shorthand property.

The shorthand property used for lists, is the list-style property:

Example

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

Try it yourself »

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

- list-style-type
- list-style-position (for a description, see the CSS properties table below)
- list-style-image

It does not matter if one of the values above are missing, as long as the rest are in the specified order.

More Examples

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

All CSS List Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
list-style	Sets all the properties for a list in one declaration	*list-style-type* *list-style-position* *list-style-image* inherit	1
list-style-image	Specifies an image as the list-item marker	URL none inherit	1
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow	inside outside inherit	1
list-style-type	Specifies the type of list-item marker	none disc circle square decimal decimal-leading-zero armenian georgian lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman inherit	1

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Tables

« Previous

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, th, and 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 »

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:

Example

```
td
{
text-align:right;
}
```

Try it yourself »

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

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 »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Box Model

« Previous   Next Chapter »

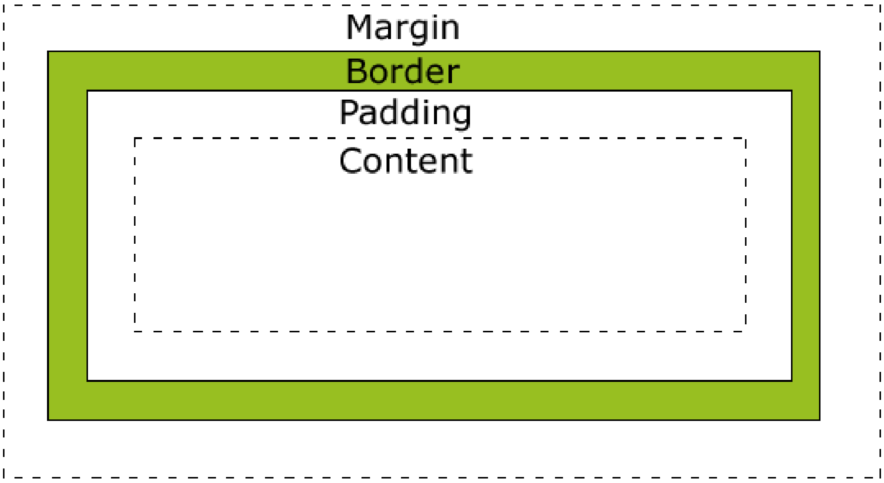
The CSS Box Model

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 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, and it is completely transparent
- **Border** - A border that lies 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

Important: When you specify the width and height properties of an element with CSS, you are just setting the width and height of the content area. To know the full size of the element, you must also add the padding, border and margin.

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

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

Example

```
width:220px;
padding:10px;
border:5px solid gray;
margin:0px;
```

Try it yourself »

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

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

Browsers Compatibility Issue

If you tested the previous example in Internet Explorer, you saw that the total width was not exactly 250px.

IE includes padding and border in the width, when the width property is set, **unless a DOCTYPE is declared**.

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

Example

```
<!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>
```

Try it yourself »

« Previous Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Border

« Previous Next Chapter »

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.

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".

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

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

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

Example

```
border:5px solid red;
```

Try it yourself »

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.

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.

All CSS Border Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
border	Sets all the border properties in one declaration	*border-width* *border-style* *border-color*	1
border-bottom	Sets all the bottom border properties in one declaration	*border-bottom-width* *border-bottom-style* *border-bottom-color*	1
border-bottom-color	Sets the color of the bottom border	*border-color*	2
border-bottom-style	Sets the style of the bottom border	*border-style*	2
border-bottom-width	Sets the width of the bottom border	*border-width*	1
border-color	Sets the color of the four borders	*color_name* *hex_number* *rgb_number* transparent inherit	1
border-left	Sets all the left border properties in one declaration	*border-left-width* *border-left-style* *border-left-color*	1
border-left-color	Sets the color of the left border	*border-color*	2
border-left-style	Sets the style of the left border	*border-style*	2
border-left-width	Sets the width of the left border	*border-width*	1
border-right	Sets all the right border properties in one declaration	*border-right-width* *border-right-style* *border-right-color*	1
border-right-color	Sets the color of the right border	*border-color*	2

border-right-style	Sets the style of the right border	*border-style*	2
border-right-width	Sets the width of the right border	*border-width*	1
border-style	Sets the style of the four borders	none hidden dotted dashed solid double groove ridge inset outset inherit	1
border-top	Sets all the top border properties in one declaration	*border-top-width* *border-top-style* *border-top-color*	1
border-top-color	Sets the color of the top border	*border-color*	2
border-top-style	Sets the style of the top border	*border-style*	2
border-top-width	Sets the width of the top border	*border-width*	1
border-width	Sets the width of the four borders	thin medium thick *length* inherit	1

« Previous Next Chapter »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Outlines

« Previous   Next Chapter »

An outline is a line that is drawn around elements, outside the border edge, to make the element "stand out".

The outline properties specifies 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.

All CSS Outline Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
outline	Sets all the outline properties in one declaration	*outline-color* *outline-style* *outline-width* inherit	2
outline-color	Sets the color of an outline	*color_name* *hex_number* *rgb_number* invert inherit	2
outline-style	Sets the style of an outline	none dotted dashed solid double groove ridge inset outset inherit	2
outline-width	Sets the width of an outline	thin medium thick *length* inherit	2

« Previous   Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Margin

« Previous   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 sets the margin. The result of this is dependant of the browser
length	Defines a fixed margin (in pixels, pt, em, etc.)
%	Defines a margin in % of the containing element

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:

Example

```
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
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

```
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 of a text using a percent value.

All CSS Margin Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
margin	A shorthand property for setting the margin properties in one declaration	*margin-top* *margin-right* *margin-bottom* *margin-left*	1
margin-bottom	Sets the bottom margin of an element	auto *length* %	1
margin-left	Sets the left margin of an element	auto *length* %	1
margin-right	Sets the right margin of an element	auto *length* %	1
margin-top	Sets the top margin of an element	auto *length* %	1

« Previous   Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Padding

« Previous 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
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:

Example

```
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
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

```
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

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Grouping and Nesting Selectors

« Previous Next Chapter »

Grouping Selectors

In style sheets there are often elements with the same style.

```
h1
{
color:green;
}
h2
{
color:green;
}
p
{
color:green;
}
```

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:

Example

```
h1,h2,p
{
color:green;
}
```

Try it yourself »

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, and a separate style is specified for p elements nested within the "marked" class:

Example

```
p
{
color:blue;
text-align:center;
}
.marked
{
background-color:blue;
}
.marked p
{
color:white;
}
```

Try it yourself »

« Previous Next Chapter »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

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.

All CSS Dimension Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
height	Sets the height of an element	auto *length* % inherit	1
max-height	Sets the maximum height of an element	none *length* % inherit	2
max-width	Sets the maximum width of an element	none *length* % inherit	2
min-height	Sets the minimum height of an element	*length* % inherit	2
min-width	Sets the minimum width of an element	*length* % inherit	2
width	Sets the width of an element	auto *length* % inherit	1

« Previous Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

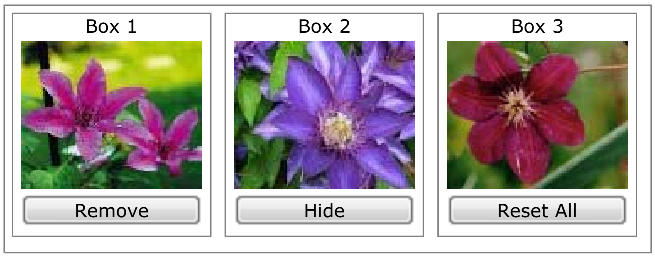
CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Display and Visibility

« Previous 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 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:

- <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:

Example

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

Try it yourself »

The following example displays span elements as block elements:

Example

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

Try it yourself »

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.

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

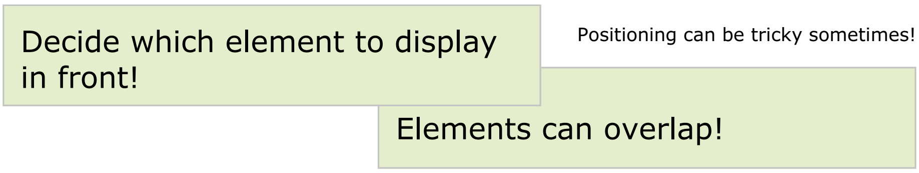
CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Positioning

« Previous

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 fixed position is positioned relative to the browser window.

It will not move even if the window is scrolled:

Example

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

Try it yourself »

Note: Internet Explorer supports 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 a 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 element 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.

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

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
bottom	Sets the bottom margin edge for a positioned box	auto *length* *%* inherit	2
clip	Clips an absolutely positioned element	*shape* auto inherit	2
cursor	Specifies the type of cursor to be displayed	*url* auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	2
left	Sets the left margin edge for a positioned box	auto *length* *%* inherit	2
overflow	Specifies what happens if content overflows an element's box	auto hidden scroll visible inherit	2
position	Specifies the type of positioning for an element	absolute fixed relative static inherit	2
right	Sets the right margin edge for a positioned box	auto *length* *%* inherit	2
top	Sets the top margin edge for a positioned box	auto *length* *%* inherit	2
z-index	Sets the stack order of an element	*number* auto inherit	2

« Previous Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

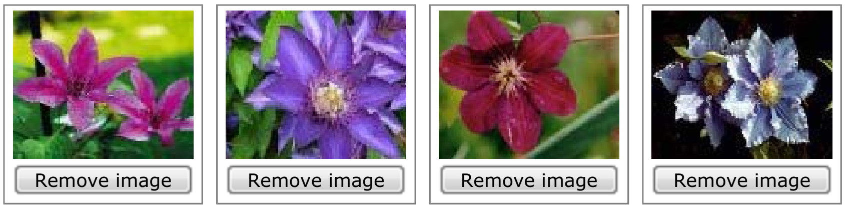
CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Float

« Previous 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 very often used for 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

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
clear	Specifies which sides of an element where other floating elements are not allowed	left right both none inherit	1
float	Specifies whether or not a box should float	left right none inherit	1

« Previous Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Horizontal Align

« Previous 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 aligned by setting the left and right margins to "auto".

Note: Using margin:auto will not work in Internet Explorer, **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: Aligning has no effect if the width is 100%.

Note: In IE 5 there is a margin handling bug for block elements. To make the example above work in IE5, add some extra code. Try it yourself

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 also another problem with IE 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, IE 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 also another problem with IE when using the float property. If the !DOCTYPE declaration is missing, IE 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;
}
```

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Pseudo-classes

« Previous Next Chapter »

CSS pseudo-classes are used to add special effects to some selectors.

Syntax

The syntax of pseudo-classes:

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

CSS classes can also be used with pseudo-classes:

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

Anchor Pseudo-classes

Links can be displayed in different ways in a CSS-supporting browser:

Example

```
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 */
```

Try it yourself »

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

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

Note: Pseudo-class names are not case-sensitive.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

```
a.red:visited {color:#FF0000;}

<a class="red" href="css_syntax.asp">CSS Syntax</a>
```

If the link in the example above has been visited, it will be displayed in red.

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 IE 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

```
<html>
<head>
```

```
<style type="text/css">
p:first-child
{
color:blue;
}
</style>
</head>

<body>
<p>I am a strong man.</p>
<p>I am a strong man.</p>
</body>
</html>
```

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

```
<html>
<head>
<style type="text/css">
p > i:first-child
{
font-weight:bold;
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
</body>
</html>
```

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

```
<html>
<head>
<style type="text/css">
p:first-child i
{
color:blue;
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
</body>
</html>
```

Try it yourself »

CSS - The :lang Pseudo-class

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

Note: Internet Explorer 8 (and higher) supports the :lang pseudo-class 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 type="text/css">
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.

Pseudo-classes

The "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Pseudo name	Description	CSS
:active	Adds a style to an element that is activated	1
:first-child	Adds a style to an element that is the first child of another element	2
:focus	Adds a style to an element that has keyboard input focus	2
:hover	Adds a style to an element when you mouse over it	1
:lang	Adds a style to an element with a specific lang attribute	2
:link	Adds a style to an unvisited link	1
:visited	Adds a style to a visited link	1

« Previous

Next Chapter »

CSS Basic

CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling

Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model

CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced

CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples

CSS Examples

CSS Quiz

CSS Quiz
CSS Exam

CSS References

CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Pseudo-elements

« Previous Next Chapter »

CSS pseudo-elements are used to add special effects to some selectors.

Syntax

The syntax of pseudo-elements:

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

CSS classes can also be used with pseudo-elements:

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

The :first-line Pseudo-element

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

In the following example the browser formats the first line of text in a p element according to the style in the "first-line" pseudo-element (where the browser breaks the line, depends on the size of the browser window):

Example

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

Try it yourself »

Note: The "first-line" pseudo-element can only be used with block-level elements.

Note: 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:

Example

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

Try it yourself »

Note: The "first-letter" pseudo-element can only be used with block-level elements.

Note: 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:

```
p.article:first-letter {color:#ff0000;}

<p class="article">A paragraph in an article</p>
```

The example above will display the first letter of all paragraphs with class="article", in red.

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 »

Pseudo-elements

The "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Pseudo name	Description	CSS
:after	Adds content after an element	2
:before	Adds content before an element	2
:first-letter	Adds a style to the first character of a text	1
:first-line	Adds a style to the first line of a text	1

« Previous Next Chapter »

CSS Basic
CSS HOME
CSS Introduction
CSS Syntax
CSS Id & Class
CSS How To

CSS Styling
Styling Backgrounds
Styling Text
Styling Fonts
Styling Links
Styling Lists
Styling Tables

CSS Box Model
CSS Box Model
CSS Border
CSS Outline
CSS Margin
CSS Padding

CSS Advanced
CSS Grouping/Nesting
CSS Dimension
CSS Display
CSS Positioning
CSS Floating
CSS Align
CSS Pseudo-class
CSS Pseudo-element
CSS Navigation Bar
CSS Image Gallery
CSS Image Opacity
CSS Image Sprites
CSS Media Types
CSS Attribute Selectors
CSS Don't
CSS Summary

CSS Examples
CSS Examples

CSS Quiz
CSS Quiz
CSS Exam

CSS References
CSS Reference
CSS Reference A to Z
CSS Reference Aural
CSS Web Safe Fonts
CSS Units
CSS Colors
CSS Colorvalues
CSS Colornames

CSS Navigation Bar

« Previous 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 <ul> and <li> 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 <li> elements as inline, in addition to the "standard" code above:

Example

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

Try it yourself »

Example explained:

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

Tip: Also take a look at our fully styled horizontal navigation bar example.

Floating List Items

In the example above the links have different widths.

For all the links to have an equal width, float the <li> 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 Next Chapter »