Syntax

A CSS rule consists of a **selector** and a **declaration** block.

```
selector {
    property1: value1;
    property2: value2;
}
```

No space between the property value and its unit

Simple selectors

name { ... }	<name>
#*id* { ... }	<element id="*id*">
.*class* { ... }	<element class="*class*">
name.class { ... }	<name class="*class*">
* { ... }	all elements
selector1, *selector2*, ..., *selectorX* {...}	Grouping selector

Combinator selectors

Based on a specific relationship between them

x y { ... }	Descendant selector: all *y* elements inside *x* elements
x > y { ... }	Child selector: all *y* elements where the parent is a *x* elements
x + y { ... }	Adjacent sibling selector: the first *y* element placed immediately after *x* elements
x ~ y { ... }	General sibling selector: all *y* elements that are preceded by a *x* element

Pseudo-elements selectors

Part of an element

Syntax	```select or: :ps eud o-e lement { property: value; }```
selector::after	Insert content after every *selector* element Onlyapply to block-level elements
selector::before	Insert content before every *selector* element
selector::first-letter	Selects the first letter of every *selector* element
selector::first-line	Selects the first line of every *selector* element
selector::marker	Selects the markers of list items

Pseudo-elements selectors (cont)

selector::selection	Selects the portion of an element that is selected by a user

Only some limited types of properties apply to each of the pseudo-elements

Attribute selectors

Based on an attribute or attribute value	All elements with a target attribute
[*attribute*]	All elements with an *attribute* attribute
[*attribute* = "*value*"]	All elements with *attribute* = *value*
[*attribute* ~= "*value*"]	All elements with an *attribute* attribute that contains a space-separated list of words, one of which is *value*
[*attribute* |= "*value*"]	All elements with an *attribute* attribute value starting with *value*
selector[*attribute* ^= "*value*"]	All *selector* elements with an *attribute* attribute value starting with *value*
selector[*attribute* $= "*value*"]	All *selector* elements with an *attribute* attribute value ending with *value*
selector[*attribute* = "*value**"]	All *selector* elements with an *attribute* attribute value containing the substring *value*

Colors and background

color	*color* | initial|inherit
background	*bg-color bg-image position/bg-size bg-repeat bg-origin bg-clip bg-attachment* | initial | inherit
background-color	*color* | transparent [def] | initial | inherit

Colors and background (cont)

background-image	*url* | none [def] | initial | inherit
background-position	*value* [def: 0% 0%] | initial | inherit
background-position-y background-position-x	like background-position, but def: 0%
background-size	auto [def] | *length* | *percentage* | cover | contain | initial | inherit
background-repeat	repeat [def] | repeat-x | repeat-y | no-repeat | initial | inherit
background-clip	border-box [def] | padding-box | content-box | initial | inherit
background-origin	border-box | padding-box [def] | content-box | initial | inherit
background-attachment	scroll [def] | fixed | local | initial | inherit
background-blend-mode	normal [def] | multiply | screen | overlay | darken | lighten | color-dodge | saturation | color | luminosity

They are different ways to tell the colors: name \| hex \| rgb(*red, green, blue*) \| rgba(*red, green, blue, alpha*) \| hsl(*hue, saturation, lightness*) \| hsla(*hue, saturation, lightness, alpha*) \| hwb

Border

border	*border-width border-style* [req] *border-color* | initial | inherit
border-top/right/bottom/left	*border-left/right/top-width* [req] *border-left/right/top-style* [req] *border-left/right/top-color* | initial | inherit

Border (cont)

border-block/inline	*border-block/inline-width border-block/inline-style* [req] *border-block/inline-color* | initial | inherit
border-block/inline-end/start	*border-block/inline-end/start-width border-block/inline-end/start-style border-block/inline-end/start-color* | initial | inherit
width	[def: medium]
border-width	1 to 4 values: medium | thin | thick | *length* | initial | inherit To take effect, the *border-style* must be set
border-top/right/bottom/left-width	1 value: like border-width
border-block/inline-width	1 to 2 values: like border-width To take effect, the *border-block/inline-style* must be set
border-block/inline-end/start-width	1 value: like border-width To take effect, the *border-block/inline-end/start-style* must be set
style	[def: none]
border-style	1 to 4 values: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | initial | inherit
border-top/right/bottom/left-style	1 value: like border-style
border-block/inline-style	1 to 2 values: like border-style
border-block/inline-end/start-style	1 value: like border-style
color	[def: text color]

Border (cont)

border-color	1 to 4 values: *colors* | transparent | initial | inherit To take effect, the *border-style* must be set
border-top/r-ight/bottom/-left-color	1 value: like border-color To take effect, the *border-style* or *border-left/right/-top-style* must be set
border-bl-ock/inline-color	1 to 2 values: like border-color To take effect, the *border-block/inline-style* must be set
border-bl-ock/inline-e-nd/start-color	1 value: like border-color To take effect, the *border-block/inline-end/start-style* must be set
radius	[def: 0]
border-radius	1 to 4 values: *length* | *%* | initial | inherit
border-top/r-ight/bottom/-left-radius	1 value: like border-radius
border-end/s-tart-end/sta-rt-radius	1 to 2 values: like border-radius
image	
border-image	*border-image-source border-image-slice border-im-age-width border-image-outset border-image-repeat* | initial | inherit
border-im-age-slice	*number* | *%* [def: 100%] | fill | initial | inherit
border-im-age-source	none [def] | *image* | initial | inherit
border-im-age-width	*number* [def: 1] | *length* | *%* | auto | initial | inherit
border-im-age-outset	*length* [def: 0] | *number* | initial | inherit

Border (cont)

border-image-repeat	stretch [def] | repeat | round | space | initial | inherit
table	
border-collapse	separate [def] | collapse | initial | inherit
border-spacing	1 or 2 values: *length* | initial | inherit Only works when border-collapse is separate

Position

position	static [def] | absolute | fixed | relative | sticky | initial | inherit
bottom top right left	auto [def] | *length* | *%* | initial | inherit Negatives values are allowed different depending on the position property
clip-path	*clip-source* | *basic-shape* | margin-box | border-box | padding-box | content-box | fill-box | stroke-box | view-box | none | initial | inherit
z-index	auto [def] | *number* | initial | inherit = stack order of an element

Display

display	*value* [def: block or inline]
visibility	visible [def] | hidden | collapse | initial | inherit

Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there.

visibility:hidden; also hides an element. However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

Float

float	none [def] | left | right | initial | inherit
clear	none [def] | left | right | both | initial | inherit
box-sizing	content-box [def] | border-box | initial | inherit

List

list-style	*list-style-type list-style-position list-style-image* | initial | inherit
list-style-type	*value* [def: disc]
list-style-p-osition	inside | outside [def] | initial | inherit
list-style-image	none [def] | *url* | initial | inherit

Table

caption-side	top | bottom | initial | inherit
empty-cells	show | hide | initial | inherit
table-layout	auto [def] | fixed | initial | inherit

Units

0	unit can be omitted
Absolute Lengths	not recommended
cm, mm, in, px, pt, pc	
Relative Lengths	
em, ex, ch, rem, vw, vh, vmin, vmax, %	
A whitespace cannot appear between the number and the unit	

Column

columns	auto [def] | *column-width column-count* | initial | inherit
column-count	*number* | auto [def] | initial | inherit
column-width	auto [def] | *length* | initial | inherit
column-fill	balance [def] | auto | initial | inherit
column-span	none [def] | all | initial | inherit
column-rule	*column-rule-width column-rule-style column-rule--color* | initial | inherit
column-ru-le-width	medium [def] | thin | thick | *length* | initial | inherit

Column (cont)

column-rule-style	none [def] | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | initial | inherit
column-rule--color	*color* [def: current color] | initial | inherit
column-gap	*length* | normal | initial | inherit

Transition

transition	*transition-property transition-duration transition-timing--function transition-delay* | initial | inherit
transitio-n-p-roperty	none | all [def] | *property* | initial | inherit
transitio-n-d-uration	*time* [def: 0s] | initial | inherit
transitio-n-timi-ng-fun-ction	linear | ease [def] | ease-in | ease-out | ease-in-out | step-start | step-end | steps(int,start | end) | cubic--bezier(*n,n,n,n*) | initial | inherit
transitio-n-delay	*time* [def: 0s] | initial | inherit

Counter

content	normal [def] | none | counter | attr | *string* | open-quote | close-quote | no-open-quote | no-close-quote | *url* | initial | inherit Used with ::before and ::after
counter-incr-ement	none [def] | *id* | initial | inherit
counter-reset	none [def]| *id number* | initial | inherit
counter(*countername*, *counterstyle*)	

By **Manon Gerrd** (manongerard)

cheatography.com/manongerard/

Not published yet.
Last updated 30th October, 2023.
Page 4 of 8.

Icons

Font Awesome Icons	
Bootstrap Icons	
Google Icons	

Specificity Hierarchy

1. Inline styles
2. IDs
3. Classes, pseudo-classes, attribute selectors
4. Elements and pseudo-elements
!important can overide this hierachy

Insert CSS

external	inside <head>: <link rel="stylesheet" href="*mystyle.css*">
internal	inside <head>: inside <style>
inline	inside and element: using the style attribute

Pseudo-class selectors

Based on a certain state	

Syntax	`select or: pse udo -class {`
	`    property: value;`
	`}`

selector.active	The active link
selector.checked	All checked *selector* elements
selector.disabled	All disabled *selector* elements
selector.empty	All *selector* elements that has no children
selector.enabled	All enabled *selector* elements
selector.first-child	All *selector* elements that is the first child of its parent
selector.first-of-type	All *selector* elements that is the first *selector* element of its parent
selector.focus	The *selector* element that has focus
selector.hover	Links on mouse over
selector.in-range	*selector* elements with a value within a specified range
selector.invalid	All *selector* elements with an invalid value

Pseudo-class selectors (cont)

selector.lang(*language*)	All *selector* elements with a lang attribute value starting with *language*
selector.last-child	All *selector* elements that is the last child of its parent
selector.last-of-type	All *selector* elements that is the last *selector* element of its parent
selector.link	All unvisited links
selector.not(*selector*)	All elements that is not a *selector* element
selector.nth-child(*n*)	All *selector* elements that is the second child of its parent
selector.nth-last-child(*n*)	All *selector* elements that is the second child of its parent, counting from the last child
selector.nth-last-of-type(*n*)	All *selector* elements that is the second *selector* element of its parent, counting from the last child
selector.nth-of-type(*n*)	All *selector* elements that is the *n*th *selector* element of its parent
selector.only-of-type	All *selector* elements that is the only *selector* element of its parent
selector.only-child	All *selector* elements that is the only child of its parent
selector.optional	*selector* elements with no "required" attribute
selector.out-of-range	*selector* elements with a value outside a specified range
selector.read-only	*selector* elements with a "readonly" attribute specified
selector.read-write	*selector* elements with no "readonly" attribute
selector.required	*selector* elements with a "required" attribute specified

By **Manon Gerrd** (manongerard)

cheatography.com/manongerard/

Not published yet.
Last updated 30th October, 2023.
Page 5 of 8.

Pseudo-class selectors (cont)

selector:root	The document's root element
selector:target	The current active #news element (clicked on a URL containing that anchor name)
selector:valid	All *selector* elements with a valid value
selector:visited	All visited links

Margin

margin	1 to 4 values: *length* [def: 0] | *%* | auto [horizontally center] | initial | inherit;
margin-bottom/left/right/top	1 value: like margin
margin-block/inline	1 to 2 values: like margin, but auto by default
margin-block/inline-end/start	1 value: like margin-block/inline

Negative values are allowed
Margin Collapse: top and bottom (not left and right !) margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

Padding

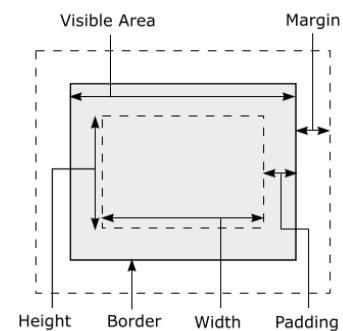|	
padding	1 to 4 values: *length* [def:0] | *%* | initial | inherit
padding-top/right/bottom/left	1 value: like padding
padding-block/inline	1 to 2 values: like padding, but def: auto
padding-block/inline--end/start	1 value: like padding-block/inline

Padding (cont)

box-sizing	content-box [def, border and padding not included]| border-box [content, padding and border included] | initial | inherit

Padding and Element Width: if an element has a specified width, the padding added to that element will be added to the total width of the element.
To keep the given width, no matter the amount of padding, you can use the box-sizing property. This causes the element to maintain its actual width; if you increase the padding, the available content space will decrease.

Box model

Height and width

height/width	*length* | *%* | auto [def] | initial | inherit min/max-height/width override the height/width property
max-height/width	*length* | *%* | none [def] | initial | inherit Improves the browser's handling of small windows
min-height/width	*length* | *%* | initial | inherit

The height and width properties do not include padding, borders, or margins.

Text

alignment and direction

text-align	left [def: if direction is ltr] | right [def: if direction is rtl] | center | justify | initial | inherit
text-align-last	auto [def] | left | right | center | justify | start | end | initial | inherit
direction	ltr [def] | rtl | initial | inherit
unicode-bidi	normal [def] | embed | bidi-override | initial | inherit
vertical--align	baseline [def] | length | sub | super | top | text-top | middle | bottom | text-bottom | initial | inherit

decoration

text-deco-ration	*text-decoration-line* [req] *text-decoration-color text-decoration-style text-decoration-thickness* | initial | inherit
text-deco-ration-line	none [def] | underline | overline | line-through | initial | inherit Can combine more than 1 value
text-deco-ration-color	*color* [def: *current color*] | initial | inherit
text-deco-ration-style	solid [def] | double | dotted | dashed | wavy | initial | inherit
text-deco-ration-th-ickness	auto [def] | from-font | *length/percentage* | initial | inherit

transformation

text-tran-sform	none [def] | capitalize | uppercase | lowercase | initial | inherit

text spacing

Text (cont)

text-i-ndent	*length* [def: 0] | *%* | initial | inherit Negative values are allowed
letter-sp-acing	normal [def] | *length* | initial | inherit Negative value also allowed
line-h-eight	normal [def] | *number* | *length* | *%* | initial | inherit
word-s-pacing	normal [def] | *length* | initial | inherit
white--space	normal [def] | nowrap | pre | pre-line | pre-wrap | initial | inherit

shadow

text-s-hadow	*h-shadow* [req] *v-shadow* [req] *blur-radius* [def: 0] *color* | none [def] | initial | inherit
text-j-ustify	auto [def] | inter-word | inter-character | none | initial | inherit
text-orie-ntation	mixed [def] | upright | sideways | sideways-right | use-glyph-orientation | initial | inherit Works only when writing-mode is set to vertical
text-o-verflow	clip [def] | ellipsis | *string* | initial | inherit requires white-space: nowrap; overflow: hidden

text-decoration: none; is used to remove the underline from links:

```
a { text-d eco ration: none; }
```

Font

font	*font-style font-variant font-weight font-size/line-height* [req] *font-family* [req] | caption | icon | menu | message-box | small-caption | status-bar | initial | inherit

Font (cont)

font-style	normal [def]	italic	oblique	initial	inherit
font-variant	normal [def]	small-caps	initial	inherit
font-weight	normal [def]	bold	bolder	lighter	number	initial	inherit
font-size	medium [def]	xx-small	x-small	small	large	x-large	xx-large	smaller	larger	length	initial	inherit
font-family	*family-name*	*generic-family*	initial	inherit Start with the font you want, *[then add backup fonts,]* and always end with a generic family. Separate each value with a comma.
font-feature-settings	normal [def]	feature-value *[1 / 0 / on / off]*
font-kerning	auto	normal	none
font-size-adjust	*number*	none	initial	inherit
font-stretch	ultra-condensed	extra-condensed	condensed	semi-condensed	normal [def]	semi-expanded	expanded	extra-expanded	ultra-expanded	initial	inherit
font-variant-caps	normal [def]	small-caps	all-small-caps	petite-caps	all-petite-caps	unicase	titling-caps	initial	inherit	unset
@font-face	font-family [req] src [req] font-stretch font-style font-weight unicode-range

Outline

outline	*outline-width outline-style* [def: invert] *outline-color*	initial	inherit
outline-width	medium [def]	thin	thick	*length*	initial	inherit Must Always declare outline-style before

Outline (cont)

outline-style	none [def]	hidden	dotted	dashed	solid	double	groove	ridge	inset	outset	initial	inherit
outline-color	color [def: *current color*]	initial	inherit Must Always declare outline-style before
outline-offset	*length* [def: 0]	initial	inherit

Math functions

calc(*expression*)
max(*value1*, *value2*, ..., *valueX*)
min(*value1*, *value2*, ..., *valueX*)

Overflow

overflow	visible [def]	hidden	clip	scroll	auto	initial	inherit
overflow-wrap	normal [def]	anywhere	break-word	initial	inherit
overflow-x/y	visible [def]	hidden	scroll	auto	initial	inherit
overflow-anchor	auto [def]	none	initial	inherit
Only works for block elements with a specified height	

Other

/* ... */	Comment
opacity	*number* [def: 1]	initial	inherit
Navigation bar	https://www.w3schools.com/css/css_navbar.asp
Dropdown	https://www.w3schools.com/css/css_dropdowns.asp
Dropdown are useful for addition info to help the users	