

GarraTTs CSS Cheatsheet

A modern CSS reference for clean, responsive styling. Focused on layout systems, typography, reusable tokens, and blog friendly patterns.

Selectors and Specificity

Selector	Use	Example
<code>.class</code>	Reusable styling hook	<code>.card { padding: 16px; }</code>
<code>#id</code>	Unique element (use sparingly)	<code>#header { position: sticky; top: 0; }</code>
<code>element</code>	Base element styling	<code>article p { line-height: 1.7; }</code>
<code>[attr=value]</code>	Target attributes	<code>a[rel~="nofollow"] { opacity: 0.85; }</code>
<code>:hover/:focus</code>	Interactive states	<code>a:hover, a:focus { text-decoration: underline; }</code>

Box Model and Sizing

Property	Use	Example
<code>box-sizing</code>	Predictable sizing	<code>*, *::before, *::after { box-sizing: border-box; }</code>
<code>margin/padding</code>	Spacing	<code>.section { padding: 24px; margin: 0 auto; }</code>
<code>max-width</code>	Readable line length	<code>.prose { max-width: 70ch; }</code>
<code>aspect-ratio</code>	Stable media blocks	<code>.thumb { aspect-ratio: 16 / 9; }</code>

Typography

Goal	Property	Example
Readability	<code>line-height</code>	<code>body { line-height: 1.7; }</code>
Scale	<code>clamp()</code>	<code>h1 { font-size: clamp(2rem, 4vw, 3rem); }</code>
Tracking	<code>letter-spacing</code>	<code>.kicker { letter-spacing: 0.08em; text-transform: uppercase; }</code>
Code blocks	<code>font-family</code>	<code>code, pre { font-family: ui-monospace, SFMono-Regular, Menlo, monospace; }</code>

Flexbox Patterns

Pattern	Use	Example
Row with gap	Simple horizontal layout	<code>.row { display: flex; gap: 16px; }</code>
Space between	Logo + nav	<code>.bar { display: flex; justify-content: space-between; align-items: center; }</code>
Wrap cards	Responsive card grid	<code>.cards { display: flex; flex-wrap: wrap; gap: 16px; }</code>
Centering	Center content	<code>.center { display: flex; align-items: center; justify-content: center; }</code>

Grid Layout

Pattern	Use	Example
Auto fit columns	Responsive grid	<code>.grid { display: grid; gap: 16px; grid-template-columns: repeat(auto-fit, minmax(240px, 1fr)); }</code>
Sidebar layout	Content + aside	<code>.layout { display: grid; grid-template-columns: 1fr 320px; gap: 24px; }</code>
Place items	Align within grid	<code>.grid { place-items: center; }</code>

Responsive Design

Tool	Use	Example
Media queries	Breakpoint tweaks	<code>@media (max-width: 768px) { .layout { grid-template-columns: 1fr; } }</code>
Fluid spacing	Less breakpoint work	<code>.wrap { padding: clamp(16px, 3vw, 32px); }</code>
Images	Prevent overflow	<code>img { max-width: 100%; height: auto; }</code>

CSS Variables and Tokens

Concept	Use	Example
Define tokens	One source of truth	<code>:root { --bg: #1e2020; --card: #353535; --text: #dce8f0; }</code>
Use tokens	Consistent theme	<code>body { background: var(--bg); color: var(--text); }</code>
Component vars	Local overrides	<code>.card { --pad: 16px; padding: var(--pad); }</code>

Dark Mode

Approach	Use	Example
prefers-color-scheme	Auto dark mode	<code>@media (prefers-color-scheme: dark) { :root { --bg: #1e2020; } }</code>
data-theme	User toggle	<code>html[data-theme="dark"] { --bg: #1e2020; }</code>
Color contrast	Readable UI	<code>a { color: var(--accent); } a:focus { outline: 2px solid var(--accent2); }</code>

Effects Without Overdoing It

Effect	Use	Example
Shadows	Depth	<code>.card { box-shadow: 0 10px 30px rgba(0,0,0,0.25); }</code>
Transitions	Smooth hover	<code>a { transition: color 150ms ease; }</code>
Focus styles	Keyboard UX	<code>:focus-visible { outline: 2px solid var(--accent2); outline-offset: 2px; }</code>

Performance and Cleanliness

Tip	Why	Example
Prefer classes	Stable + reusable	<code>.btn.primary { ... }</code>
Limit deep selectors	Faster + clearer	<code>.nav a { ... }</code> (not <code>header .wrap .nav ul li a</code>)
Use modern layout	Less hacks	<code>grid/flex + gap</code> (avoid margins for spacing)