

# **Cascading Style Sheets**

---

Basics Benefits & Information

**Pierre Berryer**

**Original Date: December 2000  
Version 1.01**

## **Cascading Style Sheets (CSS)**

This specification defines Cascading Style Sheets, level 2 (CSS2). CSS2 is a style sheet language that allows authors and users to attach style (e.g., fonts, spacing, and aural cues) to structured documents (e.g., HTML documents and XML applications). By separating the presentation style of documents from the content of documents, CSS2 simplifies Web authoring and site maintenance.

CSS2 builds on CSS1 (see [CSS1]) and, with very few exceptions, all valid CSS1 style sheets are valid CSS2 style sheets. CSS2 supports media-specific style sheets so that authors may tailor the presentation of their documents to visual browsers, aural devices, printers, braille devices, handheld devices, etc. This specification also supports content positioning, downloadable fonts, table layout, features for internationalization, automatic counters and numbering, and some properties related to user interface.

*Excerpt from: Cascading Style Sheets, level 2 CSS2 Specification Guide*

### **Benefits**

CSS centralizes the storage of formatting information for a web site. The formatting codes (styles) are stored in one file that is linked to each of the web site's pages. The styles are defined and applied to a tag (i.e. a header tag such as <H1>), a Class, or an ID name. To use the formatting information, elements merely need to use the tag, apply a class to a tag or name an element's ID. When implemented well, it becomes easy to alter the entire site's look from within one document. Because the formatting is independent of the web pages, be they HTML, JHTML, XML, or DHTML, it can be controlled and modified by the designer while the content is still being worked on.

Implementation of CSS is facilitated through the use of graphical web editors such as Macromedia Dreamweaver. This software provides excellent support for the editing and display of CSS. The editor can also be configured to format HTML code as desired to maintain human reading and editing ability. Yet, once the user is aware of its use, CSS coding remains easy to modify by hand. Because CSS is straightforward to understand and implement, it can coexist with static pages as well as dynamically created pages. Also, through the creative use of templates and library objects from within Dreamweaver, a web site's pages can be broken up into individual pieces which can be integrated and used to update the entire site in one pass.

### **Attributes**

CSS allows the layers and positioning information to be stored in a style sheet, rather than inline in the HTML. This allows absolute positioning, changeable afterwards via one document. This also allows for the HTML code to be cleaner because the detailed positioning information for sections, headers, images, text, etc. can be entered externally and kept free of accidental positioning changes in the editing of individual pages.

Style sheets can affect the font and position of text in ways previously impossible in straight HTML. From a designer's perspective, this enables much more precise control over the "look" of a site during all stages of development. The positional layout control enables interface designers to alter the placement of repeated elements such as content areas, headers, and buttons without needing to modify pages individually.

The CSS instruction set can be used in part or completely, with varying degrees of success dependant on the browser being used. Non CSS browsers still show pages correctly if they've been designed to degrade gracefully. This can be a big benefit for sites adhering to the Americans with Disabilities Act (ADA) standard for web site design. When the browser does not support CSS, the pages can still display as traditionally appearing HTML pages. If properly designed, they can degrade to work in older browsers very well and maintain the logical machine readability that many handicapped individuals require. If the site uses styles to look great when seen, yet is still able to be browsed and accurately represented by a non visual web device, the design succeeds. By applying styles to standard tags, a designer can show section headers (<H1> etc.) in any font variation, yet, because they are web standard headers, a text to speech web browser would still understand their role in the organization of the page's contents.

CSS can also integrate with HTML or XML and Javascript to form DHTML, or Dynamic HTML. Because the CSS attributes remain modifiable even after the page loads, its design can be dynamically modified in realtime. Elements such as the look of the fonts, color of the page, or location of the layers can all be modified in response to user interaction. This allows simple usability effects such as text links that change their appearance when you roll over them, yet are still in accordance with the ADA. Layers can also be animated and triggered via user input. Macromedia Dreamweaver facilitates the implementation of the underlying code.

An interesting aspect of CSS to keep in mind is that text can be forced to appear at a certain point size, regardless of the browser's font size settings. This can be ideal to ensure that the site is seen as intended during presentations or technology demos and that text fits as planned within layers. The unfortunate side effect of this is that the user is limited to viewing the site at a font size that may be too small. Issues like this should be kept in mind when designing sites that look good, but maintain effective usability. Specifying text in variable units like x-small, etc. let's the user adjust the text size.

### **Cons**

Like any newer Internet technology, CSS is not without its problems. The primary concern is that it is not yet fully supported in Internet Explorer 5 (IE) or Netscape Communicator 4.7 (NC). While they both tend to display text formatted relatively accurately, NC does not adequately support layers. It exhibits one bug in particular, in which, after the page has loaded, the layers narrow to the width of the smallest word, regardless of the size specified in the HTML document or the style sheet. The current workaround to this is to place a single cell table with the correct size inside the layer and to nest the layer's contents in that table or a transparent single pixel image sized to the width you want. This forces Netscape to widen the layer appropriately.

CSS also tends to display differently in different browsers, whether using different software programs or different operating systems. And as CSS cannot currently be disabled in all browsers, the users are forced to view it in one way, which may not be the way the designer's browser shows the site.

When text size is specified in units, it cannot be changed from within the browser. The solution to this is to use variable text sizes such as small, x-small, large, etc. These will resize depending on the base font size of your browser.

### **Links**

Webmonkey | StyleSheets: <http://hotwired.lycos.com/webmonkey/authoring/stylesheets/>

Webmonkey | [Mulder's Stylesheets Tutorial](#)

<http://hotwired.lycos.com/webmonkey/authoring/stylesheets/tutorials/tutorial1.html>

Stylesheets let you control layout like never before, right down to the spacing between letters.

Mulder shows you how it's all done. *16 Sep 1999*

Cascading Style Sheets, level 2, W3C Official Guide

CSS2 Specification: <http://www.w3.org/TR/REC-CSS2/>

Cascading Style Sheets, level 1 (CSS1) [Recommendation - 17 December 1996, revised 11 January 1999]

<http://www.w3.org/TR/REC-CSS1>