

Complete CSS3 CheatSheet

By: Paul Carlo Tordecilla

Selector

- Universal: *
- Element: div, p, h1
- Class: .container
- ID: #header
- Attribute:
 - [href]
 - [target="_blank"]
- Pseudo-classes:
 - :hover, :active, :focus
 - :nth-child(n), :first-child, :last-child
 - :not(.class)
- Pseudo-elements:
 - ::before, ::after
 - ::first-letter, ::first-line
- Combinators:
 - Descendant: div p
 - Child: ul > li
 - Adjacent: h1 + p
 - General: h2 ~ p

BOX MODEL

- box-sizing: border-box;
- width: 100px | 50% | auto;
- height: auto | 100vh;
- margin: 10px 5% 0 auto; /* TRBL */
- padding: 1rem 2rem;
- border: 2px solid #f00;
- border-radius: 8px | 50%;
- box-shadow: 3px 3px 0px rgba(0,0,0,0.3);

FLEXBOX

display: flex | inline-flex;
flex-direction: row | column | row-reverse | column-reverse;
flex-wrap: nowrap | wrap | wrap-reverse;
justify-content: flex-start | center | space-between | space-around;
align-items: stretch | center | flex-start | baseline;
align-content: space-around | stretch;
gap: 10px | 1rem;

/* Child properties */
flex: 1 0 auto; /* grow shrink basis */
align-self: auto | flex-end;
order: 2;

GRID

display: grid | inline-grid;
grid-template-columns: 1fr 2fr 1fr | repeat(3, 100px);
grid-template-rows: auto 200px;
grid-template-areas: "header header" "sidebar main";
gap: 20px | 1rem;
place-items: center stretch;

/* Child properties */
grid-column: 1 / 3 | span 2;
grid-row: 2 / 4;
grid-area: header;
align-self: end;
justify-self: center;

TRANSITIONS & TRANSFORMS

/* Transitions */
transition: all 0.3s ease-in-out;
transition-property: transform, opacity;
transition-duration: 400ms;
transition-timing-function: cubic-bezier(0.4, 0, 0.2, 1);

/* Transforms */
transform: rotate(15deg) scale(1.2) translateX(20px);
transform-origin: top left | 30% 80%;
transform-style: preserve-3d;
perspective: 1000px;

ANIMATIONS

```
@keyframes slidein {  
  0% { transform: translateX(-100%); }  
  100% { transform: translateX(0); }  
}  
  
animation: slidein 0.5s ease-out 0.2s 3 alternate forwards;  
animation-name: pulse;  
animation-duration: 2s;  
animation-delay: 1s;  
animation-iteration-count: infinite;  
animation-fill-mode: both;
```

TYPOGRAPHY

font-family: "Segoe UI", system-ui, sans-serif;
font-size: 1.125rem | clamp(1rem, 2vw, 1.5rem);
font-weight: 700 | normal;
line-height: 1.6;
text-align: justify | start | end;
text-transform: uppercase | capitalize;
text-shadow: 1px 1px 2px #000;
letter-spacing: 0.05em;
word-spacing: 0.1em;
white-space: nowrap | pre-wrap;
overflow-wrap: break-word;

RESPONSIVE DESIGN

```
/* Media Queries */  
@media (max-width: 768px) { /* Mobile styles */ }  
@media (min-width: 1024px) and (orientation: landscape) { /* Desktop */ }
```

```
/* Units */  
width: 100vw; /* Viewport width */  
height: 50vh; /* Viewport height */  
font-size: calc(16px + 0.5vw); /* Fluid typography */
```

```
/* Layout */  
.container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));  
}
```

POSITIONING & LAYERS

position: static | relative | absolute | fixed | sticky;
top: 20px; right: 0; /* Works with relative/abs/fixed */
z-index: 10 | -1;
display: block | inline-block | none;
visibility: visible | hidden;
overflow: hidden | scroll | auto;
overflow-x: clip;
resize: both | horizontal;

POSITIONING & LAYERS

1. Specificity Hierarchy:
2. Inline > ID > Class/Attribute > Element
3. !important overrides all
4. Inheritance:
5. Text-related properties (font, color, line-height) inherit
6. Layout properties do NOT inherit
7. Stacking Context:
8. Created by: position: absolute/fixed, opacity < 1, transform, filter
9. BFC (Block Formatting Context):
10. Contain floats: overflow: hidden, display: flow-root
11. Performance Tips:
 - Use transform/opacity for animations
 - Avoid @import for CSS
 - Minimize layout thrashing

```
/* Reset */  
*, ::before, ::after {  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
}
```

```
/* Fluid responsive base */  
html {  
  font-size: 62.5%; /* 1rem = 10px */  
}
```

ADVANCED FEATURES

```
/* Variables */  
:root {  
  --primary: #4361ee;  
  --spacing: 1.5rem;  
}  
.element { color: var(--primary); }  
  
/* Filters */  
filter: blur(2px) brightness(120%);  
backdrop-filter: blur(10px);  
  
/* Blend Modes */  
mix-blend-mode: multiply;  
background-blend-mode: screen;  
  
/* Clip/Shape */  
clip-path: polygon(0 0, 100% 0, 50% 100%);  
shape-outside: circle(50%);
```

UI/INTERACTION

cursor: pointer | grab | not-allowed;
user-select: none | text;
scroll-behavior: smooth;
scroll-snap-type: y mandatory;
scroll-snap-align: start;
caret-color: #f00;
accent-color: #4361ee; /* Form controls */

PSEUDO-ELEMENTS & CONTENT

```
.element::before {  
  content: "★" attr(data-text);  
  display: block;  
  position: absolute;  
}  
q::after { content: " »"; }
```

Pro Tip:

Use CSS variables for theming and design tokens:

```
:root {  
  --primary: #4361ee;  
  --shadow: 0 4px 6px rgba(50, 50, 93, 0.11);  
  --transition: all 0.15s ease;  
}  
.btn {  
  background: var(--primary);  
  box-shadow: var(--shadow);  
  transition: var(--transition);  
}
```