

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](#) [Shadows](#) [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](#):

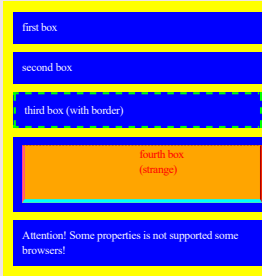
Template: put the style sheet into the document	Template: link to an external style sheet	Syntax
<pre><html lang="en"> <head> <title>... replace with your document's title ...</title> <style type="text/css"> /* CSS Document */ ... replace with your css's content ... </style> </head> <body> ... replace with your document's content ... </body> </html></pre>	<pre>your_document.html: <html 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 ... </body> </html> your_document.css: ... replace with your css's content ...</pre>	<pre>/* Comment */ @import url("fancyfonts.css") media_type; @media media_type { selector { property: values; property: values; } } /* @media media_type is optional */</pre>

Media types (media_type)		Units	Related References
all	For all devices	px	Pixels
braille	For braille tactile feedback devices	em	1 em equal to font size of parent (same as 100%)
embossed	For paged braille printers	ex	Height of lower case "x"
handheld	For handheld devices	%	Percentage
print	For paged material	in	Inches. 1 inch is equal to 2.54 centimeters.
projection	For projected presentations	cm	Centimeters
screen	For color computer screens	mm	Millimeters. 1 millimeter is equal to 1/10nd of an centimeter
speech	For speech synthesizers	pt	Points. 1 pt is equal to 1/72nd of an inch.
ty	For media using a fixed-pitch character grid	pc	Picas. 1 pc is equal to 12 pt
tv	For television-type devices	#789abc	RGB HEX Notation
More and details about media types >>>		#abc	Equates to "#aabbcc"
Selectors		rgb(0,100,255)	Value (0 to 255) of each of red, green and blue
*	All elements (universal selector)	rgb(0%, 50%, 100%)	Value (0% to 100%) of each of red, green and blue
<div>	(type selector)	ms	Milliseconds
<div *	Elements within <div>	s	Seconds
<div span	 within <div> (descendant selector)	Hz	Hertz
<div span	<div> and (grouping selectors)	kHz	Kilohertz
<div > span	 with <div> as parent (child selector)	0	0 requires no unit
<div + span	 preceded by <div>...<div> (adjacent sibling selector)	More and details about lengths >>>	
*.any_class	Elements of class="any_class" (class selector)	Tools	
.any_class		Validators:	
div.any_class	<div class="any_class"> (class selector)	<ul style="list-style-type: none">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 newsfeeds 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) >>	
*#item_id	Element with id="item_id" (aid selector)		
#item_id			
div#item_id	<div id="any_class"> (id selector)		
[any_attr]	Elements with sets any_attr attribute (attribute selector)		
[any_attr]			
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)		

Related themes	
HTML/XHTML (HTML SU)	
XML - eXtensible Markup Language (XML SU)	
DTD - DocType Declaration	
Other Manuals (MANUAL SU)	
Free archive of icons for download (IconsFree.org)	

Box model	
Another TOP boxes	
Another LEFT boxes	<div>Margin</div> <div>Border</div> <div>Padding</div> <div>Content</div>
Another BOTTOM boxes	
More and details about selectors >>>	

Properties		Box	
		Box	
width	Specifies the content width of boxes generated by block-level and replaced elements	<length> <percentage> auto inherit	p { width: 100px }
min-width	These two properties allow authors to constrain content widths to a certain range	<length> <percentage> inherit	h1 { min-width: 10px }
max-width		<length> <percentage> none inherit	div { max-width: 600px }
height	specifies the content height of boxes generated by block-level, inline-block and replaced elements	<length> <percentage> auto inherit	p { height: 150px }
min-height	These two properties allow authors to constrain content widths to a certain range	<length> <percentage> inherit	h1 { min-height: 10px }
max-height		<length> <percentage> none inherit	div { max-height: 600px }
margin-top	Margin properties specify the width of the margin area of a box	<margin-width> inherit	stylesheet fragment:
margin-bottom		Negative values for margin properties are allowed, but there may be implementation-specific limits.	ul {
margin-right			background: yellow;
margin-left			margin: 12px 12px 12px 12px;
margin			padding: 3px 3px 3px 3px;
padding-top	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	}
padding-bottom			li {
padding-right			color: white;
padding-left			background: blue;
padding			margin: 12px 12px 12px 12px;
border-top-width	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.	padding: 12px 0px 12px 12px;
border-right-width		medium - a medium border.	list-style: none
border-bottom-width		thick - a thick border.	}
border-left-width		<length> - the border's thickness has an explicit value. Explicit border widths cannot be negative.	#third {
border-width			border-style: dashed;
border-top-color	The border color properties specify the color of a box's border	<color> transparent inherit	border-width: medium;
border-right-color			border-color: lime;
border-bottom-color			}
border-left-color			#fourth {
			color: red;
			background: #ffa500;
			}



Attention! Some properties is not supported some browsers!

border-color			padding-top: 0.1em; padding-bottom: 2em; padding-left: 10em; padding-right: 1em;	
border-top-style	The border style properties specify the line style of a box's border (solid, double, dashed, etc.)	none - no border; the border width is zero. hidden - same as 'none', except in terms of border conflict resolution for table elements.	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;	
border-right-style		<div> <div></div> <div></div> <div></div> <div></div> </div> dotted - the border is a series of dots. <div> <div> -----</div> <div> -----</div> <div> -----</div> <div> -----</div> </div> dashed - the border is a series of short line segments.		
border-bottom-style		<div> <div> =====</div> <div> =====</div> </div> solid - the border is a single line segment. <div> <div> =====</div> <div> =====</div> </div> double - the border is two solid lines <div> <div> =====</div> <div> =====</div> </div> groove - the border looks as though it were carved into the canvas <div> <div> =====</div> <div> =====</div> </div> ridge - the opposite of 'groove': the border looks as though it were coming out of the canvas <div> <div> =====</div> <div> =====</div> </div> inset - the border makes the box look as though it were embedded in the canvas <div> <div> =====</div> <div> =====</div> </div> outset - the opposite of 'inset': the border makes the box look as though it were coming out of the canvas		
border-left-style				
border-style				
border-top	This is a shorthand property for setting the width, style, and color of the top, right, bottom, and left border of a box.	[-border-width> > -border-style> > -border-top-color>] inherit		
border-right				
border-bottom				
border-left				
border	Shorthand property for setting the same width, color, and style for all four borders of a box.	[-border-width> > -border-style> > -border-top-color>] inherit		
				html document fragment: first box second box <li id="third">third box (with border) <div id="fourth">fourth box (strange)</div> Attention! Some properties is not supported some browsers

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

Show boxes (Controlling box generation)

display	The values of this property have the different meanings	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		
		block - this value causes an element to generate a block box	CSS fragment: em { display: block } HTML fragment: First blockSecond block	<i>First block</i> <i>Second block</i>
		inline-block - 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	CSS fragment: em { display: inline-block } HTML fragment: <p>First element box</p> <p>Second box</p>	<i>First element block block</i> Second block
		inline - this value causes an element to generate one or more inline boxes	CSS fragment: p { display: inline } HTML fragment: <p>First</p> <p>Second</p>	First Second
		list-item - 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.	CSS fragment: span { display: list-item } HTML fragment: First Second	<ul style="list-style-type: none"> First Second
		none - 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	CSS fragment: h3 { display: none } HTML fragment: <h3>First (hidden)</h3> <p>Second</p>	Second
		run-in - 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).	CSS fragment: h3 { display: run-in } HTML fragment: <h3>A run-in heading.</h3> <p>And a paragraph of text that follows it.</p>	A run-in heading. And a paragraph of text that follows it.
		table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell, and table-caption - these values cause an element to behave like a table element		

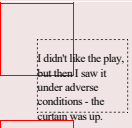
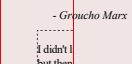


More and details about show boxes (controlling box generation) [>>>](#)

Visual superposition of boxes (Positioning schemes)


position	The values of this property have the different meanings	static relative absolute fixed inherit	CSS fragmant for all examples: body { display: block; line-height: 200%; width: 400px; height: 150px } p { display: block } span { display: inline }	
		static - the box is a normal box, laid out according to the normal flow. The 'top', 'right', 'bottom', and 'left' properties do not apply.	CSS fragment: #outer { position: static; color: red } #inner { position: static; color: blue; background-color: #FFFF99 } HTML fragment: <p>Beginning of body contents. Start of outer contents. Inner contents.End of outer contents.End of body contents. </p>	Beginning of body contents. <i>Start of outer contents.</i> <i>Inner contents.</i> End of outer contents.End of body contents.
		relative - 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.	CSS fragment: #outer { position: relative; top: -12px; color: red } #inner { position: relative; top: 12px; color: blue; background-color: #FFFF99 } HTML fragment: <p>Beginning of body contents. Start of outer contents. Inner contents.End of outer contents.End of body contents. </p>	Beginning of body contents. <i>Start of outer contents.</i> <i>Inner contents.</i> End of outer contents. End of body contents.
		absolute - 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.	Example N#1 Example N#2 Example N#3 Example N#4 (Implement change bars)	
		fixed - 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, tty, 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.	Example	
top	Specifies how far an absolutely positioned box's top margin edge is offset below the top edge of the box's containing block	<length> <percentage> auto inherit	div.a8 { position: relative; direction: ltr; top: -1em; right: auto }	
right	Specifies how far a box's right margin edge is offset to the left of the right edge of the box's containing block	<length> <percentage> auto inherit	div.a8 { position: relative; direction: ltr; left: -1em; right: auto }	
bottom	Specifies how far a box's bottom margin edge is offset above the bottom of the box's containing block	<length> <percentage> auto inherit	div.a8 { position: relative; direction: ltr; bottom: -1em; right: auto }	
left	Specifies how far a box's left margin edge is offset to the right of the left edge of the box's containing block	<length> <percentage> auto inherit	div.a8 { position: relative; direction: ltr; left: -1em; right: auto }	
float	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.	left right none inherit none - the box is not floated.	CSS fragment: #outer { color: red } #inner { float: none; width: 130px; color: blue; background-color: #FFFF99 } #sibling { color: maroon } HTML fragment: <p>Beginning of body contents. Start of outer contents. Inner contents. Sibling contents.End of outer contents.End of body contents. </p>	Beginning of body contents. <i>Start of outer contents.</i> <i>Inner contents.</i> <i>Sibling contents.</i> End of outer contents. End of body contents.
		left - 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).	CSS fragment: #outer { color: red } #inner { float: left; width: 130px; color: blue; background-color: #FFFF99 } #sibling { color: maroon } HTML fragment: <p>Beginning of body contents. Start of outer contents. Inner contents. Sibling contents.End of outer contents.End of body contents. </p>	Beginning of body contents. <i>Start of outer contents.</i> <i>Sibling contents.</i> <i>Inner contents.</i> End of outer contents.End of body contents.

		right - 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.	CSS fragment: #outer { color: red ; #inner { float: right; width: 130px; color: blue; background-color: #FFFFFF99 } #sibling { color: maroon } HTML fragment: <p>Beginning of body contents. Start of outer contents. Inner contents. Sibling contents.End of outer contents.</p>	Beginning of body contents. Start of outer contents. Sibling contents. End of outer contents. End of body Inner contents.
clear	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 formatting contexts.	none left right both inherit left - 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 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.	p { clear: left ; img { clear: right ; div { clear: both ; em { clear: none ;	
z-index	'z-index' property specifies the stack level of the box in the current stacking context and whether the box establishes a local stacking context.	auto <integer> inherit <integer> - 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. 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.	Example	

More and details about visual superposition of boxes (positioning schemes) >>>

Visual effects				
overflow	Specifies whether content of a block-level element is clipped when it overflows the element's box	visible hidden scroll auto inherit visible - indicates that content is not clipped, i.e., it may be rendered outside the block box hidden - indicates that the content is clipped and that no scrolling user interface should be provided to view the content outside the clipping region. 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. auto - the behavior of the 'auto' value is user agent-dependent, but should cause a scrolling mechanism to be provided for overflowing boxes	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> <cite>- Groucho Marx</cite> </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 } CSS fragment: 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 } CSS fragment: 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 }	   
clip	Applies only to absolutely positioned elements	<shape> auto inherit In CSS 2.1, the only valid <shape> value is rect(<top>, <right>, <bottom>, <left>) where <top> and <bottom> specify offsets from the top border edge of the box, and <right>, and <left> 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 { clip: rect(5px, 40px, 45px, 5px); } p { clip: rect(5px, 55px, 45px, 5px); }	
visibility	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).	visible hidden collapse inherit visible - the generated box is visible 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'. 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'.	Example	

More and details about visual effects >>>

Colors and Background				
color	The foreground color of an element's text content	maroon (#800000)	span { color: red ;	
background	Sets the background color of an element	red (#ff0000, #ff0, rgb(255,0,0); rgb(100%, 0%, 0%))	em { color: #ff0000 }	
background-color		orange (#ffa500)	h5 { color: #ff0 }	
		yellow (#ffff00, #ff0, rgb(255,255,0); rgb(100%, 100%, 0))	p { color: rgb(255,0,0) }	
		olive (#808000)	div { color: rgb(100%, 0%, 0%) }	
		purple (#800080)	body {color: navy }	
		fuchsia (#ff00ff, #ff0, rgb(255,0,255); rgb(100%, 0%, 100%))	visited { color: #505050 }	
		white (#ffff, #fff, rgb(255,255,255); rgb(100%, 100%, 100%))	! { background-color: #ff0 }	
		lime (#00ff00, #0ff, rgb(0,255,0); rgb(0%, 100%, 0%))	body { color: black; background: white }	
		green (#008000)	body { color: white; background: black }	
		navy (#000080)	css file:	
		blue (#0000ff, #00f, rgb(0,0,255); rgb(0%, 0%, 100%))	body {	Example: 
		aqua (#00ffff, #0ff, rgb(0,255,255); rgb(0%, 0%, 100%))	background: aqua;	
		teal (#008080)	color: blue	
		black (#000000; #000; rgb(0,0,0); rgb(0%, 0%, 0%))	p { color: red }	
		silver (#c0c0c0)	span { color: #ffa500 }	
		gray (#808080)	html file:	
			Example: <p>This is a color example</p>	
background-image	Sets the background image of an element	<uri> none inherit	body { background-image: url("marble.png") }	
background-repeat	Specifies whether the image is repeated (tiled), and how	repeat repeat-x repeat-y no-repeat inherit	body { background: white url("pendant.png"); background-repeat: repeat-y; background-position: center; }	
background-attachment	Specifies whether it is fixed with regard to the viewport ('fixed') or scrolls along with the containing block ('scroll')	scroll fixed inherit	body { background: red url("pendant.png"); background-repeat: repeat-y; background-attachment: fixed; }	
background-position	Specified, this property specifies its initial position.	[[<percentage> <length> left center right] [<percentage> <length> top center bottom]?] [[left center right]] [[top center bottom]] inherit	body { background: url("banner.jpeg") right top }	
background	The 'background' property is a shorthand property for setting the individual background properties	[<background-color> <background-image> <background-repeat> <background-attachment> <background-position>] inherit	p { background: url("chess.png") gray 50% repeat fixed }	

More and details about colors and background >>> or >>>

Fonts				
font-family	Prioritized list of font family names and/or generic family names [[<font-family-name> <generic-family>]*] [<font-family-name> <generic-family>]*] inherit	'serif' (e.g. Times) 'sans-serif' (e.g. Helvetica) 'cursive' (e.g. Zapf-Chancery) 'fantasy' (e.g. Western) 'monospace' (e.g. Courier)	body { font-family: Gill, Helvetica, sans-serif } em { font-family: serif } p { font-family: sans-serif } a { font-family: cursive } h { font-family: fantasy } ul { font-family: monospace }	
font-style	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	normal italic oblique	h5 { font-style: normal } ol { font-style: italic } h3 { font-style: oblique } dl { font-variant: normal }	
font-variant	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	normal	dd { font-variant: small-caps }	
font-weight	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	normal bold	dl { font-weight: normal } dt { font-weight: bold }	

		<div><div><div><div><div><div></div><div>bolder</div></div></div><div><div><div></div><div>lighter</div></div><div><div></div><div>100</div></div><div><div></div><div>200</div></div><div><div></div><div>300</div></div><div><div></div><div>400 (eq normal)</div></div><div><div></div><div>500</div></div><div><div></div><div>600</div></div><div><div></div><div>700 (eq bold)</div></div><div><div></div><div>800</div></div><div><div></div><div>900</div></div></div></div></div><div><div><div><div></div><div>dt { font-weight: bolder }</div></div><div><div></div><div>dt { font-weight: lighter }</div></div><div><div></div><div>dt { font-weight: 100 }</div></div><div><div></div><div>dt { font-weight: 200 }</div></div><div><div></div><div>dt { font-weight: 300 }</div></div><div><div></div><div>dt { font-weight: 400 }</div></div><div><div></div><div>dt { font-weight: 500 }</div></div><div><div></div><div>dt { font-weight: 600 }</div></div><div><div></div><div>dt { font-weight: 700 }</div></div><div><div></div><div>dt { font-weight: 800 }</div></div><div><div></div><div>dt { font-weight: 900 }</div></div></div></div></div>
font-size	Selects between normal (sometimes referred to as "roman" or "upright"), italic and oblique faces within a font family	<div><div><div><div></div><div><absolute-size> <relative-size> <length> <percentage> inherit</div></div><div><div></div><div><absolute-size>: [xx-small x-small small medium large x-large xx-large]</div></div></div><div><div><div></div><div><relative-size>: [larger smaller]</div></div><div><div></div><div><length>: [px pt pc ex in cm mm]</div></div><div><div></div><div><percentage>: [em %]</div></div><div><div></div><div>* see units</div></div></div><div><div><div><div></div><div>h6 { font-size: x-small }</div></div><div><div></div><div>em { font-size: x-small }</div></div><div><div></div><div>h5 { font-size: small }</div></div><div><div></div><div>h4 { font-size: medium }</div></div><div><div></div><div>h3 { font-size: large }</div></div><div><div></div><div>h2 { font-size: x-large }</div></div><div><div></div><div>h1 { font-size: xx-large }</div></div><div><div></div><div>blockquote { font-size: larger }</div></div><div><div></div><div>p { font-size: 16px; }</div></div><div><div></div><div>@media print { p { font-size: 12pt; } }</div></div><div><div></div><div>em { font-size: 1.5em }</div></div><div><div></div><div>em { font-size: 150% }</div></div><div><div></div><div>p { font: 12px/14px sans-serif }</div></div><div><div></div><div>p { font: 80% sans-serif }</div></div><div><div></div><div>p { font: x-large/110% "New Century Schoolbook", serif }</div></div><div><div></div><div><i>p { font: bold italic large Palatino, serif }</i></div></div><div><div></div><div>p { font: normal small-caps 120%/120% fantasy }</div></div><div><div></div><div>span { font: caption }</div></div><div><div></div><div>span { font: icon }</div></div><div><div></div><div>span { font: menu }</div></div><div><div></div><div>span { font: message-box }</div></div><div><div></div><div>span { font: small-caption }</div></div><div><div></div><div>span { font: status-bar }</div></div></div></div></div>
font	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.	<div><div><div><div></div><div>[[<font-style> <font-variant> <font-weight> ?<font-size> <line-height> ?<font-family> <caption> <icon> <menu> <message-box> <small-caption> <status-bar> inherit</div></div><div><div></div><div>caption - The font used for captioned controls (e.g., buttons, drop-downs, etc.).</div></div><div><div></div><div>icon - The font used to label icons.</div></div><div><div></div><div>menu - The font used in menus (e.g., dropdown menus and menu lists).</div></div><div><div></div><div>message-box - The font used in dialog boxes.</div></div><div><div></div><div>small-caption - The font used for labeling small controls.</div></div><div><div></div><div>status-bar - The font used in window status bars.</div></div></div></div>
More and details about fonts >>>		
Text		
text-indent	Specifies the indentation of the first line of text in a block	<div><div><div><div></div><div><length> <percentage> inherit</div></div><div><div></div><div>p { text-indent: 16px }</div></div><div><div></div><div>div { text-indent: 3em }</div></div></div><div><div><div></div><div></div></div></div></div>
text-align	Describes how inline content of a block is aligned	<div><div><div><div></div><div>left right center justify inherit</div></div><div><div></div><div>p { text-align: left }</div></div><div><div></div><div></div></div><div><div></div><div>p { text-align: center }</div></div><div><div></div><div>p { text-align: right }</div></div></div><div><div><div></div><div></div></div></div></div>
vertical-align	Affects the vertical positioning inside a line box of the boxes generated by an inline-level element	<div><div><div><div></div><div>baseline sub super top text-top middle bottom text-bottom <percentage> <length> inherit</div></div><div><div></div><div>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.</div></div><div><div></div><div>middle - align the vertical midpoint of the box with the baseline of the parent box plus half the x-height of the parent.</div></div><div><div></div><div>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.)</div></div><div><div></div><div>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.)</div></div><div><div></div><div>text-top - align the top of the box with the top of the parent's content area</div></div><div><div></div><div>text-bottom - align the bottom of the box with the bottom of the parent's content area</div></div><div><div></div><div><percentage> - 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'.</div></div><div><div></div><div><length> - raise (positive value) or lower (negative value) the box by this distance. The value '0em' means the same as 'baseline'.</div></div><div><div></div><div>top - align the top of the aligned subtree with the top of the line box</div></div><div><div></div><div>bottom - align the bottom of the aligned subtree with the bottom of the line box.</div></div></div><div><div><div><div></div><div>div { vertical-align: baseline }</div></div><div><div></div><div>div { vertical-align: middle }</div></div><div><div></div><div>div { vertical-align: sub }</div></div><div><div></div><div>div { vertical-align: super }</div></div><div><div></div><div>div { vertical-align: text-top }</div></div><div><div></div><div>div { vertical-align: text-bottom }</div></div><div><div></div><div>div { vertical-align: -20% }</div></div><div><div></div><div>div { vertical-align: 15px }</div></div><div><div></div><div>div { vertical-align: top }</div></div><div><div></div><div>div { vertical-align: bottom }</div></div></div></div></div>
text-decoration	Describes decorations that are added to the text of an element using the element's color	<div><div><div><div></div><div>none [underline overline line-through blink] inherit</div></div><div><div></div><div>none - produces no text decoration</div></div><div><div></div><div>underline - each line of text is underlined</div></div><div><div></div><div>overline - each line of text has a line above it</div></div><div><div></div><div>line-through - each line of text has a line through the middle</div></div><div><div></div><div>blink - text blinks (alternates between visible and invisible)</div></div></div><div><div><div><div></div><div>stylesheet fragment:</div></div><div><div></div><div>blockquote { text-decoration: underline; color: blue; }</div></div><div><div></div><div>em { display: block; }</div></div><div><div></div><div>cite { color: fuchsia; }</div></div><div><div></div><div>html document fragment:</div></div><div><div></div><div><blockquote></div></div><div><div></div><div><p></div></div><div><div></div><div></div></div><div><div></div><div>Help, help!</div></div><div><div></div><div>I am under a hat!</div></div><div><div></div><div><cite>—GwieF</cite></div></div><div><div></div><div></p></div></div><div><div></div><div></blockquote></div></div><div><div></div><div>blockquote { letter-spacing: 0.1em }</div></div><div><div></div><div>h1 { word-spacing: 1em }</div></div><div><div></div><div>h1 { line-height: normal }</div></div><div><div></div><div>/* normal */</div></div><div><div></div><div>div { line-height: 1.2em }</div></div><div><div></div><div>/* length */</div></div><div><div></div><div>div { line-height: 1.2 }</div></div><div><div></div><div>/* number */</div></div><div><div></div><div>div { line-height: 55% }</div></div><div><div></div><div>div { line-height: 55% }</div></div></div></div></div>
letter-spacing	Specifies spacing behavior between text characters	<div><div><div><div></div><div>normal <length> inherit</div></div><div><div></div><div>blockquote { letter-spacing: 0.1em }</div></div></div><div><div><div></div><div></div></div></div></div>
word-spacing	Specifies spacing behavior between words	<div><div><div><div></div><div>normal <length> inherit</div></div><div><div></div><div>h1 { word-spacing: 1em }</div></div></div><div><div><div></div><div></div></div></div></div>
line-height	specifies the minimal height of line boxes within the element	<div><div><div><div></div><div>normal <number> <length> <percentage> inherit</div></div><div><div></</div></div></div></div>

		algorithm. The direction of this embedding level is given by the 'direction' property. Inside the element, reordering is done implicitly.		CSS fragment: hebrew, english, par {display: block} emph {font-weight: bold}	HEBREW1 HEBREW2 english3 HEBREW4 HEBREW5 HEBREW6 HEBREW7 HEBREW8 english9 english10 english11 HEBREW12 HEBREW13 english14 english15 english16 english17 HEBREW18 english19 HEBREW20	
More and details about text >>> about visual formatting model ('width', 'height', 'line-height' and 'vertical-align' properties) >>> about direction ('direction' and 'unicode-bid' properties) >>>						
Generated content						
content	This property is used with the :before and :after pseudo-elements to generate content in a document.	normal none [<string> <uri> <counter> attr(<identifier>) 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. <string> - text content (see the section on strings).	span:before { content: none; } li:before { content: normal; }			
			CSS fragment: span:before { content: "Chapter: "; }	Chapter: this is a chapter		
			HTML fragment: this is a chapter			
		<uri> - 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.	CSS fragment:			
			HTML fragment:			
		<counter> - 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)'.	CSS fragment:			
			HTML fragment:			
		open-quote and close-quote - these values are replaced by the appropriate string from the 'quotes' property.	CSS fragment: q:before { content: open-quote; } q:after { content: close-quote; } HTML fragment: <q>"Quote me!"</q>	"Quote me!"		
		no-open-quote and no-close-quote - introduces no content, but increments (decrements) the level of nesting for quotes.	CSS fragment: q:before { content: no-open-quote; } q:after { content: no-close-quote; } HTML fragment: <q>No quote me!"</q>	No quote me!		
quotes	This property specifies quotation marks for any number of embedded quotations.	attr(X) - this function returns as a string the value of attribute X for the subject of the 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.	CSS fragment:			
			HTML fragment:			
			HTML fragment:			
More and details about generated content >>>						
Automatic counters and numbering						
counter-increment	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	[<identifier> <integer>?]+ none inherit	CSS fragment: h3:before { content: "Chapter " counter(chapter) ". "; } counter-increment: chapter; } h3 { counter-reset: section; } h4:before { content: counter(chapter) ". " counter(section) " "; } counter-increment: section; }	Chapter 1. First chapter 0.1 First section 0.2 Second section		
counter-reset	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.	[<identifier> <integer>?]+ none inherit	HTML fragment: <h3>First chapter</h3> <h4>First section</h4> <h4>Second section</h4> <h3>Second chapter</h3> <h4>First section</h4> <h4>Second section</h4>	Chapter 1. Second chapter 0.1 First section 0.2 Second section		
More and details about automatic counters and numbering >>>						
Lists						
list-style-type	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.	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit	HTML fragment for all examples: First Second Third Fourth 			
		disc - rendering depends on the user agent.	CSS fragment: ol { list-style-type: disc; }	<ul style="list-style-type: none">• First• Second• Third• Fourth		
		circle - rendering depends on the user agent.	CSS fragment: ol { list-style-type: circle; }	<ul style="list-style-type: none">○ First○ Second○ Third○ Fourth		
		square - rendering depends on the user agent.	CSS fragment: ol { list-style-type: square; }	<ul style="list-style-type: none">■ First■ Second■ Third■ Fourth		
		decimal - decimal numbers, beginning with 1.	CSS fragment: ol { list-style-type: decimal; }	<ol style="list-style-type: none">1. First2. Second3. Third4. Fourth		
		decimal-leading-zero - decimal numbers padded by initial zeros (e.g., 01, 02, 03, ..., 98, 99).	CSS fragment: ol { list-style-type: decimal-leading-zero; }	<ol style="list-style-type: none">01. First02. Second03. Third04. Fourth		
		lower-roman - lowercase roman numerals (i, ii, iii, iv, v, etc.).	CSS fragment: ol { list-style-type: lower-roman; }	<ol style="list-style-type: none">i. Firstii. Secondiii. Thirdiv. Fourth		
		upper-roman - uppercase roman numerals (I, II, III, IV, V, etc.).	CSS fragment: ol { list-style-type: upper-roman; }	<ol style="list-style-type: none">I. FirstII. SecondIII. ThirdIV. Fourth		
		georgian - traditional Georgian numbering (an, ban, gan, ..., he, tan, in, in-an, ...).	CSS fragment: ol { list-style-type: georgian; }	<ol style="list-style-type: none">ა. Firstბ. Secondგ. Thirdდ. Fourth		
		armenian - traditional Armenian numbering	CSS fragment: ol { list-style-type: armenian; }	<ol style="list-style-type: none">Ա. FirstԲ. SecondԳ. ThirdԴ. Fourth		
		lower-latin or lower-alpha - lowercase ascii letters (a, b, c, ... Z).	CSS fragment: ol { list-style-type: lower-latin; }	<ol style="list-style-type: none">a. Firstb. Secondc. Thirdd. Fourth		
		upper-latin or upper-alpha - uppercase ascii letters (A, B, C, ... Z).	CSS fragment: ol { list-style-type: upper-alpha; }	<ol style="list-style-type: none">A. FirstB. SecondC. ThirdD. Fourth		
		lower-greek - lowercase classical Greek alpha, beta, gamma, ... (α, β, γ, ...)	CSS fragment: ol { list-style-type: lower-greek; }	<ol style="list-style-type: none">α. Firstβ. Secondγ. Thirdδ. Fourth		
list-style-image	Sets the image that will be used as the list item marker	<uri> none inherit	ul { list-style-image: url("http://www.iconsfree.org/icon/image/image20051107/http_www_20051107021654.icons-design.thesign.ro_tkcPainter_images_ellipse.png"); }			
list-style-position	Specifies the position of the marker box in the principal block box	inside outside inherit outside - the marker box is outside the principal block box inside - the marker box is the first inline box in the principal block box, after which the element's content flows.	CSS fragment: ul { list-style-position: outside; } ul.compact { list-style-position: inside; } HTML fragment:	<ul style="list-style-type: none">• firstlist item• secondlist item		

			<div> first
 list item second
 list item <ul class="compact"> first
 list item second
 list item </div>	<div><ul style="list-style-type: none">firstlist itemsecondlist item</div>
list-style	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	[<list-style-type> <list-style-position> <list-style-image>] inherit	ul > li > ul { list-style: circle outside }	
More and details about lists >>>				
Tables				
caption-side	Specifies the position of the caption box with respect to the table box.	top bottom inherit	caption { caption-side: bottom; width: auto; text-align: left }	
table-layout	Controls the algorithm used to lay out the table cells, rows, and columns.	auto fixed inherit	table { table-layout: fixed; margin-left: 2em; margin-right: 2em }	
border-collapse	Selects a table's border model.	collapse separate inherit	table { border: outset 10pt; border-collapse: separate; border-spacing: 15pt }	
border-spacing	Selects a table's border model.	<length> <length>? inherit		
empty-cells	Controls the rendering of borders and backgrounds around cells that have no visible content	show hide inherit	table { empty-cells: show }	
More and details about tables >>>				
Miscellaneous				
Public Domain 2006-2015 Alexander Krassotkin				
<div><div> W3C XHTML 3.0</div><div> W3C CSS</div></div>				