



Technologies ▼

References & Guides ▼

Feedback ▼

CSS reference

English ▼

Use this **CSS reference** to browse an alphabetical index of all of the standard CSS properties, pseudo-classes, pseudo-elements, data types, and at-rules. You can also browse key CSS concepts and a list of selectors organized by type. Also included is a brief DOM-CSS / CSSOM reference.

Basic rule syntax

Style rule syntax

```
style-rule ::=
  selectors-list {
    properties-list
}
```

... where:

```
selectors-list ::=
   selector[:pseudo-class] [::pseudo-element]
   [, selectors-list]
```

```
properties-list ::=
  [property : value] [; properties-list]
```

See the index of *selectors*, *pseudo-classes*, and *pseudo-elements* below. The syntax for each specified *value* depends on the data type defined for each specified *property*.

Style rule examples

```
1 strong {
2  color: red;
3  }
4 
5  div.menu-bar li:hover > ul {
6   display: block;
7  }
```

For a beginner-level introduction to the syntax of selectors, see our guide on CSS Selectors. Be aware that any syntax error in a rule definition invalidates the entire rule. Invalid rules are ignored by the browser. Note that CSS rule definitions are entirely (ASCII) text-based, whereas DOM-CSS / CSSOM (the rule management system) is object-based.

At-rule syntax

As the structure of at-rules varies widely, please see At-rule to find the syntax of the specific one you want.

Keyword index



Note: The property names in this index do **not** include the JavaScript names where they differ from the CSS standard names.

- A
-webkit-line-clamp :active

additive-symbols (@counter-style)	background-origin
::after (:after)	background-position
align-content	background-repeat
align-items	background-size
align-self	<basic-shape></basic-shape>
all	::before (:before)
<an-plus-b></an-plus-b>	<ble><ble><bre><bre><bre><bre><bre><bre><bre><br< td=""></br<></bre></bre></bre></bre></bre></bre></bre></ble></ble>
<angle></angle>	block-size
<angle-percentage></angle-percentage>	blur()
animation	border
animation-delay	border-block
animation-direction	border-block-color
animation-duration	border-block-end
animation-fill-mode	border-block-end-color
animation-iteration-count	border-block-end-style
animation-name	border-block-end-width
animation-play-state	border-block-start
animation-timing-function	border-block-start-color
@annotation	border-block-start-style
annotation()	border-block-start-width
attr()	border-block-style
	border-block-width
В	border-bottom
::backdrop	border-bottom-color
backface-visibility	border-bottom-left-radius
background	border-bottom-right-radius
background-attachment	border-bottom-style
background-blend-mode	border-bottom-width
background-clip	border-collapse
background-color	border-color
background-image	border-end-end-radius

border-end-start-radius	border-style
border-image	border-top
border-image-outset	border-top-color
border-image-repeat	border-top-left-radius
border-image-slice	border-top-right-radius
border-image-source	border-top-style
border-image-width	border-top-width
border-inline	border-width
border-inline-color	bottom
border-inline-end	@bottom-center
border-inline-end-color	box-decoration-break
border-inline-end-style	box-shadow
border-inline-end-width	box-sizing
border-inline-start	break-after
border-inline-start-color	break-before
border-inline-start-style	break-inside
border-inline-start-width	<pre>brightness()</pre>
border-inline-style	
border-inline-width	C
border-left	calc()
border-left-color	caption-side
border-left-style	caret-color
border-left-width	ch
border-radius	<pre>@character-variant</pre>
border-right	<pre>character-variant()</pre>
border-right-color	@charset
border-right-style	:checked
border-right-width	circle()
border-spacing	<pre>clamp()</pre>
border-start-end-radius	clear
border-start-start-radius	clip

clip-path	:default
CM	deg
<color></color>	<dimension></dimension>
color	:dir
color-adjust	direction
column-count	:disabled
column-fill	display
column-gap	<display-box></display-box>
column-rule	<display-inside></display-inside>
column-rule-color	<display-internal></display-internal>
column-rule-style	<display-legacy></display-legacy>
column-rule-width	<display-listitem></display-listitem>
column-span	<display-outside></display-outside>
column-width	dpcm
columns	dpi
<pre>conic-gradient()</pre>	dppx
content	drop-shadow()
contrast()	_
<counter></counter>	E
counter-increment	element()
counter-reset	ellipse()
counter-set	em
@counter-style	:empty
counters()	empty-cells
<pre>cross-fade()</pre>	:enabled
<pre>cubic-bezier()</pre>	env()
::cue	ex
cursor	Г
<custom-ident></custom-ident>	F
D	fallback (@counter-style)
D	filter
	<filter-function></filter-function>

```
:first
                                           font-style (@font-face)
:first-child
                                           font-synthesis
::first-letter (:first-letter)
                                           font-variant
::first-line (:first-line)
                                           font-variant (@font-face)
:first-of-type
                                           font-variant-alternates
fit-content()
                                           font-variant-caps
<flex>
                                           font-variant-east-asian
flex
                                           font-variant-ligatures
flex-basis
                                           font-variant-numeric
flex-direction
                                           font-variant-position
flex-flow
                                           font-variation-settings (@font-face)
flex-grow
                                           font-weight
flex-shrink
                                           font-weight (@font-face)
flex-wrap
                                           format()
                                           fr
float
:focus
                                           <frequency>
font
                                           <frequency-percentage>
@font-face
                                           :fullscreen
font-family
                                           G
font-family (@font-face)
                                           gap
font-feature-settings
font-feature-settings (@font-face)
                                           grad
                                           <gradient>
@font-feature-values
                                           grayscale()
font-kerning
                                           grid
font-language-override
                                           grid-area
font-optical-sizing
                                           grid-auto-columns
font-size
                                           grid-auto-flow
font-size-adjust
                                           grid-auto-rows
font-stretch
                                           grid-column
font-stretch (@font-face)
                                           grid-column-end
font-style
```

grid-column-start	inherit
grid-row	initial
grid-row-end	inline-size
grid-row-start	inset
grid-template	<pre>inset()</pre>
grid-template-areas	inset-block
grid-template-columns	inset-block-end
grid-template-rows	inset-block-start
	inset-inline
H	inset-inline-end
Hz	inset-inline-start
hanging-punctuation	<integer></integer>
height	:invalid
height (@viewport)	invert()
@historical-forms	isolation
havan	
:hover	
hsl()	J
	J justify-content
hsl()	
hsl() hsla()	justify-content
hsl() hsla() hue-rotate()	<pre>justify-content justify-items justify-self</pre>
hsl() hsla() hue-rotate()	<pre>justify-content justify-items</pre>
hsl() hsla() hue-rotate()	<pre>justify-content justify-items justify-self</pre>
hsl() hsla() hue-rotate() hyphens	<pre>justify-content justify-items justify-self</pre>
<pre>hsl() hsla() hue-rotate() hyphens </pre> <pre>I <ident></ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes</pre>
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/></ident></pre>	<pre>justify-content justify-items justify-self </pre> <pre>K</pre> kHz
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image()</ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes</pre>
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image() image-orientation</ident></pre>	justify-content justify-items justify-self K kHz @keyframes
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image() image-orientation image-rendering</ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes</pre> L :lang
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image() image-orientation image-rendering image-set()</ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes L :lang :last-child</pre>
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image() image-orientation image-rendering image-set() @import</ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes L :lang :last-child :last-of-type</pre>
<pre>hsl() hsla() hue-rotate() hyphens I <ident> <image/> image() image-orientation image-rendering image-set() @import in</ident></pre>	<pre>justify-content justify-items justify-self K kHz @keyframes L :lang :last-child :last-of-type leader()</pre>

@left-bottom	mask-mode
<length></length>	mask-origin
<length-percentage></length-percentage>	mask-position
letter-spacing	mask-repeat
line-break	mask-size
line-height	mask-type
<pre>linear-gradient()</pre>	<pre>matrix()</pre>
:link	matrix3d()
list-style	max()
list-style-image	max-height
list-style-position	<pre>max-height (@viewport)</pre>
list-style-type	max-width
local()	<pre>max-width (@viewport)</pre>
	max-zoom (@viewport)
M	@media
margin	min()
margin-block	min-block-size
margin-block-end	min-height
margin-block-start	min-height (@viewport)
margin-bottom	min-inline-size
margin-inline	min-width
margin-inline-end	min-width (@viewport)
margin-inline-start	min-zoom (@viewport)
margin-left	minmax()
margin-right	mix-blend-mode
margin-top	mm
::marker	ms
mask	
mask-clip	N
mask-composite	@namespace
mask-image	negative (@counter-style)

:not	P
:nth-child	pad (@counter-style)
:nth-last-child	padding
:nth-last-of-type	padding-block
:nth-of-type	padding-block-end
<number></number>	padding-block-start
	padding-bottom
0	padding-inline
object-fit	padding-inline-end
object-position	padding-inline-start
:only-child	padding-left
:only-of-type	padding-right
opacity	padding-top
opacity()	@page
:optional	page-break-after
order	page-break-before
orientation (@viewport)	page-break-inside
@ornaments	paint()
ornaments()	рс
orphans	<pre><percentage></percentage></pre>
:out-of-range	perspective
outline	perspective()
outline-color	perspective-origin
outline-offset	place-content
outline-style	place-items
outline-width	•
overflow	place-self
overflow-wrap	::placeholder
overflow-x	pointer-events
overflow-y	polygon()
	<pre><position></position></pre>
	position

<pre>prefix (@counter-style)</pre>	rotate3d()
pt	rotateX()
px	rotateY()
	rotateZ()
Q	row-gap
Q	
quotes	S
_	S
R	<pre>saturate()</pre>
rad	scale
<pre>radial-gradient()</pre>	scale()
<pre>range (@counter-style)</pre>	scale3d()
<ratio></ratio>	scaleX()
:read-only	scaleY()
:read-write	scaleZ()
rect()	:scope
rem	scroll-behavior
repeat()	scroll-margin
<pre>repeating-linear-gradient()</pre>	scroll-margin-block
repeating-radial-gradient()	scroll-margin-block-end
:required	scroll-margin-block-start
resize	scroll-margin-bottom
<resolution></resolution>	scroll-margin-inline
revert	scroll-margin-inline-end
rgb()	scroll-margin-inline-start
rgba()	scroll-margin-left
:right	scroll-margin-right
right	scroll-margin-top
@right-bottom	scroll-padding
:root	scroll-padding-block
rotate	scroll-padding-block-end
rotate()	

```
scroll-padding-block-start
                                           stylistic()
scroll-padding-bottom
                                           suffix (@counter-style)
scroll-padding-inline
                                           @supports
scroll-padding-inline-end
                                           @swash
scroll-padding-inline-start
                                           swash()
scroll-padding-left
                                           symbols (@counter-style)
scroll-padding-right
                                           symbols()
                                           system (@counter-style)
scroll-padding-top
scroll-snap-align
                                           T
scroll-snap-stop
                                           tab-size
scroll-snap-type
                                           table-layout
scrollbar-color
                                           :target
scrollbar-width
                                           target-counter()
::selection
                                           target-counters()
selector()
                                           target-text()
sepia()
                                           text-align
<shape>
                                           text-align-last
shape-image-threshold
                                           text-combine-upright
shape-margin
                                           text-decoration
shape-outside
                                           text-decoration-color
skew()
                                           text-decoration-line
skewX()
                                           text-decoration-style
skewY()
                                           text-decoration-thickness
::slotted
                                           text-emphasis
speak-as (@counter-style)
                                           text-emphasis-color
src (@font-face)
                                           text-emphasis-position
steps()
                                           text-emphasis-style
<string>
                                           text-indent
@styleset
                                           text-justify
styleset()
                                           text-orientation
@stylistic
```

text-overflow unicode-bidi text-rendering unicode-range (@font-face) text-shadow unset text-transform <url> text-underline-offset url() text-underline-position user-zoom (@viewport) <time> V <time-percentage> :valid <timing-function> var() top vertical-align @top-center vh touch-action @viewport transform visibility transform-box :visited <transform-function> vmax transform-origin vmin transform-style VW transition transition-delay W transition-duration white-space transition-property widows transition-timing-function width translate width (@viewport) translate() will-change translate3d() word-break translateX() word-spacing translateY() word-wrap translateZ() writing-mode turn X IJ

Х

Z Others

z-index

zoom (@viewport)

Selectors

The following are the various selectors, which allow styles to be conditional based on various features of elements within the DOM.

Basic selectors

Basic selectors are fundamental selectors; these are the most basic selectors that are frequently combined to create other, more complex selectors.

- Universal selector *, ns | *, * | *, | *
- Type selector elementname
- Class selector .classname
- ID selector #idname
- Attribute selector [attr=value]

Grouping selectors

Selector list A, B

Specifies that both A and B elements are selected. This is a grouping method to select several matching elements.

Combinators

Combinators are selectors that establish a relationship between two or more simple selectors, such as "A is a child of B" or "A is adjacent to B."

Adjacent sibling combinator A + B

Specifies that the elements selected by both A and B have the same parent and that the element selected by B immediately follows the element selected by A horizontally.

General sibling combinator A ~ B

Specifies that the elements selected by both A and B share the same parent and that the element selected by A comes before—but not necessarily immediately before—the element selected by B.

Child combinator A > B

Specifies that the element selected by B is the direct child of the element selected by A.

Descendant combinator A B

Specifies that the element selected by B is a descendant of the element selected by A, but is not necessarily a direct child.

Column combinator A | | B

Specifies that the element selected by B is located within the table column specified by A. Elements which span multiple columns are considered to be a member of all of those columns.

Pseudo

Pseudo classes:

Specifies a special state of the selected element(s).

Pseudo elements ::

Represents entities that are not included in HTML.



See also: Selectors in the Selectors Level 4 specification.

Concepts

Syntax and semantics

- CSS syntax
- At-rules
- Cascade
- Comments
- Descriptor
- Inheritance
- · Shorthand properties
- Specificity
- Value definition syntax
- · CSS unit and value types

Values

- Actual value
- Computed value
- Initial value
- Resolved value
- · Specified value
- Used value

Layout

- · Block formatting context
- Box model
- Containing block
- Layout mode
- Margin collapsing
- Replaced elements
- Stacking context
- Visual formatting model

DOM-CSS / CSSOM

Major object types

- DocumentOrShadowRoot.styleSheets
- styleSheets[i].cssRules
- cssRules[i].cssText (selector & style)
- cssRules[i].selectorText
- HTMLElement.style
- HTMLElement.style.cssText (just style)
- Element.className
- Element.classList

Important methods

- CSSStyleSheet.insertRule()
- CSSStyleSheet.deleteRule()

See also

- Mozilla CSS extensions (prefixed with -moz-)
- WebKit CSS extensions (mostly prefixed with -webkit-)
- Microsoft CSS extensions (prefixed with -ms-)
 - ② Last modified: Oct 26, 2019, by MDN contributors

Basic rule syntax
Keyword index
Selectors
Concepts
DOM-CSS / CSSOM
See also

Related Topics

CSS

CSS Reference

Learn the best of web development

Get the latest and greatest from MDN delivered straight to your inbox.

you@example.com

Sign up now

×