*Essential Information for Developers*

# *What's New in*

# CSS3

O'REILLY®

*Estelle Weyl*

O'REILLY®

# Velocity

## Web Performance and Operations

## EUROPE

2–4 October, 2012 | London, England

Experience three days immersed in the web ops and performance solutions you need to master automation strategies, hone your mobile and web optimization skills, manage big data, excel in metrics and monitoring, develop failover and outage responses, and more.

Gather with hundreds of web ops and performance professionals in London this October to share ideas and to discover the technical skills, tools, and best practices to build a faster, stronger web.

Learn more and **save 20%** with **code CSS3**
**http://oreil.ly/VEU12**



O'REILLY®
Spreading the knowledge of innovators.

# What's New in CSS3

*Estelle Weyl*

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

**What's New in CSS3**

by Estelle Weyl



Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

**Cover Designer:** Karen Montgomery **Interior Designer:** David Futato

September 2012: First Edition

**Revision History for the First Edition:**

2012-09-01 First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449344931* for release details.



ISBN: 978-1-449-34493-1

# Table of Contents

# CHAPTER 1
# Introduction

Cascading Style Sheets (CSS), the presentation layer of websites, are growing ever more important and more sophisticated. What was once a toolkit for specifying consistent fonts and layouts has grown into a much larger design toolbox that even ventures into animation and transitions. CSS Level 3 (CSS3), the latest generation of CSS, crosses the boundary from presentation into user interface while building on earlier capabilities.

CSS3 isn't finished yet, but you can start using it in browsers today. While you can't use the features of CSS3 that have yet to be defined, you certainly can and should start using features of CSS3 that are stable and well-supported in modern browsers. Some portions of CSS3 are already published as recommendations; other parts are well on their way. Some properties are deployed, but the specification process lags behind. Other properties require vendor prefixes because the syntax in the specifications is still in flux. New modules are just being suggested, and are experimentally supported in a single browser. Even before work on CSS3 is complete, work has already begun on CSS4.

# CHAPTER 2
# Why CSS?

Users should be able to access your content no matter which device they use or what software is on those devices. CSS lets developers define how content is viewed, including methods for tailoring or customizing the presentation of the content based on the device. For example, your users may access your content via a browser on a netbook, a browser on a phone, on their TV, with a screen reader, as a presentation, or even printed as a PDF. CSS provides mechanisms for controlling the appearance or presentation of your content no matter the device.

CSS is the presentation layer of the Web. With CSS, web developers lay out web pages, make web pages match web designers' intentions, and make pages look uniform as users navigate from page to page within a website.

CSS is a stylesheet language. CSS is a markup language, not a programming language.

CSS is used to define the style information for the various elements of the document that it styles. CSS is used to define the look and formatting, or presentation semantics, of a document written in a markup language such as HTML, XHTML, XML, or SVG.

The benefits of CSS include the maintainability, accessibility, and consistency they provide you and your users. In addition, well-coded CSS is fast and portable.

By linking all content of a website or property to a single stylesheet, when a change is made to the stylesheet, such as the color of the body's

background or the color of the font, these changes impact the entire site. This makes maintaining a site easier and faster, since only one location must be updated. It also ensures consistent look and appearance across your pages. If you were using the archaic HTML elements to define appearance (or even embedding styles with the style attribute or element), making an edit to the look and feel of your site would require updating every file site-wide.

Sites using stylesheets for the presentation layer are also generally more accessible. While coding the underlying HTML is the most important component in accessibility, stylesheets can aid in making sites more accessible. Browsers ignore CSS that they do not understand, ensuring that user agents are still able to render the content even if they cannot render the styles.

(CSS also lets developers create new problems. CSS will not stop you from placing white text on white background, which is impossible to read. With great power comes great responsibility!)

Stylesheets also improve the download speed. Yes, the first time a page is loaded the page may be slower to render as all the CSS is downloaded. But, future attempts to access the site stylesheet can pull from the cache instead of the server, reducing download time and the number of HTTP requests.

When you employ stylesheets, you can define which stylesheet is used depending on the media type, view port size, and other factors. A site can use the same content with different stylesheets for mobile, television, and print. By just applying a well-coded print stylesheet, with paged media, you can print a document, complete with header and footer and without images based on its print stylesheet, from the same HTML document that might win "best design" based on its gradients and rounded corners when the screen stylesheet is attached.

CHAPTER 3

# CSS3 Specifications: A Modular Approach

CSS2.1 took 13 years to become a Recommendation, and it wasn't even that large of a specification. Browsers implemented it in pieces over that period, but it wasn't until the release of Internet Explorer 8 (IE8) in March of 2009 that we saw a browser with virtually full support of CSS2.1. However, even though IE8 had better support for CSS2.1 than any other browser, Firefox, Opera, Chrome, and Safari are already supporting CSS3 features, whereas IE8 supports virtually no CSS3 features.

Instead of producing one humongous specification like CSS2, CSS Level 3 is divided into many separate specifications or documents called "Modules". Each module adds new capabilities or extends features defined in CSS2. Because the specifications are modularized, different modules are at different levels of stability and implementation.

So far, the CSS Working Group at the W3C has initiated work on more than 40 CSS modules. Some modules, such as Selectors, Namespaces, Color, and Media Queries, are considered stable and are either in Candidate Recommendation or Proposed Recommendation status. The first module to become a W3C Recommendation was the CSS3 Color Module, published the same day that the CSS2.1 specification became a Recommendation. Work on different modules has progressed at different speeds. Blockages in one module, fortunately, do not hold up any other modules.

While more than 40 modules have been initiated, not all of them are actively being worked on or even supported in any browser. Of the 40-plus specifications initiated, three-quarters seem to have a chance of becoming Recommendations.

The CSS Working Group defines the specifications by status and priority. As new features are proposed, some are making it into existing CSS3 modules, some new modules are being added to the specifications, and some proposals are already being added under the CSS Level 4 umbrella. Of the more than 28 or so modules currently progressing in the CSS3 specifications, each module deserves its own chapter (or two or three). Unfortunately, we can't cover all the details about all of them, but we can provide a quick overview of the focus of each of them.

## Modules in CSS3

CSS3 is being developed by a single working group: the CSS Working Group. This group is working in teams on the various modules at various speeds, with different modules having different priority levels.

Each module will pass through five steps as it works its way from (1) Working Draft to (2) Last Call Working Draft to (3) Candidate Recommendation or Call for Implementation to (4) Proposed Recommendation to (5) W3C Recommendation (REC).

While a spec is being updated, the most recent version may seem to be the Working Draft, when, in actuality, the Working Draft may be grossly out of date. While under active development, a module's Editor's Draft may have many more up-to-date and supported features than the published Working Draft. For example, the Animation Editor's draft is up to date and includes properties and lists properties as animatable that were not in the Working Draft from 2009 to 2012. All browsers that supported CSS animation supported the specification listed in the Editor's Draft, even though several newer features, like the animation-fill-mode property, were not in the Working Draft until it was finally updated in April 2012 after three years of seeming silence.

At this time, there are several CSS3 modules that are well supported across all modern browsers, with some properties and values being supported with vendor prefixes. Vender prefixes allow browser man-

ufacturers to include CSS features that are not fully specified. By including a vendor prefix, if the spec changes, browser support for a prefixed non-conforming syntax will not break in future browser releases.

# CSS3 W3C Recommendations

As of this writing, only four modules are complete as Recommendations: Colors, Selectors, Namespaces, and Media Queries. A fifth, the Basic User Interface module, has been a Candidate Recommendation for a long time.

## Color Module

Originally, we had web-safe colors. The CSS Color module expands the number of named colors, adds alpha transparent colors, includes colors based on hue, saturation, and lightness (instead of just red, green, and blue), and even adds color names based on UI features. The color module specifies color-related aspects of CSS, including transparency and the various notations for the `<color>` value type.

## Selector Module

Selectors in CSS allow developers to target elements and pseudo elements based on their position in relation to other elements. Prior versions of CSS allowed selection based on IDs, classes, attributes, and descendant relations. CSS3 allows for more finely tuned attribute matching, structural selections, and even selecting elements based on features they don't have.

All modern browsers support all of the CSS3 Selectors. You can now match elements based on attributes and attribute values, including partial matches, and current state such as checked checkboxes and valid and required form elements.

Work has already begun on the CSS 4 Selector Module. When supported, Selectors will provide new ways to select elements, including selecting ancestor elements based on properties of their descendant nodes.

## Namespaces

Even though it is one of the least-known of all of the CSS3 modules, the Namespaces Module is also a Recommendation. With CSS3, you can use "namespaces," often the value of the xmlns attribute, even if omitted, to distinguish multiple uses of the same element name from each other. With namespacing, CSS selectors can be extended to select elements based on their XML namespace as well as their local name.

## Media Queries

Media Queries is an enhancement of the @media rules of previous versions of CSS and the "media" attribute in HTML. Before, we just had media type, like screen and print. The Media Queries Module adds parameters such as size of display, color depth, and aspect ratio. We all know that TVs, handheld devices, smartphones, and even monitors come in all sizes. The Media Queries Module is the basis for the responsive web design movement.

While the module only became a recommendation in June 2012, some browsers are supporting features beyond the scope of the spec. Not only are browsers supporting media queries based on browser size as defined in the spec, but also on browser capabilities. In addition, work has begun on a Conditional Rules Module, adding @supports and @document to the @rule for targeting styles based on feature support and document URL.

The Basic User Interface Module became a Candidate Recommendation in 2004, but still hasn't achieved recommendation status, and is being re-revised. The module contains several useful features for styling interactive, dynamic aspects of web pages, like outlining styles. Some of these features that were added so long ago finally became relevant with browser support of HTML5 features like web form validation. Finally, we can style form elements based on their current status, like validity, focus, and whether or not the form element is required or checked with dynamic pseudo classes that update as the user interacts with the interface.

## (Re)Defining CSS

Some of the modules are simply extensions of CSS2 features. Some modules are pulled from the CSS2 specification with no added features, others have limited added properties and values, and some clarify missing functionality from the original specification.

The modules that define how CSS is written haven't changed much, as CSS must always be backward (and forward) compatible. The Style Attributes Module, written by the CSS Working Group, defines the syntax of CSS rules in HTML's "style" attribute.

The CSS Syntax Module defines the grammar that all levels of CSS adhere to. Since everything in CSS must be backward and forward compatible, this module hasn't been updated since its original publication in 2003. The CSS Tables Module, describing the layout of tables, rows, columns, cells, and captions, with their borders and alignments, is inactive and was never actually published.

Unlike syntax, tables, and style attributes, the CSS Text Module has been updated. It includes all the text-related properties of CSS2 (justification, text-wrap, etc.) and adds several new properties. The module defines new properties for handling line breaks, hyphenation, and issues related to text in different languages and scripts (kashida text justification, for example). The module was a Candidate Recommendation in 2003, but has been back in Working Draft status (and updated) since 2005. The original text module was split into three parts: Text, Writing Modes, and Line Grid, with the latter two being most relevant in internationalization.

The CSS Values and Units Module describes the common values and units that CSS properties accept. It also describes how "specified values," which is what a stylesheet contains, are processed into "computed values" and "actual values."

Most people think that @font-face is a feature new to CSS3, but it was actually first introduced in CSS2. It was removed in CSS2.1 since no browser supported the spec as written. The CSS Fonts Level 3 Module includes all the font features from CSS2 with added features. The properties that are new in CSS3 include text enhancement: there are properties to emboss and outline text, and even to kern and smooth text.

The CSS Paged Media Module builds on the CSS2 Box Model Module, defining page model and paged media. It provides properties and rules for pagination, page margins, page size and orientation, headers and footers, widows and orphans, image orientation, and extends generated content to enable headers, footers, and page numbering. The CSS Generated Content for Paged Media Module provides for printing properties that extend beyond the Paged Media Module, providing properties for creating footnotes, internal and external page cross referencing, and constructing running headers from section titles. The CSS Generated and Replaced Content Module replicated CSS2 in defining how to put content before, after, or in place of an element, but, like the CSS Hyperlinks Presentation and Line Layout Modules, is not being worked on and is, in fact, currently obsolete.

The working draft for the Paged Media Module currently dates from 2006, but the Editor's Draft is active, updated, and getting browser support.

# New Modules

While some modules are already recommendations, other modules were added as recently as 2011 and 2012. The CSS Conditional Rules Module, CSS Device Adaptation Module, Regions Module, Grid Layout Module, and Object Model Module are relatively new. Some of these modules were based on browser implementations of non-specified features, so they are already supported, in some cases with vendor prefixes, in the browsers that initiated the features.

## Conditional Rules

The Conditional Rules Module adds @supports and @document to the @rule for targeting styles based on feature support and document URL. We've had @media for a while, and the device adaptation module (see below) adds comparison features like height, width, and orientation. But, as we've seen with Modernizr, and other libraries that feature detect, sometimes you want to know what your browser is capable of —not just what size it is—when styling it. This module covers that angle.

## Device Adaptation Module

The CSS Device Adaptation Module came into being because of the proliferation of mobile devices and other devices that don't adhere to standard monitor sizes: browsers now come in all shapes and sizes. Well, at least all sizes (they're still pretty much all rectangular).

The Device Adaptation Module defines how a containing block relates to the viewport and how CSS units relate to real units. When developing for smartphones, many developers want their pages to be exactly the size of the window. They disable scrolling and zooming, generally with the `<meta name="viewport">` tag. The Device Adaptation Module defines @viewport rules that allow the initial containing block to be a different size than the viewport. It basically provides CSS support for what has been hijacked by meta tags. This module also provides for zooming, and mapping between CSS units and real units. These are all features that have been set by `<meta>` since 2007, but really should be included as part of the presentation layer, rather than in the head of the document as content.

## Object Model

The Object Model Module isn't CSS. It's JavaScript. It is basically a definition of how CSS is translated into JavaScript properties: how CSS can be targeted or read as part of the DOM. The CSS Object Model Module, CSSOM View Model or CSS-OM, is an API for manipulating and/or reading CSS from JavaScript. It provides methods of inspecting and manipulating the view information of a document. It provides ways of getting and setting an element's position, getting the width of a viewport, and more.

CHAPTER 4

# High-Priority CSS3 Modules

There are a few modules that aren't yet at Recommendation status, but they are listed as high priority for the CSS Working Group. The CSS Working Group is fairly small, so these high-priority modules get a disproportionate amount of attention.

## Image Values

The Image Values Module, or CSS Image Values and Replaced Content Module Level 3, defines how properties can refer to images, and how to make images on the fly with **CSS Gradients**.

Background-image and list-style-image properties were included in CSS1, but were limited to a single imported image.

Images in CSS are not limited to GIFs, JPEGs, and PNGs. You can include these types of images, gradients, or even SVG as images in CSS. The CSS Image Values and Replaced Content Module defines the syntax for image values in CSS. We can now include single or multiple URIs denoting references to image files or gradients. In addition, with CSS3, we can control the size and fully control the placement of our background images.

Between gradients and SVG, you can create almost any desired effect. CSS also allows for the inclusion of images as data URIs. Between SVG, data URI, and gradient support, it is possible to include all the design elements for a site without adding any HTTP requests for external content.

## Backgrounds and Borders

Backgrounds and Borders is one of the most popular modules. It expands upon the background colors and images and the style of borders from previous CSS. New functionality includes the ability to resize background images, place them based on any corner (not just top/left), allows for images as borders, provides for native rounded corners without adding images, and defines box shadows. Most browsers support the new features, with border image and four-point background image positioning still waiting for full browser support.

CHAPTER 5

# CSS3 Transitions, Transforms, and Animations

CSS3 transition, transforms, and animations are examples of where Editor Drafts and browsers sometime make much more progress than indicated by the Working Drafts.

Transforms have been supported in WebKit for a long time and are now supported in the latest versions of all browsers, including mobile browsers. No longer an experimental feature, some browser vendors have dropped the prefix for the transform property.

The 2D Transformations Module defines a property that applies rotations, translations, and other transformations to a box, similar to the transformations provided in the SVG specification. The 3D Transformations Module extends the 2D transformations with a perspective transformation. These two transforms modules have been merged into the CSS Transforms Module.

The Transitions Module enables the animation of properties between two states (before and after) and defines how the appearance of an element should change as it moves from state to state, such as transitioning between pseudo classes. Supported in all modern browsers with vendor prefixes (and it will be supported in IE10 without prefixing), you can define properties that should transition over a given duration and even after a given delay if you so choose, rather than instantly, from one state to another.

The Animations Module specifies which properties change their values during an animation, what values they take successively, and how much time it should take to fully complete an iteration of the animation. The Animations Module does not define what causes a particular animation to start, only what happens during an animation.

The difference between Transitions and Animations is that, with Transitions, the cause of the change is defined by a state change, whereas the Animation initiation can be defined for any reason. Also, Animation allows for finely tuned control of changes over time, whereas Transitions only allow animation with two keyframes, a beginning and an end.

# CHAPTER 6
# Layout Modules

The Multi-Column Layout Module includes new properties to flow content into developer-defined flexible columns. The Multi-Column Layout Candidate Recommendation provides properties to allow the display of content in multiple columns with definitions like the column-gap, column-count, column-width, and column rule. While some browsers support the above features, the support for column breaking, spanning, and filling isn't quite there yet.

The Template Module, Advance Layout Module, or Grid Template Layout Module describes a new way to position elements using constraints on their alignment to each other and on their flexibility. The layout is described by a template, which resembles a traditional layout grid, with rows and columns as in a table. The developer defines the display of the grid, or template, and then dictates which position within the display each sub-container element should take. It can be applied to a page or to individual elements, for example, to lay out a form. This module will be merged with the Grid Layout Module.

The Grid Layout Module provides for creating a flexible design grid for an element so that the descendants of the element can be positioned relative to that grid. Descendant elements can be aligned to each other in two dimensions. Areas of the grid can be assigned names both for ease of use and to create a level of indirection that facilitates reordering of elements.

The Grid Layout Module overlaps the Multi-Column Layout, Template Layout, Flexible Box Layout, Grid Positioning, and Regions Modules, but doesn't replace them.

Vertical alignment has always been difficult without the use of tables. The Flexible Box Layout Module solves this issue. The Flexible Box Layout Module defines new values for the display property, which cause an element to be displayed as either a column or a row of child elements. Additional properties determine the order of the child boxes (left to right, bottom to top, etc.) and how space is distributed over the children and the spaces between them. The module is primarily intended for forcing rows of controls in a GUI to equal height or width. It is currently being "finalized" and property names and values are seemingly no longer in flux. Most browsers support prefixed versions of the specification, with some browsers dropping the prefix in what appears to be the final version of the syntax.

The Grid Positioning Module provided for laying out elements in a grid. An element with columns, like those created with the Multi-Column Module defined above, creates an implicit grid. This module was made obsolete in favor of the Grid Layout Template Module described above.

One of the newer proposed modules is the Regions Module, which allows a "box" of content to not look like a "box." To date, everything in a document is in a rectangular box. The Regions Module will allow chunks of content to appear non-rectangular. The module defines two complementary methods: several boxes can form a chain where each is filled with the text that overflows from the previous one; and a box can define a shape inside its rectangle that constrains all text to the inside of the shape. This module enables developers to create layouts that more closely resemble print magazine layouts.

The Exclusions and Shapes Module has yet to be implemented. When fully defined and supported, it will enable wrapping inline content around an exclusion shape or to flow inline content within a shape. It has a similar goal to the Regions Module of enabling more magazine-like layout.

The newest module is the CSS Box Alignment Module, which defines properties for alignment for all the layout modules described above. It handles the alignment of boxes within their containers.

The features defined in some of these module will likely be migrated out of their own modules and merged into fewer grid or template modules. Attempts are being made to reduce the number of separate specifications. If this works out, the seven modules—Grid Layout,

Multi-column Layout, Template Layout, Flexible Box Layout, Grid Positioning, Regions and Exclusions, and Shapes—may eventually be condensed to just three: Multi-Column, Flexible Box, and a third one for grids/templates/regions.

# CHAPTER 7
# New and Shiny

There are two draft specifications that have yet to be published, but are supported with prefixes in at least one nightly browser build. The Filter Effects Module (currently here, permanent home here) enables filters such as blur, shadow, color modification, etc., to be applied to an element after rendering but before being composited. Filter support began in Chrome 18 (with vendor prefixing), but is a nascent feature, using lots of RAM and battery power.

Compositing and Blending (currently here, permanent home here) is another new module. It also allows for Photoshop-like effects, such as color difference, color mask, color shift, etc.

CHAPTER 8

# Internationalization and Accessibility

Contrary to what sometimes seems to be popular belief, the entire Internet audience is not made up of sighted, young, English-speaking individuals. There are different fonts, character sets, and even text directions. Some people read the Web in right-to-left languages, and others even read the Web with screen readers or their fingers. CSS has modules to help handle some of the different ways people consume the Web.

## Ruby

Not to be confused with the Ruby programming language, the CSS3 Ruby Module provides the ability to add annotations on top of or next to words, most often used in Chinese and Japanese to give the pronunciation or meaning of ideograms. Ruby describes CSS properties to manipulate the position of "ruby," which are small annotations on top of or next to words. They are often used to give the pronunciation or meaning of difficult ideograms.

## Lists

The CSS3 Lists and Counters Module, formerly the Lists Module, contains properties for styling lists and counter styles. CSS1 already had lists and counters. This module adds various types of bullets and

numbering systems. Lists now supports a whole slew of new list-style types, including ordering lists in the character sets of all alphabets. The lists module also adds a marker pseudo element just in case the plethora of available bullets don't meet your needs.

## Writing Modes

Not all text is written from left to right, top to bottom. The CSS3 Writing Modes Module provides for horizontal lines of text that are stacked from top to bottom, like CSS, bottom to top (like pavement painted signs), vertical text stacked from right to left (Japanese) and left to right (Mongolian). The module also describes bi-directionality, when the order of letters within a line and the rotation of letters that might occur inside vertical text.

The Line Grid Module, formerly part of the Writing Modes Module, has not yet been published, but it should describe text where characters are aligned to an invisible grid, such as found in Japanese writing.

## Speech

Some visually impaired users use screen readers to have documents read aloud to them. The CSS Speech Module provides for properties that allow developers to define how screen readers aurally render documents. CSS3 provides for properties that enable controlling of volume, balance, pausing, pitch, cues, pauses, and more. The speech media type has also been added to the CSS3 specifications.

Screen readers actually pre-date the Web. Screen readers are used to read all running programs, not just browser content, to users. So speech is not new. In fact, CSS2 contained an Aural Module, but that model was never completely implemented.

The original CSS2 Audio Module was to become the Aural Style Sheets Module. It contains properties for attaching background sounds to elements and sound effects to state transitions, such as link activation or "hovering" over an element. It was anticipated that it would be used for overlaying multiple sounds, positioning a sound left or right in stereo space, and playing a sound in a loop, but was never carried over in to the Aural Module.

The speech properties in CSS3 are similar to those in CSS2, but they have different values. If you are supporting the old, CSS2 values, use the "aural" media type. If you are using the new values, use the "speech" (or all) media type.

CHAPTER 9

# Ignored and Abandoned Specifications

The CSS Working Group revisited most of the components that made up CSS2 and divided them into many modules. Some of these modules have had drastic changes and/or are actively being updated.

There are some modules that were initiated but abandoned. Some haven't been edited in more than five years, and some have been explicitly made obsolete.

The CSS Positioning Module aimed to define "absolute," "fixed," and "relative" positioning, which is supported everywhere, brought nothing new to the table, and it hasn't been updated since 2003. The CSS Presentation Levels, which introduced a presentation-level property useful for slide show presentations, was also made obsolete. The CSS Tables Module is inactive and has never been published.

The Extended Box Model was also made obsolete, with internationalization features being merged into the Basic Box Model Module, and marquee being spun off into its own module. The Basic Box Model Module describes the layout of block-level content in normal flow with very few new features. It hasn't been updated since mid-2007, and its "newer" features are either not supported, or, if supported, defined in other modules.

The CSS Introduction was also dropped. It has been replaced with a semi-annual CSS Snapshot. These snapshots provide summaries of the status of each of the modules.

## Marquee

Now associated with the mobile profile, the Marquee Module contains the properties that control the speed and direction of the "marquee" effect. No, we no longer need to include the deprecated `<marquee>` tag. Instead, we can scroll overflowed text over time with just CSS. Fortunately, this relic of the '90s is not well supported. Marquee is a scrolling mechanism that needs no user intervention: overflowing content moves into and out of view by itself. Marquee is mostly used on mobile phones. This module was added in 2008 when Marquee was pulled out of the non-updated Box Model Module, and hasn't been updated since. It has been a Candidate Recommendation for more than three years.

## CSS Media Profiles

There are a few, mostly non-updated, media profile modules. The CSS Mobile Profile describes a subset of CSS that is suitable for handheld devices, such as mobile phones. The CSS Print Profile describes a subset of CSS that is suitable for documents printed on low-cost printers. The CSS TV Profile describes a subset of CSS that is suitable for documents displayed on TV sets, including text documents that are broadcast over digital TV. The CSS Reader Media Type Module was also dropped in March 2008, but the "reader" media type keyword it introduces was added to Media Queries.

Some modules have to do with markup languages other than HTML. For example, the Behavioral Extensions to CSS defines the "binding" property for XBL. It too has been abandoned. There were plans for a CSS Math Module to style mathematical formulas, but there is no working draft as of yet.

# CHAPTER 10
# SVG

Scalable Vector Graphics, or SVG, is a completely different W3C specification, not in any CSS Working Group specification. The properties for styling SVG (or similar graphics languages) are defined in the SVG spec, not in any CSS module, even though some of the properties are similar. So why mention SVG here? SVG can be used together with other properties in a stylesheet. SVG images can be included anywhere a background image can be included, either linked as an external file or as a data URI.

# CHAPTER 11
# What We've Learned

This has been a brief overview of features that are in the specifications, from newly written modules to specifications at the final recommendation level. New modules are being added, new features are being considered, and work has begun on some CSS Level 4 modules. In addition, we may soon see flexible images with file source based on viewport size, selectors based on grid layout and time, and box alignment properties.

Browsers are continuously integrating new and experimental features, sometimes before a specification exists to define those features. Fortunately, we don't have to wait for specifications to reach final recommendation status before being able to play with them.

## About the Author

Estelle Weyl (*http://www.standardista.com*) is a front-end engineer, author, and trainer who has been developing standards-based accessible websites since 1999. She writes two technical blogs pulling millions of visitors and speaks about CSS3, HTML5, JavaScript, and mobile wed development at conferences around the world.