

CSS Reset and Normalize

CSS Resets: Starting with a Blank Slate

Web browsers apply their own default styles to HTML elements. These built-in styles—known as **user agent stylesheets**.

```
body {                                user agent stylesheet
  display: block;
  margin: ► 8px;
}
```

A **CSS reset** is a technique that removes all default styling applied by the browser. It effectively "resets" the appearance of elements so that developers can define **every aspect of the design from scratch**.

A reset **eliminates interference from the browser's defaults**, allowing you to establish your own consistent styles.

Example: Basic Minimal CSS Reset

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

This snippet targets all elements and removes their default margin and padding. It also applies box-sizing: border-box so that padding and borders are included in an element's total width and height. This makes layout behavior more predictable.

There are more large-scale resets available that will take out many more default settings. One example is: <https://html5doctor.com/html-5-reset-stylesheet/>

However, it's important to note that CSS resets remove **everything**, including useful default styles. That's why some developers prefer **Normalize.css**, which we'll discuss next.

Normalize.css: Smoothing Out the Differences

While CSS resets aim to remove all default browser styling, another approach is to use **Normalize.css**—a modern, community-maintained stylesheet that doesn't remove all styles, but instead **preserves useful defaults and fixes inconsistencies** between browsers.

Normalize.css is often described as a **CSS baseline**. The result is a consistent, **cross-browser-compatible foundation** that still feels familiar and accessible.

- It **fixes** browser inconsistencies rather than eliminating all styling
- Preserves **semantic defaults** that are useful (like emphasis on `` and ``)
- Helps maintain **accessibility features** in forms and controls
- Provides a stable base for modern responsive design and component styling

How to Use Normalize.css

Normalize.css can be included in your project in a few ways. One common way is to include the **CDN Link (in your HTML file)**:

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css">
```

You can also download the file and include it in your project files.

But make sure the link element referencing your css file you create for your page is placed below the normalize link element. When multiple styles apply to the same element, the one loaded **last** applies—**unless** the earlier one has higher specificity.

```
<link rel="stylesheet" href="normalize.css">
<link rel="stylesheet" href="styles.css">
```

By placing your own stylesheet **after** Normalize.css, your custom rules will override the normalized defaults as needed.

In summary, **CSS resets** give you a blank slate, while **normalize.css** gives you a level playing field. Your choice depends on whether you want complete control or smart defaults. In practice, many developers even combine both, starting with a minimal reset followed by normalizing.