

CSS Quick Reference

Selectors, layout, flexbox, grid, animation, responsive design

Selectors

Basic Selectors	
*	Universal — all elements
div	Type — all <code><div></code> elements
.class	Class — elements with class
#id	ID — element with id
[attr]	Attribute — has attribute
[attr="val"]	Attribute equals value

Combinators

A B	Descendant (any depth)
A > B	Direct child only
A + B	Adjacent sibling (next)
A ~ B	General sibling (all after)

Pseudo-classes

:hover	Mouse over element
:focus	Element has focus
:first-child	First child of parent
:nth-child(n)	Nth child (1-based, odd, even, 2n+1)
:not(sel)	Negation — exclude matches
:has(sel)	Parent selector (contains match)

Pseudo-elements

::before	Insert content before element
::after	Insert content after element
::placeholder	Style input placeholder text
::selection	Style highlighted text

Box Model

Box Sizing
<pre>/* Include padding/border in width */ *, ::before, ::after { box-sizing: border-box; }</pre>

Properties

margin	Space outside border
border	Edge around padding
padding	Space inside border
width / height	Content dimensions
outline	Ring outside margin (no space)

Shorthand

<pre>margin: 10px; /* all sides */ margin: 10px 20px; /* vertical horizontal */ margin: 10px 20px 30px; /* top horiz bottom */ margin: 10px 20px 30px 40px; /* T R B L */</pre>

Flexbox

Container
<pre>.flex { display: flex; justify-content: center; /* main axis */ align-items: center; /* cross axis */ gap: 1rem; }</pre>

Container Properties

flex-direction	row column row-reverse column-reverse
flex-wrap	nowrap wrap wrap-reverse
justify-content	flex-start center space-between space-around space-evenly
align-items	stretch center flex-start flex-end baseline
align-content	Multi-line cross axis alignment
gap	Space between flex items

Item Properties

flex: 1	Grow to fill available space
flex: 0 0 200px	Fixed width, no grow/shrink
align-self	Override align-items for one item
order	Change visual order (default 0)

Grid

Container
<pre>.grid { display: grid; grid-template-columns: repeat(3, 1fr); grid-template-rows: auto 1fr auto; gap: 1rem; }</pre>

Container Properties

grid-template-columns	Define column tracks: 1fr 2fr, repeat(3, 1fr)
grid-template-rows	Define row tracks
grid-template-areas	Named areas: " header header" "nav main"
gap	Row and column gaps
justify-items	Align items inline (horizontal)
align-items	Align items block (vertical)

Item Placement

<pre>.item { grid-column: 1 / 3; /* span cols 1-2 */ grid-row: 1 / -1; /* span all rows */ grid-area: header; /* named area */ }</pre>
--

Typography

Font Properties	
font-family	Typeface stack: 'Inter', sans-serif
font-size	Size: 1rem, 16px, clamp(1rem, 2vw, 2rem)
font-weight	normal (400) bold (700) 100-900
font-style	normal italic oblique
line-height	Line spacing: 1.5 (unitless recommended)
letter-spacing	Character spacing: 0.05em

Text Properties

text-align	left center right justify
text-decoration	none underline line-through
text-transform	uppercase lowercase capitalize
text-overflow	ellipsis (with overflow: hidden)
white-space	nowrap pre pre-wrap
word-break	break-all break-word

Colors & Backgrounds

Color Formats
<pre>color: #fff660; /* hex */ color: rgb(255, 102, 0); /* rgb */ color: hsl(24, 100%, 50%); /* hsl */ color: oklch(70% 0.15 50); /* oklch */</pre>

Background Properties

background-color	Solid fill: #f0f0f0
background-image	url(img.jpg) or gradient
background-size	cover contain 100px 200px
background-position	center top right 50% 50%
background-repeat	no-repeat repeat-x repeat-y

Gradients

<pre>background: linear-gradient(to right, #f00, #00f); background: radial-gradient(circle, #fff, #000); background: conic-gradient(red, yellow, green);</pre>
--

Transitions & Animation

Transitions
<pre>.btn { transition: background 0.3s ease, transform 0.2s; } .btn:hover { background: #0056b3; transform: scale(1.05); }</pre>

Transition Properties

transition-property	Which property to animate
transition-duration	Length: 0.3s, 300ms
transition-timing-function	ease linear ease-in-out cubic-bezier()
transition-delay	Wait before starting

Keyframe Animation

<pre>@keyframes spin { from { transform: rotate(0deg); } to { transform: rotate(360deg); } } .icon { animation: spin 1s linear infinite; }</pre>
--

Animation Properties

animation-name	Reference @keyframes name
animation-duration	Length of one cycle
animation-iteration-count	1 3 infinite
animation-direction	normal alternate reverse
animation-fill-mode	forwards backwards both

Responsive Design

Media Queries
<pre>@media (max-width: 768px) { .sidebar { display: none; } } @media (prefers-color-scheme: dark) { body { background: #1a1a1a; color: #eee; } }</pre>

CSS Quick Reference

Common Breakpoints

max-width: 480px	Mobile phones
max-width: 768px	Tablets
max-width: 1024px	Small laptops
max-width: 1280px	Desktops

Viewport Units

vw / vh	% of viewport width / height
dvh	Dynamic viewport height (mobile-safe)
svh / lvh	Small / large viewport height
cqi	Container query inline size

Container Queries

```
.card-wrapper { container-type: inline-size; }
@container (min-width: 400px) {
  .card { flex-direction: row; }
}
```

Positioning

Position Values

static	Default — normal document flow
relative	Offset from normal position; keeps space
absolute	Positioned to nearest positioned ancestor
fixed	Positioned to viewport; stays on scroll
sticky	Toggles relative/fixed based on scroll

Centering Patterns

```
/* Flex center */
display: flex;
justify-content: center;
align-items: center;

/* Grid center */
display: grid;
place-items: center;
```

Stacking

z-index	Stack order (higher = on top); needs position
isolation: isolate	Creates new stacking context

Custom Properties

Define & Use

```
:root {
  --color-primary: #3b82f6;
  --spacing-md: 1rem;
}
.btn {
  background: var(--color-primary);
  padding: var(--spacing-md);
}
```

Fallback Values

```
color: var(--accent, #ff6600);
/* uses #ff6600 if --accent is not defined */
```

Dynamic Theming

```
[data-theme="dark"] {
  --bg: #1a1a2e;
  --text: #e0e0e0;
}
body { background: var(--bg); color: var(--text); }
```