

# CSS (Cascading Style Sheets) in one page

**Contents:**  
 Templates: [Style sheet into the document](#), [Link to an external style sheet](#), [Syntax](#)  
 Main elements: [Media types](#), [Selectors](#), [Properties](#)  
 Properties: [Box](#), [Show boxes \(Controlling box generation\)](#), [Visual superposition of boxes \(Positioning schemes\)](#), [Visual effects](#), [Colors](#), [Background](#), [Fonts](#), [Text](#), [Generated content](#), [Automatic counters and numbering](#), [Lists](#), [Tables](#)  
 Supplemental information: [Units](#), [Box model](#)  
 Other: [Appendix "CSS Property Index"](#), [Related References \(Review, Documentation, etc.\)](#), [Tools \(Validators, etc.\)](#), [Related themes](#), [Miscellaneous](#)  
 Similar Sites: [MANUAL](#), [Cheat sheets](#), [HTML](#), [CSS](#), [XML](#), [DTD](#), [JavaScript](#), [W3C DOM](#), [SQL](#), [SSI](#), [Unicode](#), [Free Unicode Fonts](#), [Tell a friend](#), [Free Icons](#), [Ilibitum Corp.](#)



## Template: put the style sheet into the document

```
your_document.html:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional/EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<title>... replace with your document's title ...</title>

<style type="text/css">
/* CSS Document */
...replace with your css's content ...
body {
background: white;
color: black
}
a:link { color: red }
a:visited { color: maroon }
a:active { color: fuchsia }
h1 { color: red }
p { color: blue }
... replace with your css's content ...
</style>

</head>
<body>
... replace with your document's content ...
<h1>Bach's home page</h1>
<p>Johann Sebastian Bach was a prolific composer.</p>
... replace with your document's content ...
</body>
</html>
```

## Syntax

```
/* Comment */
@import url("fancyfonts.css") media_type;

@media media_type {
selector {
property: values;
property: values;
}
}

/* @media media_type is optional */
```

## Media types (media\_type)

all	For all devices
braille	For braille tactile feedback devices
embossed	For paged braille printers
handheld	For handheld devices
print	For paged material
projection	For projected presentations
screen	For color computer screens
speech	For speech synthesizers
tty	For media using a fixed-pitch character grid
tv	For television-type devices

## Selectors

*	All elements (universal selector)
div	<div> (type selector)
div *	Elements within <div>
div span	<span> within <div> (descendant selector)
div, span	<div> and <span> (grouping selectors)
div > span	<span> with <div> as parent (child selector)
div + span	<span> preceded by <div>...</div> (adjacent sibling selector)
*.any_class	Elements of class="any_class" (class selector)
.any_class	<div class="any_class"> (class selector)
#item_id	Element with id="item_id" (id selector)
div#item_id	<div id="any_class"> (id selector)
div[any_attr]	Elements with sets any_attr attribute (attribute selector)
div[any_attr]	<div any_attr="..."> (attribute selector)
div[any_attr="value"]	<div any_attr="value"> (attribute selector)
div[any_attr~="value"]	<div any_attr="another value another"> (attribute selector)
div[any_attr =value]	<div any_attr="value another"> (attribute selector)
div:first-child	First child of <div> (pseudo-class)
div:first-line	First line of <div> (pseudo-element)
div:first-letter	First letter of <div> (pseudo-element)
div:before	Element before <div> (pseudo-element)
div:after	Element after <div> (pseudo-element)
a:link	Non-active, unvisited links without mouse over (link pseudo-class)
a:visited	Visited links (link pseudo-class)
div:hover	<div> when mouse over (dynamic pseudo-class)
div:active	Active <div> (dynamic pseudo-class)
div:focus	<div> with focus (dynamic pseudo-class)
div:lang(la)	<div> with language la (language pseudo-class)

\* Use any element instead div (span, p, a, h1, etc.)

[More and details about selectors >>>](#)

## Units

px	Pixels
em	1 em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
%	Percentage
in	Inches. 1 inch is equal to 2.54 centimeters.
cm	Centimeters
mm	Millimeters. 1 millimeter is equal to 1/10th of a centimeter
pt	Points. 1 pt is equal to 1/72nd of an inch.
pc	Picas. 1 pc is equal to 12 pt
#789abc	RGB HEX Notation
#abc	Equates to "#aabbc"
rgb(0,100,255)	Value (0 to 255) of each of red, green and blue
rgb(0%, 50%, 100%)	Value (0% to 100%) of each of red, green and blue
ms	Milliseconds
s	Seconds
Hz	Hertz
kHz	Kilohertz
0	0 requires no unit

[More and details about lengths >>>](#)

## Validators:

- W3C Markup Validator - Also known as the HTML validator, it helps check Web documents in formats like HTML and XHTML, SVG or MathML.
- Checklink - Checks anchors (hyperlinks) in a HTML/XHTML document. Useful to find broken links, etc.
- CSS Validator - validates CSS stylesheets or documents using CSS stylesheets.
- RDF Validator
- Feed Validator - it helps check newfeeds in formats like ATOM, RSS of various flavors.
- P3P Validator - Checks whether a site is P3P enabled and controls protocol and syntax of Policy-Reference-File and Policy
- XML Schema Validator
- MUTAT - a human-centered testing tool (framework)

## Template: link to an external style sheet

```
your_document.html:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional/EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<title>... replace with your document's title ...</title>

<link rel="stylesheet" type="text/css" href="your_document.css" />
<link rel="stylesheet" type="text/css" media="print, handheld" href="another_document.css" />

</head>

<body>
... replace with your document's content ...
<h1>Bach's home page</h1>
<p>Johann Sebastian Bach was a prolific composer.</p>
... replace with your document's content ...
</body>
</html>

your_document.css:
/* CSS Document */

... replace with your css's content ...
body {
background: white;
color: black
}
a:link { color: red }
a:visited { color: maroon }
a:active { color: fuchsia }
h1 { color: red }
p { color: blue }
... replace with your css's content ...
```

## Related References

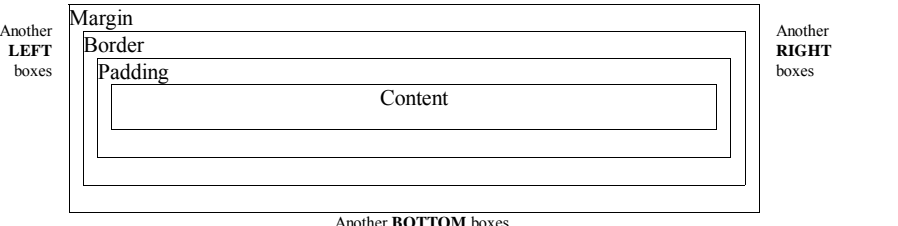
- Main pages:**  
[W3C Cascading Style Sheets Home Page](#)
- Documentation:**  
[Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification](#)  
[Syntax and basic data types](#)  
[Selectors](#)  
[Assigning property values, Cascading, and Inheritance](#)  
[Media types](#)  
[Box model](#)  
[Visual formatting model](#)  
[Visual formatting model details](#)  
[Visual effects](#)  
[Generated content, automatic numbering, and lists](#)  
[Paged media](#)  
[Colors and Backgrounds](#)  
[Fonts](#)  
[Text](#)  
[Tables](#)  
[User interface](#)  
[Aural style sheets](#)  
[Default style sheet for HTML 4.0](#)  
[Full property sheet](#)

## Related themes

- [HTML/XHTML \(HTML SU\)](#)
- [XML - eXtensible Markup Language \(XML SU\)](#)
- [DTD - Doc Type Declaration](#)
- [Other Manuals \(MANUAL SU\)](#)
- [Free archive of icons for download \(IconsFree.org\)](#)

## Box model

Another TOP boxes



[More and details about box model >>>](#)

## Properties

### Box

<b>width</b>	Specifies the content width of boxes generated by block-level and replaced elements	<length>   <percentage>   auto   inherit	p { width: 100px }
<b>min-width</b>	These two properties allow authors to constrain content widths to a certain range	<length>   <percentage>   inherit	h1 { min-width: 10px }
<b>max-width</b>		<length>   <percentage>   none   inherit	div { max-width: 600px }
<b>height</b>	Specifies the content height of boxes generated by block-level, inline-block and replaced elements	<length>   <percentage>   auto   inherit	p { height: 150px }
<b>min-height</b>	These two properties allow authors to constrain content widths to a certain range	<length>   <percentage>   inherit	h1 { min-height: 10px }
<b>max-height</b>		<length>   <percentage>   none   inherit	div { max-height: 600px }
<b>margin-top</b>	Margin properties specify the width of the margin area of a box	<margin-width>   inherit	<b>stylesheet fragment:</b>
<b>margin-bottom</b>		Negative values for margin properties are allowed, but there may be implementation-specific limits.	ul { background: yellow; margin: 12px 12px 12px 12px; padding: 3px 3px 3px 3px; }
<b>margin-right</b>			first box
<b>margin-left</b>			second box
<b>padding</b>	Padding properties specify the width of the padding area of a box. The 'padding' shorthand property sets the padding for all four sides while the other padding properties only set their respective side.	<padding-width>   inherit	li { color: white; background: blue; margin: 12px 12px 12px 12px; padding: 12px 0px 12px 12px; list-style: none }
<b>padding-top</b>			third box (with border)
<b>padding-bottom</b>			fourth box (strange)
<b>padding-right</b>			Attention! Some properties is not supported some
<b>padding-left</b>			
<b>border-top-width</b>	The border properties specify the width, color, and style of the border area of a box. These properties apply to all elements. Note. Notably for HTML, user agents may render borders for certain user interface elements (e.g., buttons, menus, etc.) differently than for "ordinary" elements.	thin - a thin border. medium - a medium border. thick - a thick border.	thick { border-style: dashed; border-width: medium; border-color: lime; }
<b>border-right-width</b>		<length> - the border's thickness has an explicit value. Explicit border widths cannot be	

<b>border-width</b>	The border color properties specify the color of a box's border	<code>&lt;color&gt;   transparent   inherit</code>	<code>negative</code>	browsers!
<b>border-top-color</b> <b>border-right-color</b> <b>border-left-color</b> <b>border-bottom-color</b>				
<b>border-style</b>	The border style properties specify the line style of a box's border (solid, double, dashed, etc.)	<code>none</code> - no border; the border width is zero. <code>hidden</code> - same as 'none', except in terms of border conflict resolution for table elements. <code>dotted</code> - the border is a series of dots. <code>dashed</code> - the border is a series of short line segments.	<code>none</code> - no border; the border width is zero. <code>hidden</code> - same as 'none', except in terms of border conflict resolution for table elements. <code>dotted</code> - the border is a series of dots. <code>dashed</code> - the border is a series of short line segments.	
<b>border-right-style</b>				
<b>border-bottom-style</b>				
<b>border-left-style</b>				
<b>border-style</b>				
<b>border-top</b> <b>border-right</b> <b>border-bottom</b> <b>border-left</b> <b>border</b>	This is a shorthand property for setting the width, style, and color of the top, right, bottom, and left border of a box.  Shorthand property for setting the same width, color, and style for all four borders of a box.	<code>[-border-width&gt;   &lt;border-style&gt;   &lt;border-top-color&gt;]   inherit</code>  <code>[-border-width&gt;   &lt;border-style&gt;   &lt;border-top-color&gt;]   inherit</code>	<code>#fourth { color: red; background: #ffa500; padding-top: 0.1em; padding-bottom: 2em; padding-left: 10em; padding-right: 1em; border-top-style: dotted; border-bottom-style: solid; border-left-style: double; border-right-style: groove; border-top-width: thin; border-bottom-width: thick; border-left-width: medium; border-right-width: medium; border-top-color: maroon; border-bottom-color: aqua; border-left-color: fuchsia; border-right-color: red; }</code>  <b>html document fragment:</b> <code>&lt;ul&gt; &lt;li&gt;first box&lt;/li&gt; &lt;li&gt;second box&lt;/li&gt; &lt;li id="third"&gt;third box (with border)&lt;/li&gt; &lt;li&gt;div id="fourth"&gt;fourth box &lt;br /&gt;(strange) &lt;/div&gt;&lt;/li&gt; &lt;li&gt;Attention! Some properties is not supported some browsers&lt;/li&gt; &lt;/ul&gt;</code>	

[More and details about box model >>>](#)

**Show boxes (Controlling box generation)**

<b>display</b>	The values of this property have the different meanings	<code>inline   block   list-item   run-in   inline-block   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   none   inherit</code>		
		<code>block</code> - this value causes an element to generate a block box	<b>CSS fragment:</b> <code>em { display: block }</code> <b>HTML fragment:</b> <code>&lt;em&gt;First block&lt;/em&gt;&lt;em&gt;Second block&lt;/em&gt;</code>	<i>First block</i> <i>Second block</i>
		<code>inline-block</code> - this value causes an element to generate a block box, which itself is flowed as a single inline box, similar to a replaced element. The inside of an inline-block is formatted as a block box, and the element itself is formatted as an inline replaced element	<b>CSS fragment:</b> <code>em { display: inline-block }</code> <b>HTML fragment:</b> <code>&lt;p&gt;First &lt;em&gt;element&lt;/em&gt; box&lt;/p&gt; &lt;p&gt;Second box&lt;/p&gt;</code>	<i>First element block box</i> <i>Second block</i>
		<code>inline</code> - this value causes an element to generate one or more inline boxes	<b>CSS fragment:</b> <code>p { display: inline }</code> <b>HTML fragment:</b> <code>&lt;p&gt;First&lt;/p&gt; &lt;p&gt;Second&lt;/p&gt;</code>	<i>First Second</i>
		<code>list-item</code> - this value causes an element (e.g., LI in HTML) to generate a principal block box and a list-item inline box. For information about lists and examples of list formatting, please consult the section on lists.	<b>CSS fragment:</b> <code>span { display: list-item }</code> <b>HTML fragment:</b> <code>&lt;p&gt;First&lt;/p&gt; &lt;p&gt;Second&lt;/p&gt;</code>	<i>• First</i> <i>• Second</i>
		<code>none</code> - this value causes an element to generate no boxes in the formatting structure (i.e., the element has no effect on layout). Descendant elements do not generate any boxes either; this behavior cannot be overridden by setting the 'display' property on the descendants	<b>CSS fragment:</b> <code>h3 { display: none }</code> <b>HTML fragment:</b> <code>&lt;h3&gt;First (hidden)&lt;/h3&gt; &lt;p&gt;Second&lt;/p&gt;</code>	<i>Second</i>
		<code>run-in</code> - this value creates either block or inline boxes, depending on context. Properties apply to run-in boxes based on their final status (inline-level or block-level).	<b>CSS fragment:</b> <code>h3 { display: run-in }</code> <b>HTML fragment:</b> <code>&lt;h3&gt;A run-in heading&lt;/h3&gt; &lt;p&gt;And a paragraph of text that follows it.&lt;/p&gt;</code>	<i>A run-in heading.</i> <i>And a paragraph of text that follows it.</i>
		<code>table</code> , <code>inline-table</code> , <code>table-row-group</code> , <code>table-column</code> , <code>table-column-group</code> , <code>table-header-group</code> , <code>table-footer-group</code> , <code>table-row</code> , <code>table-cell</code> , and <code>table-caption</code> - these values cause an element to behave like a table element		

[More and details about show boxes \(controlling box generation\) >>>](#)

**Visual superposition of boxes (Positioning schemes)**

<b>position</b>	The values of this property have the different meanings	<code>static   relative   absolute   fixed   inherit</code>		
		<code>static</code> - the box is a normal box, laid out according to the normal flow. The 'top', 'right', 'bottom', and 'left' properties do not apply.	<b>CSS fragment for all examples:</b> <code>body { display: block; line-height: 200%; width: 400px; height: 150px }</code> <code>p { display: block   span { display: inline }</code> <b>CSS fragment:</b> <code>#outer { position: static; color: red } #inner { position: static; color: blue; background-color: #FFFF99 }</code> <b>HTML fragment:</b> <code>&lt;p&gt;Beginning of body contents. &lt;span class="outer"&gt;Start of outer contents. &lt;span class="inner"&gt;Inner contents. &lt;/span&gt;End of outer contents. &lt;/span&gt;End of body contents. &lt;/p&gt;</code>	<i>Beginning of body contents. Start of outer contents. Inner contents. End of outer contents. End of body contents.</i>
		<code>relative</code> - the box's position is calculated according to the normal flow (this is called the position in normal flow). Then the box is offset relative to its normal position. When a box B is relatively positioned, the position of the following box is calculated as though B were not offset. The effect of 'position: relative' on table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, and table-caption elements is undefined.	<b>CSS fragment:</b> <code>#outer { position: relative; top: -12px; color: red } #inner { position: relative; top: 12px; color: blue; background-color: #FFFF99 }</code> <b>HTML fragment:</b> <code>&lt;p&gt;Beginning of body contents. &lt;span class="outer"&gt;Start of outer contents. &lt;span class="inner"&gt;Inner contents. &lt;/span&gt;End of outer contents. &lt;/span&gt;End of body contents. &lt;/p&gt;</code>	<i>Beginning of body contents. Start of outer contents. Inner contents. End of outer contents. End of body contents.</i>
		<code>absolute</code> - the box's position (and possibly size) is specified with the 'top', 'right', 'bottom', and 'left' properties. These properties specify offsets with respect to the box's containing block. Absolutely positioned boxes are taken out of the normal flow. This means they have no impact on the layout of later siblings. Also, though absolutely positioned boxes have margins, they do not collapse with any other margins.	<b>Example N61</b> <b>Example N62</b> <b>Example N63</b> <b>Example N64 (Implement change bars)</b>	
		<code>fixed</code> - the box's position is calculated according to the 'absolute' model, but in addition, the box is fixed with respect to some reference. As with the 'absolute' model, the box's margins do not collapse with any other margins. In the case of handheld, projection, screen, tv, and tv media types, the box is fixed with respect to the viewport and doesn't move when scrolled. In the case of the print media type, the box is rendered on every page, and is fixed with respect to the page box, even if the page is seen through a viewport (in the case of a print-preview, for example). For other media types, the presentation is undefined. Authors may wish to specify 'fixed' in a media-dependent way. For instance, an author may want a box to remain at the top of the viewport on the screen, but not at the top of each printed page.	<b>Example</b>	
<b>top</b>	Specifies how far an absolutely positioned box's top margin edge is offset below the top edge of the box's containing block	<code>&lt;length&gt;   &lt;percentage&gt;   auto   inherit</code>	<code>div a8 { position: relative; direction: ltr; top: -1em; right: auto }</code>	
<b>right</b>	Specifies how far a box's right margin edge is offset to the left of the right edge of the box's containing block	<code>&lt;length&gt;   &lt;percentage&gt;   auto   inherit</code>	<code>div a8 { position: relative; direction: ltr; left: -1em; right: auto }</code>	
<b>bottom</b>	Specifies how far a box's bottom margin edge is offset above the bottom of the box's containing block	<code>&lt;length&gt;   &lt;percentage&gt;   auto   inherit</code>	<code>div a8 { position: relative; direction: ltr; bottom: -1em; right: auto }</code>	
<b>left</b>	Specifies how far a box's left margin edge is offset to the right of the left edge of the box's containing block	<code>&lt;length&gt;   &lt;percentage&gt;   auto   inherit</code>	<code>div a8 { position: relative; direction: ltr; left: -1em; right: auto }</code>	
<b>float</b>	This property specifies whether a box should float to the left, right, or not at all. It may be set for any element, but only applies to elements that generate boxes that are not absolutely positioned.	<code>left   right   none   inherit</code> <code>none</code> - the box is not floated.	<b>CSS fragment:</b> <code>#outer { color: red } #inner { float: none; width: 130px; color: blue; background-color: #FFFF99 } #sibling { color: maroon }</code> <b>HTML fragment:</b> <code>&lt;p&gt;Beginning of body contents. &lt;span class="outer"&gt;Start of outer contents. &lt;span class="inner"&gt;Inner contents. &lt;/span&gt; &lt;span class="sibling"&gt;Sibling contents. &lt;/span&gt;End of outer contents. &lt;/span&gt;End of body contents. &lt;/p&gt;</code>	<i>Beginning of body contents. Start of outer contents. Inner contents. Sibling contents. End of outer contents. End of body contents.</i>
		<code>left</code> - the element generates a block box that is floated to the left. Content flows on the right side of the box, starting at the top (subject to the 'clear' property).	<b>CSS fragment:</b> <code>#outer { color: red } #inner { float: left; width: 130px; color: blue; background-color: #FFFF99 } #sibling { color: maroon }</code> <b>HTML fragment:</b> <code>&lt;p&gt;Beginning of body contents. &lt;span class="outer"&gt;Start of outer contents. &lt;span class="inner"&gt;Inner contents. &lt;/span&gt; &lt;span class="sibling"&gt;Sibling contents. &lt;/span&gt;End of outer contents. &lt;/span&gt;End of body contents. &lt;/p&gt;</code>	<i>Beginning of body contents. Start of outer contents. Sibling contents. Inner contents. End of outer contents. End of body contents.</i>
		<code>right</code> - similar to 'left', except the box is floated to the right, and content flows on the left side of the box, starting at the top.	<b>CSS fragment:</b> <code>#outer { color: red } #inner { float: right; width: 130px; color: blue; background-color: #FFFF99 } #sibling { color: maroon }</code> <b>HTML fragment:</b> <code>&lt;p&gt;Beginning of body contents. &lt;span class="outer"&gt;Start of outer contents. &lt;span class="inner"&gt;Inner contents. &lt;/span&gt; &lt;span class="sibling"&gt;Sibling contents. &lt;/span&gt;End of outer contents. &lt;/span&gt;End of body contents. &lt;/p&gt;</code>	<i>Beginning of body contents. Start of outer contents. Sibling contents. End of outer contents. End of Inner contents. body contents.</i>
<b>clear</b>	Indicates which sides of an element's box(es) may not be adjacent to an earlier floating box. The 'clear' property does not consider floats inside the element itself or in other block	<code>none   left   right   both   inherit</code> <code>left</code> - the clearance of the generated box is set to the amount necessary to place the top border edge below the bottom outer edge of any left-floating boxes that resulted from	<code>p { clear: left }</code>	

	formatting contexts.	elements earlier in the source document.
	right - the clearance of the generated box is set to the amount necessary to place the top border edge below the bottom outer edge of any right-floating boxes that resulted from elements earlier in the source document.	
	both - the clearance of the generated box is set to the amount necessary to place the top border edge below the bottom outer edge of any right-floating and left-floating boxes that resulted from elements earlier in the source document.	
	none - no constraint on the box's position with respect to floats.	
<b>z-index</b>	'z-index' property specifies the stack level of the box in the current stacking context and whether the box establishes a local stacking context.	<pre>auto   &lt;integer&gt;   inherit</pre> <p>&lt;integer&gt; - this integer is the stack level of the generated box in the current stacking context. The box also establishes a local stacking context in which its stack level is '0'.</p> <p>auto - the stack level of the generated box in the current stacking context is the same as its parent's box. The box does not establish a new local stacking context.</p>

	img { clear: right }
	div { clear: both }
	em { clear: none }
	<a href="#">Example</a>

[More and details about visual superposition of boxes \(positioning schemes\) >>>](#)

### Visual effects

<b>overflow</b>	Specifies whether content of a block-level element is clipped when it overflows the element's box	<pre>visible   hidden   scroll   auto   inherit</pre> <p>visible - indicates that content is not clipped, i.e., it may be rendered outside the block box</p> <p>hidden - indicates that the content is clipped and that no scrolling user interface should be provided to view the content outside the clipping region.</p> <p>scroll - indicates that the content is clipped and that if the user agent uses a scrolling mechanism that is visible on the screen (such as a scroll bar or a panner), that mechanism should be displayed for a box whether or not any of its content is clipped. This avoids any problem with scrollbars appearing and disappearing in a dynamic environment. When this value is specified and the target medium is 'print', overflowing content may be printed.</p> <p>auto - the behavior of the 'auto' value is user agent-dependent, but should cause a scrolling mechanism to be provided for overflowing boxes</p>
<b>clip</b>	Applies only to absolutely positioned elements	<pre>&lt;shape&gt;   auto   inherit</pre> <p>In CSS 2.1, the only valid &lt;shape&gt; value is: rect(&lt;top&gt;, &lt;right&gt;, &lt;bottom&gt;, &lt;left&gt;) where &lt;top&gt; and &lt;bottom&gt; specify offsets from the top border edge of the box, and &lt;right&gt;, and &lt;left&gt; specify offsets from the left border edge of the box in left-to-right text and from the right border edge of the box in right-to-left text.</p>
<b>visibility</b>	The 'visibility' property specifies whether the boxes generated by an element are rendered. Invisible boxes still affect layout (set the 'display' property to 'none' to suppress box generation altogether).	<pre>visible   hidden   collapse   inherit</pre> <p>visible - the generated box is visible</p> <p>hidden - the generated box is invisible (fully transparent, nothing is drawn), but still affects layout. Furthermore, descendants of the element will be visible if they have 'visibility: visible'.</p> <p>collapse - please consult the section on dynamic row and column effects in tables. If used on elements other than rows, row groups, columns, or column groups, 'collapse' has the same meaning as 'hidden'.</p>

**HTML fragment for all examples:**

```
<div>
<blockquote>
<p>I didn't like the play, but then I saw it under adverse conditions - the curtain was up.</p>
</blockquote>
</div>
```

**CSS fragment:**

```
div { overflow: visible; width: 100px; height: 100px; border: thin solid red; }
```

blockquote { width: 125px; height: 100px; margin-top: 50px; margin-left: 50px; border: thin dashed black }

```
cite { display: block; text-align: right; border: none }
```

```
div { overflow: hidden; width: 100px; height: 100px; border: thin solid red; }
```

blockquote { width: 125px; height: 100px; margin-top: 50px; margin-left: 50px; border: thin dashed black }

```
cite { display: block; text-align: right; border: none }
```

```
div { overflow: scroll; width: 100px; height: 100px; border: thin solid red; }
```

blockquote { width: 125px; height: 100px; margin-top: 50px; margin-left: 50px; border: thin dashed black }

```
cite { display: block; text-align: right; border: none }
```

**CSS fragment:**

```
div { overflow: auto; width: 100px; height: 100px; border: thin solid red; }
```

blockquote { width: 125px; height: 100px; margin-top: 50px; margin-left: 50px; border: thin dashed black }

```
cite { display: block; text-align: right; border: none }
```

```
p { clip: rect(5px, 40px, 45px, 5px); }
```

[Example](#)

[More and details about visual effects >>>](#)

### Colors and Background

<b>color</b>	The foreground color of an element's text content	<pre>maroon (#800000)</pre>
<b>background</b>	Sets the background color of an element	<pre>red (#ff0000; #f00; rgb(255,0,0); rgb(100%, 0%, 0%)</pre>
<b>background-color</b>	Sets the background color of an element	<pre>orange (#ffa500)</pre>
		<pre>yellow (#ffff00; #ff0; rgb(255,255,0); rgb(100%, 100%, 0%)</pre>
		<pre>olive (#808000)</pre>
		<pre>purple (#800080)</pre>
		<pre>fuchsia (#ff00ff; #f0f; rgb(255,0,255); rgb(100%, 0%, 100%))</pre>
		<pre>white (#ffffff; #fff; rgb(255,255,255); rgb(100%, 100%, 100%))</pre>
		<pre>lime (#00ff00; #0f0; rgb(0,255,0); rgb(0%, 100%, 0%))</pre>
		<pre>green (#008000)</pre>
		<pre>navy (#000080)</pre>
		<pre>blue (#0000ff; #00f; rgb(0,0,255); rgb(0%, 0%, 100%))</pre>
		<pre>aqua (#00ffff; #0ff; rgb(0,255,255); rgb(0%, 0%, 100%))</pre>
		<pre>teal (#008080)</pre>
		<pre>black (#000000; #000; rgb(0,0,0); rgb(0%, 0%, 0%))</pre>
		<pre>silver (#c0c0c0)</pre>
		<pre>gray (#808080)</pre>
<b>background-image</b>	Sets the background image of an element	<pre>&lt;uri&gt;   none   inherit</pre>
<b>background-repeat</b>	Specifies whether the image is repeated (tiled), and how	<pre>repeat   repeat-x   repeat-y   no-repeat   inherit</pre>
<b>background-attachment</b>	Specifies whether it is fixed with regard to the viewport ('fixed') or scrolls along with the containing block ('scroll')	<pre>scroll   fixed   inherit</pre>
<b>background-position</b>	Specified, this property specifies its initial position.	<pre>[[&lt;percentage&gt;   &lt;length&gt;]   left   center   right]   [&lt;percentage&gt;   &lt;length&gt;]   top   center   bottom   ? ]   [ [ left   center   right ]   [ top   center   bottom ] ]   inherit</pre>
<b>background</b>	The 'background' property is a shorthand property for setting the individual background properties	<pre>[&lt;background-color&gt;   &lt;background-image&gt;   &lt;background-repeat&gt;   &lt;background-attachment&gt;   &lt;background-position&gt; ]   inherit</pre>

```
span { color: red }
```

```
em { color: #ff0000 }
```

```
h5 { color: #f00 }
```

```
p { color: rgb(255,0,0) }
```

```
div { color: rgb(100%, 0%, 0%) }
```

```
body { color: navy }
```

```
div { color: #505050 }
```

```
h1 { background-color: #f00 }
```

```
body { color: black; background: white }
```

```
body { color: white; background: black }
```

**css file:**

```
body { background: aqua; color: blue }
```

```
p { color: red }
```

```
span { color: #ffa500 }
```

**html file:**

```
Example: <p>This is a <span>color example</span></p>
```

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

```
body { background: white url("pendant.png"); background-repeat: repeat-y; background-position: center; }
```

```
body { background: red url("pendant.png"); background-repeat: repeat-y; background-attachment: fixed; }
```

```
body { background: url("banner.jpeg") right top }
```

```
p { background: url("chess.png") gray 50% repeat fixed }
```

[Example:](#)

This is a color example

[More and details about colors and background >>> or >>>](#)

### Fonts

<b>font-family</b>	Prioritized list of font family names and/or generic family names	<pre>[ [&lt;family-name&gt;   &lt;generic-family&gt; ] *, [&lt;family-name&gt;   &lt;generic-family&gt; ]* ]   inherit</pre> <p>'serif' (e.g. Times)</p> <p>'sans-serif' (e.g. Helvetica)</p> <p>'cursive' (e.g. Zapf-Chancery)</p> <p>'fantasy' (e.g. Western)</p> <p>'monospace' (e.g. Courier)</p>
<b>font-style</b>	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	<pre>normal</pre> <p><i>italic</i></p> <p><i>oblique</i></p>
<b>font-variant</b>	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	<pre>normal</pre> <p>SMALL-CAPS</p>
<b>font-weight</b>	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	<pre>normal</pre> <p><b>bold</b></p> <p><b>bolder</b></p> <p>lighter</p> <p>100</p> <p>200</p> <p>300</p> <p>400 (eq normal)</p> <p>500</p> <p><b>600</b></p> <p><b>700 (eq bold)</b></p> <p><b>800</b></p> <p><b>900</b></p>
<b>font-size</b>	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	<pre>&lt;absolute-size&gt;   &lt;relative-size&gt;   &lt;length&gt;   &lt;percentage&gt;   inherit</pre> <pre>&lt;absolute-size&gt;: [ xx-small   x-small   small   medium   large   x-large   xx-large ]</pre>

```
body { font-family: Gill, Helvetica, sans-serif }
```

```
em { font-family: serif }
```

```
p { font-family: sans-serif }
```

```
o { font-family: cursive }
```

```
li { font-family: fantasy }
```

```
ul { font-family: monospace }
```

```
h5 { font-style: normal }
```

```
ol { font-style: italic }
```

```
h3 { font-style: oblique }
```

```
div { font-variant: normal }
```

```
div { font-variant: small-caps }
```

```
div { font-weight: normal }
```

```
div { font-weight: bold }
```

```
div { font-weight: bolder }
```

```
div { font-weight: lighter }
```

```
div { font-weight: 100 }
```

```
div { font-weight: 200 }
```

```
div { font-weight: 300 }
```

```
div { font-weight: 400 }
```

```
div { font-weight: 500 }
```

```
div { font-weight: 600 }
```

```
div { font-weight: 700 }
```

```
div { font-weight: 800 }
```

```
div { font-weight: 900 }
```

```
h6 { font-size: xx-small }
```

```
em { font-size: x-small }
```

```
h5 { font-size: small }
```

```
h4 { font-size: medium }
```

```
h3 { font-size: large }
```

```
h2 { font-size: x-large }
```

```
h1 { font-size: xx-large }
```

```
blockquote { font-size: larger }
```

```
p { font-size: 16px; }
```

```
@media print { p { font-size: 12pt; }
```

<p><b>font</b></p>	<p>The 'font' property is, except as described below, a shorthand property for setting 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' and 'font-family' at the same place in the style sheet. The syntax of this property is based on a traditional typographical shorthand notation to set multiple properties related to fonts.</p>	<p>&lt;percentage&gt;: [ em   % ]                  * see <a href="#">units</a></p> <p>[[ &lt;font-style&gt; &lt;font-variant&gt; &lt;font-weight&gt; &lt;font-size&gt; &lt;line-height&gt; &lt;font-family&gt; ]                  [ caption   icon   menu   message-box   small-caption   status-bar   inherit</p> <p>caption - The font used for captioned controls (e.g., buttons, drop-downs, etc.).                  icon - The font used to label icons.                  menu - The font used in menus (e.g., dropdown menus and menu lists).                  message-box - The font used in dialog boxes.                  small-caption - The font used for labeling small controls.                  status-bar - The font used in window status bars.</p>	<p>em { font-size: 1.5em }                  em { font-size: 150% }                  p { font: 12px/14px sans-serif }                  p { font: 80% sans-serif }</p> <p><b>p { font: x-large/110% "New Century Schoolbook", serif }</b>                  p { font: bold italic large Palatino, serif }                  P { FONT: NORMAL SMALL-CAPS 120%/120% FANTASY }</p> <p>span { font: caption }                  span { font: icon }                  span { font: menu }                  span { font: message-box }                  span { font: small-caption }                  span { font: status-bar }</p>	<p>More and details about fonts &gt;&gt;&gt;</p>
<p><b>Text</b></p>				
<p><b>text-indent</b></p>	<p>Specifies the indentation of the first line of text in a block</p>	<p>&lt;length&gt;   &lt;percentage&gt;   inherit</p>	<p>p { text-indent: 16px }                  div { text-indent: 3em }</p>	
<p><b>text-align</b></p>	<p>Describes how inline content of a block is aligned</p>	<p>left   right   center   justify   inherit</p>	<p>p { text-align: left }</p>	<p>p { text-align: right }</p>
<p><b>vertical-align</b></p>	<p>Affects the vertical positioning inside a line box of the boxes generated by an inline-level element</p>	<p>baseline   sub   super   top   text-top   middle   bottom   text-bottom   &lt;percentage&gt;   &lt;length&gt;   inherit</p> <p>baseline - align the baseline of the box with the baseline of the parent box. If the box doesn't have a baseline, align the bottom margin edge with the parent's baseline.                  middle - align the vertical midpoint of the box with the baseline of the parent box plus half the x-height of the parent.                  sub - lower the baseline of the box to the proper position for subscripts of the parent's box. (This value has no effect on the font size of the element's text.)                  super - raise the baseline of the box to the proper position for superscripts of the parent's box. (This value has no effect on the font size of the element's text.)                  text-top - align the top of the box with the top of the parent's content area                  text-bottom - align the bottom of the box with the bottom of the parent's content area                  &lt;percentage&gt; - raise (positive value) or lower (negative value) the box by this distance (a percentage of the 'line-height' value). The value '0%' means the same as 'baseline'.                  &lt;length&gt; - raise (positive value) or lower (negative value) the box by this distance. The value '0cm' means the same as 'baseline'.                  top - align the top of the aligned subtree with the top of the line box.                  bottom - align the bottom of the aligned subtree with the bottom of the line box.</p>	<p>p { text-align: justify }                  div { }</p> <p>div { vertical-align: baseline }                  div { vertical-align: middle }                  div { vertical-align: sub }                  div { vertical-align: super }                  div { vertical-align: text-top }                  div { vertical-align: text-bottom }                  div { vertical-align: -20% }                  div { vertical-align: 15px }                  div { vertical-align: top }                  div { vertical-align: bottom }</p>	
<p><b>text-decoration</b></p>	<p>Describes decorations that are added to the text of an element using the element's color</p>	<p>none   [ underline    overline    line-through    blink ]   inherit                  none - produces no text decoration                  underline - each line of text is underlined                  overline - each line of text has a line above it                  line-through - each line of text has a line through the middle                  blink - text blinks (alternates between visible and invisible)</p>	<p><b>stylesheet fragment:</b>                  blockquote { text-decoration: underline; color: blue; }                  em { display: block; }                  cite { color: fuchsia; }  <b>html document fragment:</b>                  &lt;blockquote&gt;                  &lt;p&gt;                  &lt;span&gt;                  Help, help!                  &lt;em&gt;I am under a hat!&lt;/em&gt;                  &lt;cite&gt;—GwieF&lt;/cite&gt;                  &lt;/span&gt;&lt;/p&gt;                  &lt;/blockquote&gt;</p>	<p><a href="#">Help, help!</a>  <a href="#">I am under a hat!</a>  <a href="#">GwieF</a></p>
<p><b>letter-spacing</b></p>	<p>Specifies spacing behavior between text characters</p>	<p>normal   &lt;length&gt;   inherit</p>	<p>blockquote { letter-spacing: 0.1em }</p>	
<p><b>word-spacing</b></p>	<p>Specifies spacing behavior between words</p>	<p>normal   &lt;length&gt;   inherit</p>	<p>h1 { word-spacing: 1em }</p>	
<p><b>line-height</b></p>	<p>specifies the minimal height of line boxes within the element</p>	<p>normal   &lt;number&gt;   &lt;length&gt;   &lt;percentage&gt;   inherit</p> <p>&lt;length&gt; - the specified length is used in the calculation of the line box height. Negative values are illegal.                  &lt;number&gt; - the used value of the property is this number multiplied by the element's font size. Negative values are illegal. The computed value is the same as the specified                  &lt;percentage&gt; - the computed value of the property is this percentage multiplied by the element's computed font size. Negative values are illegal.</p>	<p>h1 { line-height: normal }                  /* normal */                  div { line-height: 1.2em }                  /* length */                  div { line-height: 1.2 }                  /* number */                  div { line-height: 55% }                  /* percentage */</p>	
<p><b>text-transform</b></p>	<p>Controls capitalization effects of an element's text</p>	<p>Capitalize - Puts The First Character Of Each Word In Uppercase                  UPPERCASE - PUTS ALL CHARACTERS OF EACH WORD IN UPPERCASE                  lowercase - puts all characters of each word in lowercase                  none - no capitalization effects</p>	<p>P { Text-transform: Capitalize }                  P { TEXT-TRANSFORM: UPPERCASE }                  p { text-transform: lowercase }                  p { text-transform: none }</p>	
<p><b>white-space</b></p>	<p>Directs user agents to collapse sequences of whitespace, and break lines as necessary to fill line boxes</p>	<p>normal   pre   nowrap   pre-wrap   pre-line   inherit                  normal - directs user agents to collapse sequences of whitespace, and break lines as necessary to fill line boxes.                  pre - prevents user agents from collapsing sequences of whitespace. Lines are only broken at newlines in the source, or at occurrences of "A" in generated content                  nowrap - collapses whitespace as for 'normal', but suppresses line breaks within text                  pre-wrap - reverts user agents from collapsing sequences of whitespace. Lines are broken at newlines in the source, at occurrences of "A" in generated content, and as necessary to fill line boxes                  pre-line - directs user agents to collapse sequences of whitespace. Lines are broken at newlines in the source, at occurrences of "A" in generated content, and as necessary to fill line boxes</p>	<p>p { white-space: normal }</p> <p>pre { white-space: pre }                  td[nowrap] { white-space: nowrap }                  pre[wrap] { white-space: pre-wrap }                  :before,after { white-space: pre-line }</p>	
<p><b>direction</b></p>	<p>Specifies the base writing direction of blocks and the direction of embeddings and overrides (see 'unicode-bidi') for the Unicode bidirectional algorithm</p>	<p>ltr   rtl   inherit                  ltr - left-to-right direction.                  rtl - right-to-left direction.</p>	<p><b>XML fragment:</b>                  &lt;hebrew&gt;                  &lt;par&gt;HEBREW1 HEBREW2 english3 HEBREW4 HEBREW5&lt;/par&gt;                  &lt;par&gt;HEBREW6 &lt;emph&gt;HEBREW7&lt;/emph&gt; HEBREW8&lt;/par&gt;                  &lt;/hebrew&gt;                  &lt;english&gt;                  &lt;par&gt;english9 english10 english11 HEBREW12 HEBREW13&lt;/par&gt;                  &lt;par&gt;english14 english15 english16&lt;/par&gt;                  &lt;par&gt;english17 &lt;he-quo&gt;HEBREW18 english19 HEBREW20&lt;/he-quo&lt;/par&gt;                  &lt;/english&gt;</p>	
<p><b>unicode-bidi</b></p>	<p>Values for this property have the different meanings</p>	<p>normal   embed   bidi-override   inherit</p> <p>normal - the element does not open an additional level of embedding with respect to the bidirectional algorithm. For inline-level elements, implicit reordering works across element boundaries.                  embed - if the element is inline-level, this value opens an additional level of embedding with respect to the bidirectional algorithm. The direction of this embedding level is given by the 'direction' property. Inside the element, reordering is done implicitly.                  bidi-override - for inline-level elements this creates an override. For block-level, table-cell, table-caption, or inline-block elements this creates an override for inline-level descendants not within another block-level, table-cell, table-caption, or inline-block element. This means that inside the element, reordering is strictly in sequence according to the 'direction' property, the implicit part of the bidirectional algorithm is ignored.</p>	<p><b>CSS fragment:</b>                  hebrew, he-quo { direction: rtl; unicode-bidi: embed }                  english { direction: ltr; unicode-bidi: embed }</p> <p><b>CSS fragment:</b>                  hebrew, english, par {display: block }                  emph {font-weight: bold }</p>	<p>HEBREW1 HEBREW2 english3 HEBREW4 HEBREW5 HEBREW6 HEBREW7 HEBREW8 english9 english10 english11 HEBREW12 HEBREW13 english14 english15 english16 english17 HEBREW18 english19 HEBREW20</p> <p>HEBREW1 HEBREW2 english3 HEBREW4 HEBREW5 HEBREW6 HEBREW7 HEBREW8 english9 english10 english11 HEBREW12 HEBREW13 english14 english15 english16 english17 HEBREW18 english19 HEBREW20</p>
<p>More and details about text &gt;&gt;&gt; about visual formatting model ('width', 'height', 'line-height' and 'vertical-align' properties) &gt;&gt;&gt; about direction ('direction' and 'unicode-bidi' properties) &gt;&gt;&gt;</p>				
<p><b>Generated content</b></p>				
<p><b>content</b></p>	<p>This property is used with the :before and :after pseudo-elements to generate content in a document.</p>	<p>normal   none   [ &lt;string&gt;   &lt;uri&gt;   &lt;counter&gt;   attr(&lt;identifier&gt;)   open-quote   close-quote   no-open-quote   no-close-quote ]*   inherit                  none - the pseudo-element is not generated                  normal - computes to 'none' for the :before and :after pseudo-elements.                  &lt;string&gt; - text content (see the section on strings).</p>	<p>span:before { content: none }                  li:before { content: normal }</p> <p><b>CSS fragment:</b>                  span:before { content: "Chapter: "; }</p> <p><b>HTML fragment:</b>                  &lt;span&gt;this is a chapter&lt;/span&gt;</p> <p><b>CSS fragment:</b>                  &lt;uri&gt; - the value is a URI that designates an external resource (such as an image). If a user agent cannot display the resource it must ignore it.</p> <p><b>HTML fragment:</b>                  &lt;img alt="A picture of a cat" data-bbox="580 375 680 380"/&gt;</p>	<p>Chapter: this is a chapter</p>
<p><b>counter</b></p>	<p>Counters are used to generate page numbers, chapter numbers, etc.</p>	<p>&lt;counter&gt; - counters may be specified with two different functions: 'counter()' or 'counters()'. The former has two forms: 'counter(name)' or 'counter(name, style)'. The generated text is the value of the innermost counter of the given name in scope at this pseudo-element; it is formatted in the indicated style ('decimal' by default). The latter function also has two forms: 'counters(name, string)' or 'counters(name, string, style)'.                  open-quote and close-quote - these values are replaced by the appropriate string from the 'quotes' property.                  no-open-quote and no-close-quote - introduces no content, but increments (decrements) the level of nesting for quotes.                  attr(X) - this function returns as a string the value of attribute X for the subject of the</p>	<p><b>CSS fragment:</b>                  q:before { content: open-quote } q:after { content: close-quote }                  &lt;math&gt;e^{i\pi} = -1&lt;/math&gt;                  &lt;math&gt;e^{i\pi/2} = i&lt;/math&gt;                  &lt;math&gt;e^{i\pi/4} = \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}&lt;/math&gt;                  &lt;math&gt;e^{i3\pi/4} = \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}&lt;/math&gt;                  &lt;math&gt;e^{i\pi} = -1&lt;/math&gt;                  &lt;math&gt;e^{i5\pi/4} = \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}&lt;/math&gt;                  &lt;math&gt;e^{i3\pi/2} = -i&lt;/math&gt;                  &lt;math&gt;e^{i7\pi/4} = \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}&lt;/math&gt;                  &lt;math&gt;e^{i2\pi} = 1&lt;/math&gt;</p>	<p>“Quote me!”                  No quote me!</p>

<p><b>quotes</b></p> <p>This property specifies quotation marks for any number of embedded quotations.</p>	<p>selector. The string is not parsed by the CSS processor. If the subject of the selector doesn't have an attribute X, an empty string is returned. The case-sensitivity of attribute names depends on the document language.</p> <p>[&lt;string&gt; &lt;string&gt;+   none   inherit none - the 'open-quote' and 'close-quote' values of the 'content' property produce no quotation marks. [&lt;string&gt; &lt;string&gt;+ - values for the 'open-quote' and 'close-quote' values of the 'content' property are taken from this list of pairs of quotation marks (opening and closing). The first (leftmost) pair represents the outermost level of quotation, the second pair the first level of embedding, etc.</p>	<p><b>HTML fragment:</b></p> <pre>q:lang(en) { quotes: "" "" "" "" }</pre> <p><b>CSS fragment:</b></p> <pre>q:lang(no) { quotes: "«" "»" "" "" }</pre> <p><b>HTML fragment:</b></p> <pre>&lt;q&gt;Trøndere gråter når &lt;q&gt;Vinsjan på kaia&lt;/q&gt; blir deklamert.&lt;/q&gt;</pre>	<p>“Trøndere gråter når ‘Vinsjan på kaia’ blir deklamert.”</p> <p><a href="#">More and details about generated content &gt;&gt;&gt;</a></p>
<p><b>Automatic counters and numbering</b></p>			
<p><b>counter-increment</b></p> <p>Accepts one or more names of counters (identifiers), each one optionally followed by an integer. The integer indicates by how much the counter is incremented for every occurrence of the element. The default increment is 1. Zero and negative integers are allowed</p>	<p>[&lt;identifier&gt; &lt;integer&gt;?]+   none   inherit</p>	<p><b>CSS fragment:</b></p> <pre>h3:before { content: "Chapter " counter(chapter) ". "; counter-increment: chapter; } h3 { counter-reset: section; } h4:before { content: counter(chapter) ". " counter(section) ". "; counter-increment: section; }</pre>	<p><b>Chapter 1. First chapter</b></p> <p><b>0.1 First section</b></p> <p><b>0.2 Second section</b></p>
<p><b>counter-reset</b></p> <p>contains a list of one or more names of counters, each one optionally followed by an integer. The integer gives the value that the counter is set to on each occurrence of the element. The default is 0.</p>	<p>[&lt;identifier&gt; &lt;integer&gt;?]+   none   inherit</p>	<p><b>HTML fragment:</b></p> <pre>&lt;h3&gt;First chapter&lt;/h3&gt; &lt;h4&gt;First section&lt;/h4&gt; &lt;h4&gt;Second section&lt;/h4&gt; &lt;h3&gt;Second chapter&lt;/h3&gt; &lt;h4&gt;First section&lt;/h4&gt; &lt;h4&gt;Second section&lt;/h4&gt;</pre>	<p><b>Chapter 1. Second chapter</b></p> <p><b>0.1 First section</b></p> <p><b>0.2 Second section</b></p> <p><a href="#">More and details about automatic counters and numbering &gt;&gt;&gt;</a></p>
<p><b>Lists</b></p>			
<p><b>list-style-type</b></p> <p>Specifies appearance of the list item marker if 'list-style-image' has the value 'none' or if the image pointed to by the URI cannot be displayed. The value 'none' specifies no marker, otherwise there are three types of marker: glyphs, numbering systems, and alphabetic systems.</p> <p>disc   circle   square   decimal   decimal-leading-zero   lower-roman   upper-roman   lower-greek   lower-latin   upper-latin   armenian   georgian   lower-alpha   upper-alpha   none   none - rendering depends on the user agent.</p> <p>circle - rendering depends on the user agent.</p> <p>square - rendering depends on the user agent.</p> <p>decimal - decimal numbers, beginning with 1.</p> <p>decimal-leading-zero - decimal numbers padded by initial zeros (e.g., 01, 02, 03, ..., 98, 99).</p> <p>lower-roman - lowercase roman numerals (i, ii, iii, iv, v, etc.).</p> <p>upper-roman - uppercase roman numerals (I, II, III, IV, V, etc.).</p> <p>georgian - traditional Georgian numbering (an, ban, gan, ..., he, tan, in, in-an, ...).</p> <p>armenian - traditional Armenian numbering</p> <p>lower-latin or lower-alpha - lowercase ascii letters (a, b, c, ... z).</p> <p>upper-latin or upper-alpha - uppercase ascii letters (A, B, C, ... Z).</p> <p>lower-greek - lowercase classical Greek alpha, beta, gamma, ... (α, β, γ, ...)</p>	<p>disc   circle   square   decimal   decimal-leading-zero   lower-roman   upper-roman   lower-greek   lower-latin   upper-latin   armenian   georgian   lower-alpha   upper-alpha   none   none - rendering depends on the user agent.</p> <p>circle - rendering depends on the user agent.</p> <p>square - rendering depends on the user agent.</p> <p>decimal - decimal numbers, beginning with 1.</p> <p>decimal-leading-zero - decimal numbers padded by initial zeros (e.g., 01, 02, 03, ..., 98, 99).</p> <p>lower-roman - lowercase roman numerals (i, ii, iii, iv, v, etc.).</p> <p>upper-roman - uppercase roman numerals (I, II, III, IV, V, etc.).</p> <p>georgian - traditional Georgian numbering (an, ban, gan, ..., he, tan, in, in-an, ...).</p> <p>armenian - traditional Armenian numbering</p> <p>lower-latin or lower-alpha - lowercase ascii letters (a, b, c, ... z).</p> <p>upper-latin or upper-alpha - uppercase ascii letters (A, B, C, ... Z).</p> <p>lower-greek - lowercase classical Greek alpha, beta, gamma, ... (α, β, γ, ...)</p>	<p><b>HTML fragment for all examples:</b></p> <pre>&lt;ol&gt; &lt;li&gt;First&lt;/li&gt; &lt;li&gt;Second&lt;/li&gt; &lt;li&gt;Third&lt;/li&gt; &lt;li&gt;Fourth&lt;/li&gt; &lt;/ol&gt;</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: disc; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: circle; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: square; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: decimal; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: decimal-leading-zero; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: lower-roman; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: upper-roman; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: georgian; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: armenian; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: lower-latin; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: upper-alpha; }</pre> <p><b>CSS fragment:</b></p> <pre>ol { list-style-type: lower-greek; }</pre>	<p>• First • Second • Third • Fourth</p> <p>○ First ○ Second ○ Third ○ Fourth</p> <p>■ First ■ Second ■ Third ■ Fourth</p> <p>1. First 2. Second 3. Third 4. Fourth</p> <p>01. First 02. Second 03. Third 04. Fourth</p> <p>i. First ii. Second iii. Third iv. Fourth</p> <p>I. First II. Second III. Third IV. Fourth</p> <p>ა. First ბ. Second გ. Third დ. Fourth</p> <p>Ա. First Բ. Second Գ. Third Դ. Fourth</p> <p>a. First b. Second c. Third d. Fourth</p> <p>A. First B. Second C. Third D. Fourth</p> <p>α. First β. Second γ. Third δ. Fourth</p>
<p><b>list-style-image</b></p> <p>Sets the image that will be used as the list item marker</p> <p><b>list-style-position</b></p> <p>Specifies the position of the marker box in the principal block box</p> <p>inside   outside   inherit</p> <p>outside - the marker box is outside the principal block box</p> <p>inside - the marker box is the first inline box in the principal block box, after which the element's content flows.</p> <p><b>list-style</b></p> <p>Is a shorthand notation for setting the three properties 'list-style-type', 'list-style-image', and 'list-style-position' at the same place in the style sheet</p>	<p>&lt;uri&gt;   none   inherit</p> <p>inside   outside   inherit</p> <p>outside - the marker box is outside the principal block box</p> <p>inside - the marker box is the first inline box in the principal block box, after which the element's content flows.</p> <p>[&lt;list-style-type&gt;   &lt;list-style-position&gt;   &lt;list-style-image&gt; ]   inherit</p>	<p>ul { list-style-image: url("http://www.iconsfree.org/icon/image/image/20051107/http_www_20051107021654.icons-design.theign.ro_koPainter_images_ellipse.png") }</p> <p><b>CSS fragment:</b></p> <pre>ul { list-style-position: outside; } ul compact { list-style-position: inside; }</pre> <p><b>HTML fragment:</b></p> <pre>&lt;ul&gt; &lt;li&gt;first&lt;/li&gt; &lt;li&gt;second&lt;/li&gt; &lt;/ul&gt;</pre> <pre>&lt;ul class="compact"&gt; &lt;li&gt;first&lt;/li&gt; &lt;li&gt;second&lt;/li&gt; &lt;/ul&gt;</pre> <p>ul &gt; li &gt; ul { list-style: circle outside }</p>	<p>• list item • second list item</p> <p>• first list item • second list item</p>

[More and details about lists >>>](#)

**Tables**

<p><b>caption-side</b></p> <p>Specifies the position of the caption box with respect to the table box.</p>	<p>top   bottom   inherit</p>	<p>caption { caption-side: bottom; width: auto; text-align: left }</p>	
<p><b>table-layout</b></p> <p>Controls the algorithm used to lay out the table cells, rows, and columns.</p>	<p>auto   fixed   inherit</p>	<p>table { table-layout: fixed; margin-left: 2em; margin-right: 2em }</p>	
<p><b>border-collapse</b></p> <p>Selects a table's border model.</p>	<p>collapse   separate   inherit</p>	<p>table { border: outset 10pt; border-collapse: separate; border-spacing: 15pt }</p>	
<p><b>border-spacing</b></p> <p>Selects a table's border model.</p>	<p>&lt;length&gt; &lt;length&gt;?   inherit</p>		
<p><b>empty-cells</b></p> <p>Controls the rendering of borders and backgrounds around cells that have no visible content</p>	<p>show   hide   inherit</p>	<p>table { empty-cells: show }</p>	

[More and details about tables >>>](#)

**Miscellaneous**

