



CSS 3 cheatsheet

This is a quick reference cheat sheet for CSS goodness, listing selector syntax, properties, units and other useful bits of information.

Getting Started

<p>Introduction</p> <p>CSS is rich in capabilities and is more than simply laying out pages.</p> <p>External stylesheet</p> <pre><link href=".path/to/stylesheets/style.css" rel="stylesheet" type="text/css"></pre> <p>Internal stylesheet</p> <pre><style> body { background-color: linen; } </style></pre> <p>Inline styles</p> <pre><h2 style="text-align: center;">Centered text</h2> <p style="color: blue; font-size: 18px;">Blue, 18-point text</p></pre>	<p>Add class</p> <pre><div class="classname"></div> <div class="class1 ... classn"></div></pre> <p>Support multiple classes on one element.</p>	<p>Important</p> <pre>.post-title { color: blue !important; }</pre> <p>Overrides all previous styling rules.</p>
<p>Position</p> <pre>.box { position: relative; top: 20px; left: 20px; }</pre> <p>See also: Position</p>	<p>Background</p> <pre>background-color: blue; background-image: url("nyan-cat.gif"); background-image: url("../image.png");</pre> <p>See: Backgrounds</p>	<p>Font</p> <pre>.page-title { font-weight: bold; font-size: 30px; font-family: "Courier New"; }</pre> <p>See: Fonts</p>
		<p>Comment</p> <pre>/* This is a single line comment */ /* This is a multi-line comment */</pre>

<p>Flex layout</p> <pre>div { display: flex; justify-content: center; } div { display: flex; justify-content: flex-start; }</pre> <p>See: Flexbox Flex Tricks</p>	<p>Grid layout</p> <pre>#container { display: grid; grid: repeat(2, 60px) / auto-flow 80px; } #container > div { background-color: #8ca0ff; width: 50px; height: 50px; }</pre> <p>See: Grid Layout</p>	<p>Variable & Counter</p> <pre>counter-set: subsection; counter-increment: subsection; counter-reset: subsection 0; :root { --bg-color: brown; } element { background-color: var(--bg-color); }</pre> <p>See: Dynamic content</p>
--	---	---

CSS Selectors

Examples	Basic	Combinators
Groups Selector	* All elements	div.classname Div with certain classname
<pre>h1, h2 { color: red; }</pre>	div All div tags	div#idname Div with certain ID
Chaining Selector	.classname All elements with class	div p Paragraphs inside divs
<pre>h3.section-heading { color: blue; }</pre>	#idname Element with ID div,p All divs and paragraphs #idname * All elements inside #idname	div > p All p tags one level deep in div div + p P tags immediately after div

Attribute Selector
<pre>div[attribute="SomeValue"] { background-color: red; }</pre>
First Child Selector
<pre>p:first-child { font-weight: bold; }</pre>
No Children Selector
<pre>.box:empty { background: lime; height: 80px; width: 80px; }</pre>

See also: Type / Class / ID / Universal selectors	
Attribute selectors	
a[target]	With a target attribute
a[target="_blank"]	Open in new tab
a[href^="/index"]	Starts with /index
[class]="chair"]	Starts with chair
[class*="chair"]	containing chair
[title~="chair"]	Contains the word chair
a[href\$=".doc"]	Ends with doc
[type="button"]	Specified type
See also: Attribute selectors	

div ~ p	
See also: Adjacent / Sibling / Child selectors	
User action pseudo classes	
a:link	Link in normal state
a:active	Link in clicked state
a:hover	Link with mouse over it
a:visited	Visited link

Pseudo classes	
p::after	Add content after p
p::before	Add content before p
p::first-letter	First letter in p
p::first-line	First line in p
::selection	Selected by user
::placeholder	Placeholder attribute
:root	Documents root element
:target	Highlight active anchor
div:empty	Element with no children
p:lang(en)	P with en language attribute
:not(span)	Element that's not a span

Input pseudo classes	
input:checked	Checked inputs
input:disabled	Disabled inputs
input:enabled	Enabled inputs
input:focus	Input has focus
input:in-range	Value in range
input:out-of-range	Input value out of range
input:valid	Input with valid value
input:invalid	Input with invalid value
input:optional	No required attribute
input:required	Input with required attribute
input:read-only	With readonly attribute
input:read-write	No readonly attribute
input:indeterminate	With indeterminate state

Structural pseudo classes	
p:first-child	First child
p:last-child	Last child
p:first-of-type	First of some type
p:last-of-type	Last of some type
p:nth-child(2)	Second child of its parent
p:nth-child(3n+2)	Nth-child (an + b) formula
p:nth-last-child(2)	Second child from behind
p:nth-of-type(2)	Second p of its parent
p:nth-last-of-type(2)	...from behind
p:only-of-type	Unique of its parent
p:only-child	Only child of its parent

CSS Fonts

Properties	
font-family:	
font-size:	<size>
letter-spacing:	<size>
line-height:	<number>
font-weight:	<number> / bold / normal
font-style:	italic / normal
text-decoration:	underline / none
text-align:	left / right center / justify
text-transform:	capitalize / uppercase / lowercase
See also: Font	

Shorthand					
font:	italic	400	14px	/	1.5
	style	weight	size (required)	line-height	family (required)
Example					
<pre>font-family: Arial, sans-serif; font-size: 12pt; letter-spacing: 0.02em;</pre>					

Case	
/* Hello */	text-transform: capitalize;
/* HELLO */	text-transform: uppercase;
/* hello */	text-transform: lowercase;

CSS Colors

Named color	
color: red;	
color: orange;	
color: tan;	
color: rebeccapurple;	

Hexadecimal color	
color: #090;	
color: #009900;	
color: #090a;	
color: #009900aa;	

rgb() Colors	
color: rgb(34, 12, 64, 0.6);	
color: rgba(34, 12, 64, 0.6);	
color: rgb(34 12 64 / 0.6);	
color: rgba(34 12 64 / 0.3);	
color: rgb(34.0 12 64 / 60%);	
color: rgba(34.6 12 64 / 30%);	

HSL Colors

```
color: hsl(30, 100%, 50%, 0.6);
color: hsla(30, 100%, 50%, 0.6);
color: hsl(30 100% 50% / 0.6);
color: hsla(30 100% 50% / 0.6);
color: hsl(30.0 100% 50% / 60%);
color: hsla(30.2 100% 50% / 60%);
```

Other

```
color: inherit;
color: initial;
color: unset;
color: transparent;
color: currentcolor; /* keyword */
```

CSS Backgrounds

Properties	
<code>background:</code>	(Shorthand)
<code>background-color:</code>	See: Colors
<code>background-image:</code>	<code>url(...)</code>
<code>background-position:</code>	left/center/right top/center/bottom
<code>background-size:</code>	cover X Y
<code>background-clip:</code>	border-box padding-box content-box
<code>background-repeat:</code>	no-repeat repeat-x repeat-y
<code>background-attachment:</code>	scroll/fixed/local

Shorthand
<code>background: #ff0 url(a.jpg) left top / 100px auto no-repeat fixed;</code>
<code>background: #abc url(b.png) center center / cover repeat-x local;</code>
color image posX posY size repeat attach..

Examples
<code>background: url(img_man.jpg) no-repeat center;</code>
<code>background: url(img_flwr.gif) right bottom no-repeat, url(paper.gif) left top repeat;</code>
<code>background: rgb(2,0,36);</code> <code>background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(13,232,230,1) 35%, rgba(0,212,255,1) 100%);</code>

CSS The Box Model

Maximums/Minimums
<code>.column {</code> <code>max-width: 200px;</code> <code>width: 500px;</code> }

See also: [max-width](#) / [min-width](#) / [max-height](#) / [min-height](#)

Margin / Padding
<code>.block-one {</code> <code>margin: 20px;</code> <code>padding: 10px;</code> }

See also: [Margin](#) / [Padding](#)

Box-sizing
<code>.container {</code> <code>box-sizing: border-box;</code> }

See also: [Box-sizing](#)

Visibility
<code>.invisible-elements {</code> <code>visibility: hidden;</code> }

See also: [Visibility](#)

Auto keyword
<code>div {</code> <code>margin: auto;</code> }

See also: [Margin](#)

Overflow
<code>.small-block {</code> <code>overflow: scroll;</code> }

See also: [Overflow](#)

CSS Animation

Shorthand							
name	duration	timing-function	delay	count	direction	fill-mode	play-state

Properties
<code>animation:</code> (shorthand)
<code>animation-name:</code> <name>
<code>animation-duration:</code> <time>ms
<code>animation-timing-function:</code> ease / linear / ease-in / ease-out / ease-in-out
<code>animation-delay:</code> <time>ms
<code>animation-iteration-count:</code> infinite / <number>
<code>animation-direction:</code> normal / reverse / alternate / alternate-reverse
<code>animation-fill-mode:</code> none / forwards / backwards / both / running / initial

Example
<code>/* @keyframes duration timing-function delay iteration-count direction fill-mode play-state name */</code>
<code>animation: 3s ease-in 1s 2 reverse both paused slidein;</code>
<code>/* @keyframes duration timing-function delay name */</code>
<code>animation: 3s linear 1s slidein;</code>
<code>/* @keyframes duration name */</code>
<code>animation: 3s slidein;</code>
<code>animation: 4s linear 0s infinite alternate move_eye;</code> <code>animation: bounce 300ms linear 0s infinite normal;</code> <code>animation: bounce 300ms linear infinite;</code> <code>animation: bounce 300ms linear infinite alternate-reverse;</code> <code>animation: bounce 300ms linear 2s infinite alternate-reverse forwards normal;</code>

both / initial / inherit		
animation-play-state:	normal / reverse / alternate / alternate-reverse	Javascript Event
See also: Animation		

CSS Flexbox

Simple example <pre>.container { display: flex; } .container > div { flex: 1 1 auto; }</pre>	Container <pre>.container { display: flex; display: inline-flex; flex-direction: row; /* ltr - default */ flex-direction: row-reverse; /* rtl */ flex-direction: column; /* top-bottom */ flex-direction: column-reverse; /* bottom-top */ flex-wrap: nowrap; /* one-line */ flex-wrap: wrap; /* multi-line */ align-items: flex-start; /* vertical-align to top */ align-items: flex-end; /* vertical-align to bottom */ align-items: center; /* vertical-align to center */ align-items: stretch; /* same height on all (default) */ justify-content: flex-start; /* [xxx] */ justify-content: center; /* [xxx] */ justify-content: flex-end; /* [xxx] */ justify-content: space-between; /* [x x x] */ justify-content: space-around; /* [x x x] */ justify-content: space-evenly; /* [x x x] */ }</pre>
Child <pre>.container > div { /* This: */ flex: 1 0 auto; /* Is equivalent to this: */ flex-grow: 1; flex-shrink: 0; flex-basis: auto; order: 1; align-self: flex-start; /* left */ margin-left: auto; /* right */ }</pre>	

CSS Flexbox Tricks

Vertical center <pre>.container { display: flex; } .container > div { width: 100px; height: 100px; margin: auto; }</pre>	Vertical center (2) <pre>.container { display: flex; /* vertical */ align-items: center; /* horizontal */ justify-content: center; }</pre>	Reordering <pre>.container > .top { order: 1; } .container > .bottom { order: 2; }</pre>
Mobile layout <pre>.container { display: flex; flex-direction: column; } .container > .top { flex: 0 0 100px; } .container > .content { flex: 1 0 auto; }</pre> <p>A fixed-height top bar and a dynamic-height content area.</p>	Table-like <pre>.container { display: flex; /* the 'px' values here are just suggested percentages */ .container > .checkbox { flex: 1 0 20px; } .container > .subject { flex: 1 0 400px; } .container > .date { flex: 1 0 120px; }</pre> <p>This creates columns that have different widths, but size accordingly according to the circumstances.</p>	

Vertical <pre>.container { align-items: center; }</pre> <p>Vertically-center all items.</p>	Left and right <pre>.menu > .left { align-self: flex-start; } .menu > .right { align-self: flex-end; }</pre>
---	--

CSS Grid Layout

<p>Grid Template Columns</p> <pre>#grid-container { display: grid; width: 100px; grid-template-columns: 20px 20% 60%; }</pre>	<p>fr Relative Unit</p> <pre>.grid { display: grid; width: 100px; grid-template-columns: 1fr 60px 1fr; }</pre>	<p>Grid Gap</p> <pre>/*The distance between rows is 20px*/ /*The distance between columns is 10px*/ #grid-container { display: grid; grid-gap: 20px 10px; }</pre>
<p>CSS Block Level Grid</p> <pre>#grid-container { display: block; }</pre>	<p>CSS grid-row</p> <p>CSS syntax:</p> <ul style="list-style-type: none">grid-row: grid-row-start / grid-row-end; <p>Example</p> <pre>.item { grid-row: 1 / span 2; }</pre>	<p>CSS Inline Level Grid</p> <pre>#grid-container { display: inline-grid; }</pre>
<p>minmax() Function</p> <pre>.grid { display: grid; grid-template-columns: 100px minmax(100px, 500px) 100px; }</pre>	<p>grid-row-start & grid-row-end</p> <p>CSS syntax:</p> <ul style="list-style-type: none">grid-row-start: auto row-line;grid-row-end: auto row-line span n; <p>Example</p> <pre>grid-row-start: 2; grid-row-end: span 2;</pre>	<p>CSS grid-row-gap</p> <pre>grid-row-gap: length;</pre> <p>Any legal length value, like px or %. 0 is the default value</p>
<p>CSS grid-area</p> <pre>.item1 { grid-area: 2 / 1 / span 2 / span 3; }</pre>	<p>Justify Items</p> <pre>#container { display: grid; justify-items: center; grid-template-columns: 1fr; grid-template-rows: 1fr 1fr 1fr; grid-gap: 10px; }</pre>	<p>CSS grid-template-areas</p> <pre>.item { grid-area: nav; } .grid-container { display: grid; grid-template-areas: 'nav nav . .' 'nav nav . .'; }</pre>
<p>Justify Self</p> <pre>#grid-container { display: grid; justify-items: start; } .grid-items { justify-self: end; }</pre> <p>The grid items are positioned to the right (end) of the row.</p>	<p>Align Items</p> <pre>#container { display: grid; align-items: start; grid-template-columns: 1fr; grid-template-rows: 1fr 1fr 1fr; grid-gap: 10px; }</pre>	

CSS Dynamic Content

<p>Variable</p> <p>Define CSS Variable</p> <pre>:root { --first-color: #16f; --second-color: #ff7; }</pre> <p>Variable Usage</p> <pre>#firstParagraph { background-color: var(--first-color); color: var(--second-color); }</pre> <p>See also: CSS Variable</p>	<p>Counter</p> <pre>/* Set "my-counter" to 0 */ counter-set: my-counter; /* Increment "my-counter" by 1 */ counter-increment: my-counter; /* Decrement "my-counter" by 1 */ counter-increment: my-counter -1; /* Reset "my-counter" to 0 */ counter-reset: my-counter;</pre> <p>See also: Counter set</p>	<p>Using counters</p> <pre>body { counter-reset: section; } h3::before { counter-increment: section; content: "Section." counter(section); } ol { counter-reset: section; list-style-type: none; } li::before { counter-increment: section; content: counters(section, ".") " "; }</pre>
--	---	--

Css 3 tricks

Scrollbar smooth

```
html { scroll-behavior: smooth; }
```

Click me, the page will scroll smoothly to Getting started

Also see

[CSS selectors cheatsheet](#) (frontend30.com)[MDN: Using CSS flexbox](#)[Ultimate flexbox cheatsheet](#)[GRID: A simple visual cheatsheet](#)[CSS Tricks: A Complete Guide to Grid](#)[Browser support](#)

Related Cheatsheet

[HTML Cheatsheet](#)
Quick Reference[JavaScript Cheatsheet](#)
Quick Reference

Recent Cheatsheet

[Google Search Cheatshee](#)
Quick Reference[Kubernetes Cheatsheet](#)
Quick Reference[jQuery Cheatsheet](#)
Quick Reference[Laravel Cheatsheet](#)
Quick Reference[ES6 Cheatsheet](#)
Quick Reference[ASCII Code Cheatsheet](#)
Quick Reference

Share quick reference and cheat sheet for developers.

[中文版 #Notes](#)

Ditch content management responsibilities and get back to coding.

ADS VIA CARBON

SPONSOR [Descope — Create authentication and user journeys for your app in a no-code editor. Get set up in minutes.](#)