

CSS Style Properties

HTML-5.com is an HTML User's Guide and quick reference of HTML elements and attributes for web developers who code HTML web pages, not only for HTML 5 but for HTML coding in general, with demos and examples of HTML code plus a cheat sheet for web developers.

[HTML-5.com](#) > [itemscopehttp://data-vocabulary.org/BreadcrumbHTML 5](#) > [itemscopehttp://data-vocabulary.org/Breadcrumb](#)
[CSS Style Properties](#)

``

If you find this helpful, please click the [Google Button](#) to the left, if it is white, to make it turn blue. Thank you! (It also helps find this page again more easily.)

CSS Style Properties

[back to top](#)

Units in Property Values

- `npx` - *n* pixels
- `nin` - *n* inches
- `ncm` - *n* centimeters
- `mmm` - *n* millimeters
- `nn%` - *nn* percent
- `npt` - *n* points (1/72 inch)
- `npc` - *n* pica characters (12 points)
- `nem` - *n* times "m" width
- `nex` - *n* times "x" height

[back to top](#)

HTML Colors

These are the color names and other property values for the sixteen standard HTML colors. For other colors, use the [HTML Color Picker](#).

Color Name	Hex Color Code	Red / Green / Blue / Alpha	Hue / Saturation / Luminance / Alpha
<code>aqua</code>	<code>#00ffff</code>	<code>rgba(0, 255, 255, 1.0)</code>	<code>hsla(180, 100%, 50%, 1.00)</code>
<code>black</code>	<code>#000000</code>	<code>rgba(0, 0, 0, 1.0)</code>	<code>hsla(0, 0%, 0%, 1.00)</code>

blue	#0000ff	rgba(0, 0, 255, 1.0)	hsla(240, 100%, 50%, 1.0)
fuchsia	#ff00ff	rgba(255, 0, 255, 1.0)	hsla(300, 100%, 50%, 1.0)
gray or grey	#808080	rgba(128, 128, 128, 1.0)	hsla(0, 0%, 50%, 1.0)
green	#008000	rgba(0, 128, 0, 1.0)	hsla(120, 100%, 25%, 1.0)
lime	#00ff00	rgba(0, 255, 0, 1.0)	hsla(120, 100%, 50%, 1.0)
maroon	#800000	rgba(128, 0, 0, 1.0)	hsla(0, 100%, 25%, 1.0)
navy	#000080	rgba(0, 0, 128, 1.0)	hsla(240, 100%, 25%, 1.0)
olive	#808000	rgba(128, 128, 0, 1.0)	hsla(60, 100%, 25%, 1.0)
purple	#800080	rgba(128, 0, 128, 1.0)	hsla(300, 100%, 25%, 1.0)
red	#ff0000	rgba(255, 0, 0, 1.0)	hsla(0, 100%, 50%, 1.0)
silver	#c0c0c0	rgba(192, 192, 192, 1.0)	hsla(0, 0%, 75%, 1.0)
teal	#008080	rgba(0, 128, 128, 1.0)	hsla(180, 100%, 25%, 1.0)
white	#ffffff	rgba(255, 255, 255, 1.0)	hsla(0, 0%, 100%, 1.0)
yellow	#ffff00	rgba(255, 255, 0, 1.0)	hsla(60, 100%, 50%, 1.0)

Note that, as shown above, the alpha value *must* be specified as a decimal number (ex: 1.00), *not* a percentage (100%).

[back to top](#)

Background Properties

background:

background-color:

Use the [HTML Color Picker](#) for an easy way to choose HTML colors.

[back to top](#)

Border Properties

A `<dfn>border</dfn>` may be specified around an [element](#) or one each side individually. Any [margin spacing](#) appears *outside* the border while any [padding](#) appears *inside* the border.

`border: width style color`

The `border` property is a shorthand for specifying [border-width](#), [border-style](#) and [border-color](#)

`border-top: width style color`

The `border-top` property is a shorthand for specifying [border-width-top](#), [border-style-top](#) and [border-color-top](#)

`border-right: width style color`

The `border-right` property is a shorthand for specifying [border-width-right](#), [border-style-right](#) and [border-color-right](#)

`border-bottom: width style color`

The `border-bottom` property is a shorthand for specifying [border-width-bottom](#), [border-style-bottom](#) and [border-color-bottom](#)

`border-left: width style color`

The `border-left` property is a shorthand for specifying [border-width-left](#), [border-style-left](#) and [border-color-left](#)

`border-width`

`medium` (default)

`thin`

`thick`

`inherit`

[nunits](#)

Note that Firefox *includes* the border width in the [height](#) while Safari and Chrome *add* the border width to the [height](#).

`border-style: style;`

`border-style: top/bottom right/left;`

`border-style: top right/left bottom;`

`border-style: top right bottom left;`

See the [<div> tag tutorial](#) and [<div> tag examples](#) for various examples of border styles

`border-color:`

Use the [HTML Color Picker](#) for an easy way to choose HTML colors.

[back to top](#)

Cursor Properties

`cursor`

`cursor: auto`

`cursor: default`

`cursor: crosshair`

`cursor: help`

`cursor: move`

`cursor: pointer`

`cursor: progress`

`cursor: text`

`cursor: wait`

`cursor: n-resize`

`cursor: ne-resize`

`cursor: nw-resize`

`cursor: s-resize`

`cursor: se-resize`

`cursor: sw-resize`

`cursor: e-resize`

`cursor: w-resize`

`cursor: url` - list of URLs separated by commas; include one of the named cursors at the end of the list for browsers that do not support custom cursors

[back to top](#)

Foreground Color Properties

`color`

see [HTML Colors](#) above for some of the possible ways to code [HTML color values](#)

[back to top](#)

Float, Display and Visibility Properties

`clear`

`display`

`display: block;` - an element with this style will be separated from content outside the element by line breaks

`display: compact;`

`display: inline;` - an element with this style will appear inline with other [phrasing content](#)

`display: inline-block;` - an element with this style will have the width, height, margin, padding and other properties of a block but will appear inline with other [phrasing content](#)

`display: list-item;` - the display style of an [li element](#); can be used to override the behavior of another element type to make it look and act like a list item

`display: marker;` - a box with additional space will be allocated outside the main block in an inline manner, where a marker such as a list item bullet can be placed, possibly using the `:before` and/or `:after` pseudo element selectors

`display: inherit;` - the value of the display property will be inherited from its parent, regardless of any value selected for the element from a [style sheet](#) or [inline style](#)

display: inline-table; - an element with this style will look and act like a table, except that it will appear inline with other [phrasing content](#)

display: none; - an element with this style and its contents will not be displayed and furthermore will not affect the page layout, as if the element effectively did not exist

display: run-in;

display: table; - the display style of a [table element](#); can be used to override the behavior of another element type to make it look and act like a table

display: table-caption; - the display style of a [caption element](#); can be used to override the behavior of another element type to make it look and act like a table caption

display: table-cell; - the display style of a [td element](#); can be used to override the behavior of another element type to make it look and act like a table cell

display: table-column; - the display style of a [col element](#); can be used to override the behavior of another element type to make it look and act like a table column

display: table-column-group; - the display style of a [colgroup element](#); can be used to override the behavior of another element type to make it look and act like a table column group

display: table-header-group; - the display style of a [thead element](#); can be used to override the behavior of another element type to make it look and act like a table header

display: table-footer-group; - the display style of a [tfoot element](#); can be used to override the behavior of another element type to make it look and act like a table footer

display: table-row; - the display style of a [tr element](#); can be used to override the behavior of another element type to make it look and act like a table row

display: table-row-group; - the display style of a [tbody element](#); can be used to override the behavior of another element type to make it look and act like rows in a table body

float

overflow

overflow-style

See also [marquee styles](#)

position

position: absolute;

position: fixed;

position: relative;

visibility

visibility: collapse;

visibility: hidden;

visibility: inherit;

visibility: visible;

z-index

Specifies the depth or <dfn>level</dfn> of the element and its contents relative to its parent element. Lower z-index values, including negative numbers, represent lower (deeper) layers and higher z-index values represent higher layers, almost like laying sheets of paper one on top of another. Also requires specifying an explicit position property (position: relative, position: absolute or position: fixed).

[back to top](#)

Font Properties**font-famliy****font-size****font-weight**

[back to top](#)

Margin Spacing Properties**margin**

```
margin: <var><#width#></var>;  
margin: <var><#top/bottom#></var> <var><#right/left#></var>;  
margin: <var><#top#></var> <var><#right/left#></var>  
<var><#bottom#></var>;  
margin: <var><#top#></var> <var><#right#></var>  
<var><#bottom#></var> <var><#left#></var>;  
margin-top: <var><#width#></var>;  
margin-right: <var><#width#></var>;  
margin-bottom: <var><#width#></var>;  
margin-left: <var><#width#></var>;
```

opacity

[back to top](#)

Padding Properties**padding**

```
padding: <var><#width#></var>;  
padding: <var><#top/bottom#></var> <var><#right/left#></var>;  
padding: <var><#top#></var> <var><#right#></var>  
<var><#bottom#></var> <var><#left#></var>;  
padding-top: <var><#width#></var>;  
padding-right: <var><#width#></var>;  
padding-bottom: <var><#width#></var>;  
padding-left: <var><#width#></var>;
```

[back to top](#)

Size Properties

`line-height`

`height`

`width`

[back to top](#)

Text Properties

`text-align: left`

`text-align: center`

`text-align: right`

`text-align: justify`

`text-align: start`

`text-align: end`

`text-align: match-parent`

`text-align: "char"`

align table cells at the character *char*, for example `text-align: "."` aligns numbers in table cells based on the position of their decimal point

`text-decoration`

`vertical-align: baseline`

(default)

`vertical-align: bottom`

`vertical-align: inherit`

`vertical-align: middle`

`vertical-align: sub`

`vertical-align: super`

`vertical-align: text-bottom`

`vertical-align: text-top`

vertical-align: top

vertical-align: height

A positive value raises the content above the baseline while a negative value lowers it below the baseline. If the value is a percentage, the height is that percentage of the [line-height](#).

[back to top](#)

Text Wrapping and White Space Handling Properties

white-space

white-space: normal

equivalent to `white-space-collapsing: collapse; text-wrap: normal;`

- Any sequence of multiple consecutive spaces and other whitespace is collapsed and treated as a single space.
- Line breaks (new lines) and tabs are treated as any other whitespace.
- Content will automatically wrap onto additional lines as necessary based on the limitations of the available space. Wrapping will usually occur where breakable white space occurs in the HTML code.

white-space: nowrap

equivalent to `white-space-collapsing: collapse; text-wrap: none;`

- Any sequence of multiple consecutive spaces and other whitespace is collapsed and treated as a single space.
- Line breaks (new lines) and tabs are treated as any other whitespace.
- Content only appears on a new line where line breaks occur in the HTML source.

white-space: pre

equivalent to `white-space-collapsing: preserve; text-wrap: none;`

- Sequences of whitespace including line breaks (new lines) and tabs are preserved.
- Content only appears on a new line where line breaks occur in the HTML source.
- White space characters indenting HTML code or text at the beginning of a line are preserved like any other white space.

white-space: pre-line

equivalent to `white-space-collapsing: preserve-breaks; text-wrap: normal;`

- Line breaks (new lines) are preserved.
- Any white space characters indenting HTML code or text at the beginning of a line, which would normally be collapsed with the preceding new line character, are ignored.
- Any other sequence of multiple consecutive spaces and other whitespace (excluding line breaks mentioned above) is collapsed and treated as a single space.
- Tabs are treated as whitespace.

- Content will automatically wrap onto additional lines as necessary based on the limitations of the available space. Wrapping will usually occur where breakable white space occurs in the HTML code.

white-space: pre-wrap

equivalent to `white-space-collapsing: preserve; text-wrap: normal;`

- Sequences of whitespace including line breaks (new lines) and tabs are preserved.
- Content will automatically wrap onto additional lines as necessary based on the limitations of the available space. Wrapping will usually occur where breakable white space occurs in the HTML code.
- White space characters indenting HTML code or text at the beginning of a line are preserved like any other white space.

Select which white-space property to use

White Space Handling	text-wrap		use CSS style
New Lines, Spaces & Tabs			
collapse	normal (wrap)		white-space: normal
collapse	not wrapped		white-space: nowrap
preserve	not wrapped		white-space: pre
preserve	normal (wrap)		white-space: pre-wrap
New Lines	Spaces & Tabs	text-wrap	use CSS style
preserve	collapse	normal (wrap)	white-space: pre-line
preserve	collapse	not wrapped	white-space-collapsing: preserve-breaks;

				text-wrap: no
(not recommended)				

word-wrap: normal

word-wrap: break-word

Some browsers do not support the [white-space-collapsing property](#) or [text-wrap property](#) (below). Therefore, to control text wrapping the use of the [white-space property](#) (above) is recommended unless a specific capability, such as decimal point alignment, is required.

text-wrap: normal

Content will automatically wrap onto additional lines as necessary based on the limitations of the available space. Wrapping will usually occur where breakable white space occurs in the HTML code.

text-wrap: none

Content from one line of the source will not wrap onto additional lines.

text-wrap: suppress

Like [text-wrap: none](#), the text will start on a new line if it all cannot fit on the current line. But it *will* wrap onto additional lines if it does not fit on a single line.

text-wrap: unrestricted

Content can wrap onto additional lines between any two characters, even in the middle of a word, based on the limitations of the available space.

white-space-collapsing: collapse

- Any sequence of multiple consecutive spaces and other whitespace is collapsed and treated as a single space.
- Line breaks (new lines) and tabs are treated as any other whitespace.
- Any white space characters indenting HTML code or text at the beginning of a line are ignored.

white-space-collapsing: discard

Any sequence of white space characters is treated as a single Unicode zero-width non-joiner ([‌](#)). This might be used to keep the letters of an acronym (initialism) separate, in a language where ligatures combining adjacent characters are common.

white-space-collapsing: trim-inner

Any leading or trailing white space in a block is ignored.

white-space-collapsing: preserve

All white space is preserved, including any white space characters indenting HTML code or text at the beginning of a line.

white-space-collapsing: preserve trim-inner

In addition to `white-space-collapsing: preserve` processing, any leading or trailing white space in a block is ignored.

white-space-collapsing: preserve-breaks

- Line breaks (new lines) are preserved. Content only appears on a new line where line breaks occur in the HTML source.
- Any white space characters indenting HTML code or text at the beginning of a line, which would normally be collapsed with the preceding new line character, are ignored.
- Any other sequence of multiple consecutive spaces and other whitespace (excluding line breaks mentioned above) is collapsed and treated as a single space.
- Tabs are treated as whitespace.

white-space-collapsing: preserve-breaks trim-inner

In addition to `white-space-collapsing: preserve-breaks` processing, any leading or trailing white space in a block is ignored.

[back to top](#)

THE END