Chapter 54: Normalizing Browser Styles

Every browser has a default set of CSS styles that it uses for rendering elements. These default styles may not be consistent across browsers because: the language specifications are unclear so base styles are up for interpretation, browsers may not follow specifications that are given, or browsers may not have default styles for newer HTML elements. As a result, people may want to normalize default styles across as many browsers as possible.

Section 54.1: normalize.css

Browsers have a default set of CSS styles they use for rendering elements. Some of these styles can even be customised using the browser's settings to change default font face and size definitions, for example. The styles contain the definition of which elements are supposed to be block-level or inline, among other things.

Because these default styles are given some leeway by the language specifications and because browsers may not follow the specs properly they can differ from browser to browser.

This is where normalize.css comes into play. It overrides the most common inconsistencies and fixes known bugs.

What does it do

- Preserves useful defaults, unlike many CSS resets.
- Normalizes styles for a wide range of elements.
- Corrects bugs and common browser inconsistencies.
- Improves usability with subtle modifications.
- Explains what code does using detailed comments.

So, by including normalize.css in your project your design will look more alike and consistent across different browsers.

Difference to reset.css

You may have heard of `reset.css`. What's the difference between the two?

While normalize.css provides consistency by setting different properties to unified defaults, reset.css achieves consistency by **removing** all basic styling that a browser may apply. While this might sound like a good idea at first, this actually means you have to write **all** rules yourself, which goes against having a solid standard.

Section 54.2: Approaches and Examples

CSS resets take separate approaches to browser defaults. Eric Meyer's Reset CSS has been around for a while. His approach nullifies many of the browser elements that have been known to cause problems right off the back. The following is from his version (v2.0 | 20110126) CSS Reset.

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
```

```
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
```

Eric Meyer's Reset CSS

Normalize CSS on the other and deals with many of these separately. The following is a sample from the version (v4.2.0) of the code.

```
/**
 * 1. Change the default font family in all browsers (opinionated).
 * 2. Correct the line height in all browsers.
 * 3. Prevent adjustments of font size after orientation changes in IE and iOS.
 */

/* Document
   ========================================================================== */

html {
  font-family: sans-serif; /* 1 */
  line-height: 1.15; /* 2 */
  -ms-text-size-adjust: 100%; /* 3 */
  -webkit-text-size-adjust: 100%; /* 3 */
}

/* Sections
   ========================================================================== */

/**
 * Remove the margin in all browsers (opinionated).
 */

body {
  margin: 0;
}

/**
 * Add the correct display in IE 9-.
 */

article,
aside,
footer,
header,
nav,
section {
  display: block;
}

/**
 * Correct the font size and margin on `h1` elements within `section` and
 * `article` contexts in Chrome, Firefox, and Safari.
 */
```

```
h1 {
  font-size: 2em;
  margin: 0.67em 0;
}
```

Normalize CSS