

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.

-
[-webkit-line-clamp](#)

A
[:active](#)

additive-symbols (@counter-style)
::after (:after)
align-content
align-items
align-self
all
<an-plus-b>
<angle>
<angle-percentage>
animation
animation-delay
animation-direction
animation-duration
animation-fill-mode
animation-iteration-count
animation-name
animation-play-state
animation-timing-function
@annotation
annotation()
attr()

B

::backdrop
backface-visibility
background
background-attachment
background-blend-mode
background-clip
background-color
background-image

background-origin
background-position
background-repeat
background-size
<basic-shape>
::before (:before)
<blend-mode>
block-size
blur()
border
border-block
border-block-color
border-block-end
border-block-end-color
border-block-end-style
border-block-end-width
border-block-start
border-block-start-color
border-block-start-style
border-block-start-width
border-block-style
border-block-width
border-bottom
border-bottom-color
border-bottom-left-radius
border-bottom-right-radius
border-bottom-style
border-bottom-width
border-collapse
border-color
border-end-end-radius

border-end-start-radius
border-image
border-image-outset
border-image-repeat
border-image-slice
border-image-source
border-image-width
border-inline
border-inline-color
border-inline-end
border-inline-end-color
border-inline-end-style
border-inline-end-width
border-inline-start
border-inline-start-color
border-inline-start-style
border-inline-start-width
border-inline-style
border-inline-width
border-left
border-left-color
border-left-style
border-left-width
border-radius
border-right
border-right-color
border-right-style
border-right-width
border-spacing
border-start-end-radius
border-start-start-radius

border-style
border-top
border-top-color
border-top-left-radius
border-top-right-radius
border-top-style
border-top-width
border-width
bottom
@bottom-center
box-decoration-break
box-shadow
box-sizing
break-after
break-before
break-inside
brightness()

C

calc()
caption-side
caret-color
ch
@character-variant
character-variant()
@charset
:checked
circle()
clamp()
clear
clip

clip-path	:default
cm	deg
<color>	<dimension>
color	:dir
color-adjust	direction
column-count	:disabled
column-fill	display
column-gap	<display-box>
column-rule	<display-inside>
column-rule-color	<display-internal>
column-rule-style	<display-legacy>
column-rule-width	<display-listitem>
column-span	<display-outside>
column-width	dpcm
columns	dpi
conic-gradient()	dppx
content	drop-shadow()
contrast()	
<counter>	
counter-increment	
counter-reset	
counter-set	
@counter-style	
counters()	
cross-fade()	
cubic-bezier()	
::cue	
cursor	
<custom-ident>	

D

E

element()
ellipse()
em
:empty
empty-cells
:enabled
env()
ex

F

fallback (@counter-style)
filter
<filter-function>

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

grid-column-start
grid-row
grid-row-end
grid-row-start
grid-template
grid-template-areas
grid-template-columns
grid-template-rows

H

Hz
hanging-punctuation
height
height (@viewport)
@historical-forms
:hover
hsl()
hsla()
hue-rotate()
hyphens

I

<ident>
<image>
image()
image-orientation
image-rendering
image-set()
@import
in
:in-range
:indeterminate

inherit
initial
inline-size
inset
inset()
inset-block
inset-block-end
inset-block-start
inset-inline
inset-inline-end
inset-inline-start
<integer>
:invalid
invert()
isolation

J

justify-content
justify-items
justify-self

K

kHz
@keyframes

L

:lang
:last-child
:last-of-type
leader()
:left
left

@left-bottom
<length>
<length-percentage>
letter-spacing
line-break
line-height
linear-gradient()
:link
list-style
list-style-image
list-style-position
list-style-type
local()

M

margin
margin-block
margin-block-end
margin-block-start
margin-bottom
margin-inline
margin-inline-end
margin-inline-start
margin-left
margin-right
margin-top
::marker
mask
mask-clip
mask-composite
mask-image

mask-mode
mask-origin
mask-position
mask-repeat
mask-size
mask-type
matrix()
matrix3d()
max()
max-height
max-height (@viewport)
max-width
max-width (@viewport)
max-zoom (@viewport)
@media
min()
min-block-size
min-height
min-height (@viewport)
min-inline-size
min-width
min-width (@viewport)
min-zoom (@viewport)
minmax()
mix-blend-mode
mm
ms

N

@namespace
negative (@counter-style)

:not

:nth-child

:nth-last-child

:nth-last-of-type

:nth-of-type

<number>

O

object-fit

object-position

:only-child

:only-of-type

opacity

opacity()

:optional

order

orientation (@viewport)

@ornaments

ornaments()

orphans

:out-of-range

outline

outline-color

outline-offset

outline-style

outline-width

overflow

overflow-wrap

overflow-x

overflow-y

P

pad (@counter-style)

padding

padding-block

padding-block-end

padding-block-start

padding-bottom

padding-inline

padding-inline-end

padding-inline-start

padding-left

padding-right

padding-top

@page

page-break-after

page-break-before

page-break-inside

paint()

pc

<percentage>

perspective

perspective()

perspective-origin

place-content

place-items

place-self

::placeholder

pointer-events

polygon()

<position>

position

prefix (@counter-style)

pt

px

Q

Q

quotes

R

rad

radial-gradient()

range (@counter-style)

<ratio>

:read-only

:read-write

rect()

rem

repeat()

repeating-linear-gradient()

repeating-radial-gradient()

:required

resize

<resolution>

revert

rgb()

rgba()

:right

right

@right-bottom

:root

rotate

rotate()

rotate3d()

rotateX()

rotateY()

rotateZ()

row-gap

S

s

saturate()

scale

scale()

scale3d()

scaleX()

scaleY()

scaleZ()

:scope

scroll-behavior

scroll-margin

scroll-margin-block

scroll-margin-block-end

scroll-margin-block-start

scroll-margin-bottom

scroll-margin-inline

scroll-margin-inline-end

scroll-margin-inline-start

scroll-margin-left

scroll-margin-right

scroll-margin-top

scroll-padding

scroll-padding-block

scroll-padding-block-end

scroll-padding-block-start
scroll-padding-bottom
scroll-padding-inline
scroll-padding-inline-end
scroll-padding-inline-start
scroll-padding-left
scroll-padding-right
scroll-padding-top
scroll-snap-align
scroll-snap-stop
scroll-snap-type
scrollbar-color
scrollbar-width
::selection
selector()
sepia()
<shape>
shape-image-threshold
shape-margin
shape-outside
skew()
skewX()
skewY()
::slotted
speak-as (@counter-style)
src (@font-face)
steps()
<string>
@styleset
styleset()
@stylistic

stylistic()
suffix (@counter-style)
@supports
@swash
swash()
symbols (@counter-style)
symbols()
system (@counter-style)

T

tab-size
table-layout
:target
target-counter()
target-counters()
target-text()
text-align
text-align-last
text-combine-upright
text-decoration
text-decoration-color
text-decoration-line
text-decoration-style
text-decoration-thickness
text-emphasis
text-emphasis-color
text-emphasis-position
text-emphasis-style
text-indent
text-justify
text-orientation

text-overflow
text-rendering
text-shadow
text-transform
text-underline-offset
text-underline-position
<time>
<time-percentage>
<timing-function>
top
@top-center
touch-action
transform
transform-box
<transform-function>
transform-origin
transform-style
transition
transition-delay
transition-duration
transition-property
transition-timing-function
translate
translate()
translate3d()
translateX()
translateY()
translateZ()
turn

U

unicode-bidi
unicode-range (@font-face)
unset
<url>
url()
user-zoom (@viewport)

V

:valid
var()
vertical-align
vh
@viewport
visibility
:visited
vmax
vmin
vw

W

white-space
widows
width
width (@viewport)
will-change
word-break
word-spacing
word-wrap
writing-mode

X

x

Z

z-index

zoom (@viewport)

Others

--*

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-`)

[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.

[Sign up now](#)