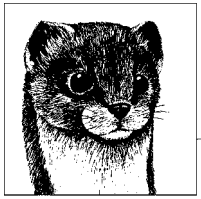

Cascading Style Sheets Fundamentals

Cascading Style Sheets (CSS) is a W3C standard for defining the presentation of web documents. *Presentation* refers to the way a document is displayed or delivered to the user, whether it's on a computer monitor, a cell phone display, or read aloud by a screen reader. This book focuses primarily on the visual aspects of presentation, such as typography, colors, alignment, layout, and so on. CSS is the mechanism for providing these types of style instructions to elements in a document that has been marked up with XHTML, HTML, or any XML language. Most important, CSS keeps these presentation instructions separate from the content and its structural and semantic markup.

Before CSS, web designers were at the mercy of the browser's rendering engine and internal style sheets for the way HTML elements looked in the browser window. Presentational elements and attributes added to HTML, such as the `font` tag and the `bgcolor` attribute, granted some additional control over visual display, but the integrity of markup suffered. Cascading Style Sheets (or just "style sheets" in these chapters) hand visual display decisions back to designers and authors. This comes as good news both for designers who want more control over presentation and for those who are eager to see HTML get back to the exclusive business of defining document structure and meaning. Style sheets make both of these goals possible.

CSS in a Nutshell

The chapters in this section provide a solid overview and reference of CSS and its properties. This book focuses on CSS used with documents written in (X)HTML, although CSS can also be used with any XML language.

This chapter lays an important foundation for understanding how CSS works, including rule syntax and how style sheets are applied to documents. It also covers some critical key concepts at the core of CSS, such as inheritance, handling conflicting styles (the cascade), how elements display, and the box model.

Browser issues are briefly addressed as well. The chapter finishes with a section on specifying values in CSS.

Chapter 17 explains all the various ways elements can be targeted for style application, and Chapters 18 through 23 cover the CSS visual display properties as they are specified in the CSS 2.1 Recommendation. These chapters document how CSS is designed to work. Browser support varies, of course, so this book provides notes if a property or its values are particularly problematic in a browser.

Finally, Chapters 24 and 25 put everything together in real-world applications. Chapter 24 is a cookbook of some of the most popular CSS techniques, such as CSS rollovers and multicolumn layouts. All of the browser-related problems and solutions are aggregated in Chapter 25, making it a handy reference if you encounter problems down the road.

In the interest of keeping everything "in a nutshell," the chapters in this section stick to visual media properties. The CSS properties related to interface, paged media, and aural (speech) media are included in Appendix B.

The Benefits of CSS

The benefits of using web standards for web page production were covered in detail in Chapter 1, however, it won't hurt to start off with a refresher of the advantages style sheets offer.

Greater typography and page layout controls
: With style sheets, you can specify traditional typography features that you could never do with HTML alone (even with its presentational extensions).

Less work
: Not only can you format all similar elements in a document with a single style rule, external style sheets make it possible to edit the appearance of an entire site at once with a single style sheet edit.

Potentially smaller documents
: Redundant `font` tags and nested tables make for bloated documents. Stripping presentational HTML out of the document saves on file size.

Potentially more accessible documents
: Well-structured and semantically rich documents are accessible to a wider variety of devices and the people who use them. Techniques based on presentational (X)HTML, such as using the `font` element to format headings and breaking up content into complex nested tables, damage the integrity of the source document.

Presentational HTML is on its way out
: The W3C has deprecated all presentational elements and attributes in the HTML and XHTML specifications. One day, browsers will not be required to support them.

It's well supported
: As of this writing, nearly every browser in current use supports nearly all of the CSS 1 specification. Most also support the majority of the Level 2 and 2.1 Recommendations.

As for the disadvantages...there aren't any, really. Some people complain that style sheets can be misused, but you can't fault CSS for that. There are some lingering hassles from inconsistent browser support that require workarounds and extra planning (see Chapter 25), but that is by no means an argument against using style sheets for presentation right away.

How CSS Works

What follows is a simplified explanation of how style sheets work. At its heart, the process actually is this simple.

1. Start with an XHTML (or HTML) document. Ideally, this document will have been given a logical structure and semantic meaning using the appropriate XHTML elements. The XHTML markup is commonly referred to as the *structural layer* of the web page. It forms the foundation upon which the *presentation layer* is applied.
2. Write style rules for how each element should ideally look. Each rule targets the element by name, and then lists properties—such as font, color, and so on—to be applied to the element. The specifics of writing style rules are covered in the upcoming "Rule Syntax" section.
3. Attach the styles to the document. The style rules may be gathered up into a separate document and applied to a whole site, or they may appear in the header and apply only to that document. Style instructions may appear within an XHTML element itself as well. Each of these methods for attaching style rules to the content document is discussed in the "Adding Styles to a Document" section in this chapter.

Needless to say, there's a bit more to each step than is described here. The next section begins to get into the nitty gritty of style sheets by looking at the parts of a style rule.

Rule Syntax

Style sheets consist of one or more rules for describing how a page element should be displayed. The following example contains two rules. The first rule makes all the `h1`s in a document gray; the second specifies that paragraphs should be set in 12-pixel high Verdana or some sans-serif font:

```
h1 {color: #eee;}
p {font-size: 12px;
   font-family: Verdana, sans-serif; }
```

Figure 16-1 shows the components of a style sheet rule.

The two main sections of a style sheet rule are the *selector* (which identifies the element to be styled) and the *declaration* (the style or display instructions to be