Advanced CSS CheatSheet

Advanced CSS Cheat Sheet

Modern CSS Techniques for Professional Web Developers

Flexbox Layout

Container Properties:

```
.container {
  display: flex;
  flex-direction: row | column;
  flex-wrap: wrap | nowrap;
  justify-content: center | space-between;
  align-items: center | stretch;
  gap: 1rem;
}
```

Item Properties:

```
.item {
  flex: 1 1 auto; /* grow shrink basis */
  flex-grow: 1;
  flex-shrink: 0;
  flex-basis: 200px;
  align-self: flex-start;
  order: 2;
}
```

Common Patterns:

```
/* Center everything */
.center {
  display: flex;
  justify-content: center;
  align-items: center;
}

/* Equal columns */
.equal-cols > * {
  flex: 1;
}
```

CSS Grid Layout

Grid Container:

```
.grid {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: auto 200px;
  gap: 20px;
  grid-auto-flow: dense;
}
```

Grid Item Placement:

```
.item {
  grid-column: 1 / 3; /* start / end */
  grid-row: 1 / span 2;
  grid-area: header;
}
```

Grid Template Areas:

```
.layout {
  display: grid;
  grid-template-areas:
    "header header header"
    "sidebar main main"
    "footer footer footer";
}

.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
```

Responsive Grid:

```
.responsive {
  grid-template-columns:
    repeat(auto-fit, minmax(250px, 1fr));
}
```

CSS Custom Properties (Variables)

Declaring Variables:

```
:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --spacing: 1rem;
  --border-radius: 8px;
  --font-size: 16px;
}
```

Using Variables:

```
.button {
  background: var(--primary-color);
  padding: var(--spacing);
  border-radius: var(--border-radius);
  font-size: var(--font-size);
}
```

Fallback Values:

```
.element {
  color: var(--text-color, #333);
  /* Uses #333 if --text-color not defined */
}
```

Dark Mode Example:

```
:root {
  --bg: white;
  --text: black;
}

[data-theme="dark"] {
  --bg: #1a1a1a;
  --text: white;
}

body {
  background: var(--bg);
  color: var(--text);
}
```

Animations & Transitions

CSS Transitions:

```
.element {
  transition: all 0.3s ease-in-out;
  /* or specific properties */
  transition: background 0.3s,
              transform 0.2s;
}
```

Keyframe Animations:

```
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

.element {
  animation: fadeIn 0.5s ease-out;
}
```

Animation Properties:

```
.animated {
  animation-name: slideIn;
  animation-duration: 1s;
  animation-timing-function: ease;
  animation-delay: 0.2s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
  animation-fill-mode: forwards;
}
```

Transform Functions:

```
.transform {
  transform: translate(50px, 100px);
  transform: rotate(45deg);
  transform: scale(1.5);
  transform: skew(10deg, 20deg);
  /* Combine multiple */
  transform: translateX(50px) rotate(45deg);
}
```

Advanced Selectors

Attribute Selectors:

```
/* Exact match */
[href="https://example.com"] {}

/* Contains */
[class*="btn"] {}

/* Starts with */
[href^="https://"] {}

/* Ends with */
[href$=".pdf"] {}

/* Contains word */
[class~="active"] {}
```

Combinators:

```
/* Descendant */
div p { }

/* Direct child */
div > p { }

/* Adjacent sibling */
h1 + p { }

/* General sibling */
h1 ~ p { }
```

Structural Pseudo-classes:

```
li:first-child { }
li:last-child { }
li:nth-child(3) { }
li:nth-child(odd) { }
li:nth-child(3n+1) { }
li:only-child { }
p:empty { }
```

:not() Selector:

```
/* All buttons except primary */
button:not(.primary) { }

/* All divs without class */
div:not([class]) { }
```

Pseudo-elements & States

Pseudo-elements:

```
/* Before & After */
.element::before {
  content: "→";
  display: inline-block;
}

.element::after {
  content: "";
  display: block;
}

/* First letter/line */
p::first-letter { font-size: 2em; }
p::first-line { font-weight: bold; }

/* Selection */
::selection {
  background: yellow;
  color: black;
}
```

Interactive States:

```
a:hover { }
a:active { }
a:visited { }
input:focus { }
input:focus-within { }
input:disabled { }
input:checked { }
input:valid { }
input:invalid { }
```

Modern Pseudo-classes:

```
/* Has selector */
form:has(input:invalid) {
  border: 2px solid red;
}

/* Is selector */
:is(h1, h2, h3) {
  margin-top: 2rem;
}

/* Where selector (no specificity) */
:where(article, section) p {
  line-height: 1.6;
}
```

Modern Responsive CSS

Container Queries:

```
.container {
  container-type: inline-size;
  container-name: sidebar;
}

@container sidebar (min-width: 400px) {
  .card {
    display: flex;
  }
}
```

Clamp() Function:

```
/* Responsive font size */
h1 {
  font-size: clamp(1.5rem, 5vw, 3rem);
  /* min, preferred, max */
}

/* Responsive width */
.container {
  width: clamp(300px, 80%, 1200px);
}
```

Min/Max Functions:

```
.element {
  width: min(100%, 800px);
  height: max(300px, 50vh);
  padding: max(1rem, 2%);
}
```

Aspect Ratio:

```
.video {
  aspect-ratio: 16 / 9;
}

.square {
  aspect-ratio: 1;
}
```

Performance & Best Practices

Will-change Property:

```
/* Optimize animations */
.animated {
  will-change: transform, opacity;
}

/* Remove after animation */
.animated:hover {
  will-change: auto;
}
```

Content-visibility:

```
/* Lazy render off-screen content */
.section {
  content-visibility: auto;
  contain-intrinsic-size: 0 500px;
}
```

Logical Properties:

```
/* Use logical properties for i18n */
.element {
  margin-inline-start: 1rem;
  padding-block: 2rem;
  border-inline-end: 1px solid;
}
```

Calc() Advanced:

```
.element {
  width: calc(100% - 2rem);
  font-size: calc(1rem + 2vw);
  padding: calc(var(--spacing) * 2);
}
```

Performance Tips:

- • Avoid expensive properties: box-shadow, filter
- • Use transform instead of position
- • Minimize repaints with contain property
- • Use CSS variables for theming
- • Prefer flexbox/grid over floats

Filters & Visual Effects

CSS Filters:

```
.image {
  filter: blur(5px);
  filter: brightness(150%);
  filter: contrast(200%);
  filter: grayscale(100%);
  filter: saturate(200%);
  filter: sepia(100%);
  filter: hue-rotate(90deg);
  filter: invert(100%);

  /* Combine multiple */
  filter: blur(2px) brightness(120%);
}
```

Backdrop Filter:

```
/* Glassmorphism effect */
.glass {
  background: rgba(255,255,255,0.1);
  backdrop-filter: blur(10px);
  border: 1px solid rgba(255,255,255,0.2);
}
```

Blend Modes:

```
.overlay {
  mix-blend-mode: multiply;
  /* multiply, screen, overlay,
     darken, lighten, color-dodge,
     color-burn, difference, exclusion */
}

.background {
  background-blend-mode: overlay;
}
```

Shapes & Clip-path

Clip-path Shapes:

```
/* Circle */
.circle {
  clip-path: circle(50%);
}

/* Ellipse */
.ellipse {
  clip-path: ellipse(50% 30%);
}

/* Polygon */
.triangle {
  clip-path: polygon(50% 0%, 0% 100%, 100%
100%);
}

/* Custom polygon */
.hexagon {
  clip-path: polygon(
    50% 0%, 100% 25%, 100% 75%,
    50% 100%, 0% 75%, 0% 25%
  );
}
```

Shape Outside:

```
/* Text wrapping around shapes */
.float-image {
  float: left;
  shape-outside: circle(50%);
  clip-path: circle(50%);
}
```

Mask Image:

```
.masked {
  mask-image: linear-gradient(
    to bottom,
    black 0%,
    transparent 100%
  );
}
```