PART 2

Performance Basics

This section of the book addresses our development methodology, how to boost page-load times (client-side performance), our use of Responsive Web Design, and the web reuse pattern. Basically, we're setting context so that the rest of the book makes sense.

We cover how separation of concerns is useful to front-end developers, why we embrace progressive enhancement, and why we think you should embrace it, too. We also cover how a browser loads a web page, because it's hard to talk about client-side performance without understanding how browsers render pages.

Then we cover basic performance guidelines that apply to all pages. As already mentioned, we cover how to improve page-load times and, to a lesser extent, how to minimize strain on the network by using as little bandwidth as possible. We also address why page-load time matters. Then we discuss each of the individual guidelines, including the long pole in the tent: reducing HTTP requests. If we succeed in teaching nothing else, we hope at least to show you how to limit fetching things from the server.

Next, we address Ethan Mercotte's excellent technique: Responsive Web Design. We discuss how to use media queries, flexible images, and flexible grids to create pages that look good on almost any device. In addition to simply resizing elements and images, we add content as displays get larger.

Note that we don't advocate taking away content as displays get smaller; rather, we embrace the Mobile First paradigm and start with the smallest device we intend to support. Embracing progressive enhancement involves giving visitors with better devices a better presentation. We want you to think in terms of adding to a minimal (but still good) presentation rather than subtracting from a complete presentation.

This part explores one of our core concepts: reusing content for multiple presentations. We call each alternate presentation of the same content a treatment. Using the same HTML structure, we apply different sets of styles. Finally, we show how we rely on CSS nesting and on the fact that CSS fails silently (that is, without error messages to the visitor) to get the presentations we want.