CSS Tricks Cheatsheet

A comprehensive cheat sheet covering various CSS tricks, hacks, and optimizations for improved web development. Includes code snippets and examples for easy reference.

General CSS Tricks

Box Sizing

Setting `box-sizing: border-box` makes it easier to manage element sizes by including padding and border in the element's total width and height.

```
*, *:before, *:after {
  box-sizing: border-box;
}
```

Using `border-width`, `border-style`, and `border-color` to control element borders.

```
div {
  border-width: 0;
  border-style: solid;
  border-color: #e5e7eb;
}
```

Typography

`text-rendering: optimizeLegibility;`	Improves the readability of text by enabling kerning and ligatures. **Example:** `h1, h2, h3 { text-rendering: optimizeLegibility; }`
Gradient Text	Applying a gradient to text using `-webkit-background-clip: text`. **Example:** `background: -webkit-gradient(linear, left top, left bottom, from(#eee), to(#333)); -webkit-background-clip: text; -webkit-text-fill-color: transparent;`
Text Stroke	Adding a stroke to text using `-webkit-text-stroke`. **Example:** `-webkit-text-stroke: 3px black;` See: Introducing text stroke

Layout

`display: contents;`	Makes the element vanish, but its children become direct children of the element's parent.
	Example:
	`<div class="parent"> <div class="contents"> <h1>Title</h1> <p>Content</p> </div> </div>`
	`.contents { display: contents; }`

iOS Specific Tricks

Native-like iOS Scrolling

Enabling native-like iOS scrolling using `-webkit-overflow-scrolling: touch`.

```
-webkit-overflow-scrolling: touch;
overflow-y: auto;
```

Preventing iOS scrolling on certain elements.

```
document.ontouchstart = (e) => {
  const $pane = $(e.target).closest('.scrollable>div');
  if ($pane.length === 0 || $pane[0].scrollHeight <=
$pane[0].innerHeight())
    e.preventDefault();
};
```

SCSS Mixin for iOS Scrollable Elements

```
%ios-scrollable {
  &, >div {
    -webkit-overflow-scrolling: touch;
    overflow: auto;
  }

  >div {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
  }
}
```

UIWebView Optimizations

Optimizations for UIWebView, such as disabling tap highlight color and user selection.

```
* {
  -webkit-tap-highlight-color: rgba(0,0,0,0);
  -webkit-user-select: none; /* disable text select */
  -webkit-touch-callout: none; /* disable callout, image save
panel (popup) */
  -webkit-tap-highlight-color: transparent; /* "turn off" link
highlight */
}

a:focus {
  outline: 0; /* Firefox (remove border on link click) */
}
```

See: iOS WebKit UIWebView Remove Tap/Click Highlight Border with CSS
See: Customizing the Mobile Safari Tap Highlight Color

Browser Hacks

Mozilla-only Hack

Targeting Mozilla Firefox using `@-moz-document`.

```
@-moz-document url-prefix() {
  .box {
    color: blue;
  }
}
```

Webkit-only Hack

Targeting Webkit browsers using `@media` query.

```
@media all and (-webkit-min-device-pixel-ratio: 1) {
  /* Styles for Webkit browsers */
}
```

CSS Reset and Best Practices

Universal Reset

A basic CSS reset to normalize styles across browsers.

```
html,:host {
  line-height: 1.5;
  -webkit-text-size-adjust: 100%;
  -moz-tab-size: 4;
  -o-tab-size: 4;
  tab-size: 4;
  font-family: ui-sans-serif,system-ui,sans-serif,"Apple Color Emoji","Segoe UI Emoji",Segoe UI Symbol,"Noto Color Emoji";
  font-feature-settings: normal;
  font-variation-settings: normal;
  -webkit-tap-highlight-color: transparent;
}

body {
  margin: 0;
  line-height: inherit;
}
```

Element Normalization

Normalizing various HTML elements for consistent styling.

```
hr {
  height: 0;
  color: inherit;
  border-top-width: 1px;
}

abbr:where([title]) {
  -webkit-text-decoration: underline dotted;
  text-decoration: underline dotted;
}

h1, h2, h3, h4, h5, h6 {
  font-size: inherit;
  font-weight: inherit;
}

a {
  color: inherit;
  text-decoration: inherit;
}
```