

The CSS Cheat Sheet

WebsiteSetup.org

How to add CSS

CSS styles HTML: color, spacing, layout, type, and motion. Three ways to attach it, in order of preference.

```
<p style="color:#d97757">inline (tests only)</p>

<style> p { color:#1f2430; } </style>    /* internal */

<link rel="stylesheet" href="style.css">  /* external, preferred */
```

Selectors

Selector	Matches
p	Element type
.box	Class
#nav	Id
*	Universal
a b	Descendant
a > b	Direct child
a + b	Adjacent sibling
a ~ b	General sibling
[type="text"]	Attribute
h1, h2	Grouping

Pseudo-classes and pseudo-elements

Selector	Purpose
:hover :focus :active	Interaction states
:first-child :last-child	Position in parent
:nth-child(2n)	Every second child
:not() :is()	Negate / selector list
:where()	Like :is(), 0 specificity
:has()	Parent selector (Baseline Dec 2023)
::before ::after	Injected, need content: ""
::first-line ::selection	Text parts
::placeholder	Input placeholder

Box model and box-sizing

Layers: content, padding, border, margin. Border-box includes padding and border in width. Apply it globally.

```
* { box-sizing: border-box; }
.card { width: 300px; padding: 20px; border: 1px solid #ccc; }
```

Units

Unit	Relative to
px	Absolute pixels
em	Element font size (compounds)
rem	Root font size (preferred for type)
%	Parent value
vw vh	1% of viewport
vmin vmax	Smaller / larger viewport side
ch	Width of "0"

Colors

```
color: #333;
color: rgb(255 0 0 / .5);
color: hsl(210 50% 40%);
color: oklch(70% 0.15 250);
```

Typography

```
font-family: system-ui, Arial, sans-serif;
font-size: 1rem; font-weight: 400; line-height: 1.5;
letter-spacing: 0.01em; text-align: left;
text-decoration: underline; text-transform: uppercase;
```

Backgrounds and gradients

```
background-color: #1f2430;
background-image: url(photo.jpg);
background-size: cover;
background-position: center;
background-repeat: no-repeat;
background: linear-gradient(90deg, #000, #fff);
```

Flexbox (one dimension)

```
.bar {
  display: flex;
  flex-direction: row;          /* or column */
  justify-content: space-between; /* start|center|space-around */
  align-items: center;          /* start|stretch */
  gap: 1rem;
  flex-wrap: wrap;
}
.bar .grow { flex: 1; }        /* child takes leftover space */
```

Grid (two dimensions)

```
.grid { display: grid; grid-template-columns: repeat(3, 1fr); gap: 1rem; }

/* responsive, no media query */
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

.feature { grid-column: 1 / 3; } /* span columns */
grid-template-areas: "header header" "sidebar main";
```

Positioning

```
.menu { position: sticky; top: 0; z-index: 10; }
.badge { position: absolute; top: 8px; right: 8px; }
.dialog { position: fixed; inset: 0; }
/* static | relative | absolute | fixed | sticky */
```

Display, visibility, overflow

```
display: block | inline | inline-block | none;
visibility: hidden; /* invisible but keeps space */
overflow: auto; /* hidden | scroll */
```

Transitions, animations, transforms

```
.btn { transition: all .3s ease; }
.btn:hover { transform: scale(1.05); }

@keyframes spin { from { transform: rotate(0); } to { transform: rotate(360deg); } }
.loader { animation: spin 2s linear infinite; }

transform: translate(10px, 0) rotate(5deg);
```

Responsive design

```
/* mobile-first */
@media (min-width: 768px) { .grid { grid-template-columns: repeat(4, 1fr); } }

/* container query, Baseline 2023 */
.wrapper { container-type: inline-size; }
@container (min-width: 400px) { .card { flex-direction: row; } }
```

Custom properties (variables)

```
:root { --brand: #0a7; --space: 1rem; }
.button { color: var(--brand, #000); padding: var(--space); }
```

Modern functions

```
width: calc(100% - 2rem);
font-size: clamp(1rem, 2vw, 2rem);
width: min(90%, 1200px);
padding: max(1rem, 3vw);
```

Modern CSS (2023 to 2026)

```
.card { & .title { font-weight: 700; } } /* native nesting */
@layer base, utils; /* cascade layers */
margin-inline: auto; padding-block: 1rem; inset: 0; /* logical props */
/* :has() parent selector is safe in 2026 */
```

Sass and SCSS

Preprocessor compiled to CSS. SCSS (.scss) is a superset of CSS. @use / @forward replace the deprecated @import.

```
$primary: #333;
.nav { color: $primary; &:hover { color: lighten($primary, 20%); } }

@use 'buttons';
@forward 'theme';

@mixin flex-center { display: flex; justify-content: center; align-items: center; }
.hero { @include flex-center; }

@function double($n) { @return $n * 2; }
@each $n, $c in (ok: green, err: red) { .badge-#{$n} { background: $c; } }
@for $i from 1 through 3 { .col-#{$i} { width: 100% / 3 * $i; } }
```

Native nesting and custom properties now cover some Sass jobs, but Sass still adds mixins, functions, and loops.

Gotchas

- Set `* { box-sizing: border-box; }` or widths will not add up.
- Specificity: inline > id > class > element. Keep `!important` last resort.
- `rem` is root-relative and stable; `em` compounds when nested.
- Flexbox is 1D, Grid is 2D. Pick by the shape of the problem.
- `:has()` and container queries are safe to ship in 2026.
- Sass `@import` is deprecated; use `@use` / `@forward`.
- Prefer mobile-first `min-width` queries.