

FIREFOX HACKS™

*Tips & Tools for Next-Generation
Web Browsing*



O'REILLY®

Nigel McFarlane

HACK
#44

Tweak and Troubleshoot CSS Designs

Get to the bottom of your CSS difficulties with smart Firefox extensions.

Since its debut, CSS browser compatibility issues have given web developers headaches. Support for CSS in contemporary browsers such as Firefox is very good, but there are still inconsistencies between different browsers, at least until the whole world wakes up to how good Firefox is. You'll have to deal with that.

Firefox's extensible design has encouraged the development of a number of indispensable extensions to help you troubleshoot and tweak your CSS. In this hack, we'll look at how two of these, Web Developer and EditCSS, can help you quickly track down exactly which CSS statements are causing problems.

Install Must-Have Extensions

The Web Developer extension, by Chris Pederick, is available at *update.mozilla.org* or here, and includes some EditCSS functionality:

<http://www.chrispederick.com/work/firefox/webdeveloper/>

The EditCSS extension, by Pascal Guimier and others, is available at *update.mozilla.org* and also here (beware version differences):

<http://editcss.mozdev.org/installation.html>

Install them as you would any other Firefox extension [**“Get Oriented,” at the beginning of Chapter 11**]. The EditCSS will perhaps be bundled inside the Web Developer extension eventually, but for now it is still available separately. Check the home pages for both extensions.

Use the Web Developer Toolbar to Locate Style Rules

The Web Developer extension places a number of menus in a new toolbar. There are far too many features to go into all of them here, so we'll stick to those relevant to this hack. However, do make sure you look at all of them.

One of the most time-consuming aspects of debugging and tweaking CSS is working out which CSS statements are causing the problem. More than one CSS statement can affect the same HTML element, and elements inherit CSS properties from their ancestors. The Web Developer toolbar helps isolate the relationship between a style sheet and an HTML document.

Open any page in Firefox. Once you have the Web Developer toolbar installed, it should look like **Figure 5-1**.

Particularly when you've used lots of classes and IDs, as any nontrivial layout requires, it's difficult to work out exactly which elements you are look-



Figure 5-1. Web Development toolbar installed on Macintosh Firefox

ing at on the page. And working this out is the first step in tracking down a troublesome statement.

Under the CSS toolbar menu, choose View Style Information. Then move the cursor over the page; it changes to a crosshair. The containment hierarchy for the element currently under the crosshair is displayed in the status bar of the window.

For instance, on O'Reilly's home page you might see the status bar shown in Figure 5-2.

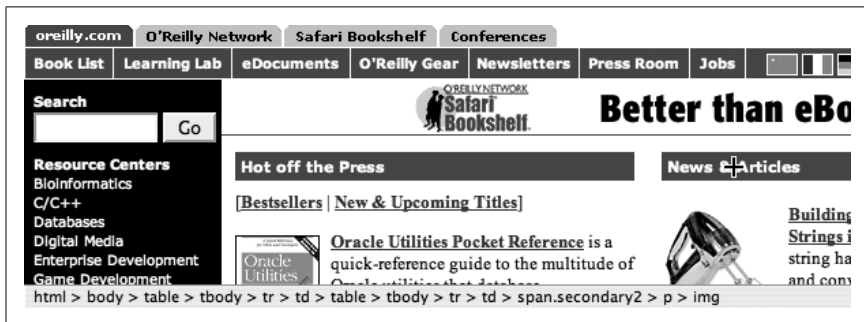


Figure 5-2. Revealing tables in an O'Reilly web page

In a flash, the status bar at the bottom tells you that O'Reilly needs to update their page-layout techniques from old-fashioned tables to modern CSS.

So, in an instant, you can see exactly which element is at a given location on the page without having to View Source, remember the content you're looking for, find it, and other laborious processes. On top of this, you get a great snapshot of the inheritance structure of that element, which is essential to working out which CSS statements might be affecting it.

However, that's not all. You can also find out exactly which CSS rules are affecting any element on the page. While in View Style Information mode,

click an element. A new tab opens, displaying all the CSS rules in all the relevant stylesheets that affect this element.

Going back to the O'Reilly example, I clicked the content near the top of the page that says “News and Articles,” which I know from the status bar is a `` of class `hdr4-w`. Here's the first set of information I saw:

```
chrome://webdeveloper/content/stylesheets/view_style_information.css
* (line 2)
{
  cursor: crosshair;
}
```

Ignore this first style: it is simply the Web Developer Toolbar's own CSS for showing the crosshair. In CSS, the asterisk (*) is a universal selector that selects any element in a document. After this first rule, we have the more relevant information:

```
http://oreillynet.com/styles/main.css
.hdr4-w (line 190)
{
  font-family: Verdana,Arial,Helvetica,sans-serif;
  font-size: 8.5pt;
  font-weight: bold;
  text-decoration: none;
  color: rgb(255, 255, 255);
}
```

First, you can see the URI for the stylesheet. Most importantly, you then get a rule in this stylesheet, which selects the specific element of class `hdr4-w`.

Working down this tab, you can see all the statements that explicitly select and style the element. If a property you are interested in is explicitly set for this element, you'll find it here. Sometimes, this will identify the problem, and you can review and modify the CSS.

Use the DOM Inspector to Find Inherited Values

Because of inheritance, not all rendered property values (called *computed values*) are explicitly set. These values are often affected by inheritance, and their source can take the longest to track down.

To find out where an element is inheriting a property from, you can turn to a built-in part of Firefox, the [DOM Inspector](#) [Hack #53], which is discussed here in the context of CSS.

Open the DOM Inspector from the Tools menu. In the left pane, you'll see the document's DOM tree; you can select any element (or node in XML/XHTML-speak) in the document here. Sometimes it can be hard to find the element you're looking for if all you can see is the DOM tree. Click Inspect

at the top right and make sure Blink Selected Element is chosen in the View menu. Then, as you click nodes in the DOM tree, they will flash in the Preview pane at the bottom.

In the right pane, use the drop-down menu to choose different property panes for a selected element. Initially, this shows DOM Node. Change it to Computed Style. Note that Computed Style won't be available until you select the `<html>` element or something below it in the DOM tree.

You'll see all of the computed values, explicit or inherited, for all of the CSS properties of the selected element.

To find out where a value is being inherited from, move up the DOM tree, watching the property in question. In the left pane, click the parent nodes, and their computed values will be displayed in the right pane. When the property you are watching changes in value, the culprit is the node directly below this in the hierarchy. The value is explicitly set at this node and then inherited all the way down to the element causing you trouble.

From here, you can quickly find the statement in question by switching in the right pane from Computed Style to CSS Style Rules. This, like the Web Developer Toolbar, shows all the statements that explicitly select this element.

Edit CSS in Firefox

Now that you've found the exact statement you need to tweak, you can actually *edit* it in Firefox and see the effects *live*. Go back to the page itself. In the Web Developer toolbar, turn off View Style Information by selecting it again under the CSS toolbar menu. Then, still under the CSS toolbar menu, choose Edit CSS. A panel opens on the left side, showing each of the stylesheets affecting this page. This is the EditCSS extension at work. If it's installed separately, start it from the Tools menu.

Not only can you view the stylesheets with this sidebar, but you can also edit their text directly, and the page is updated as you make changes. Note that you are not making changes to the actual stylesheet that is linked to the page when you do this. However, you can save what you have created at any time. This is a real bonus, because it makes it easy to do numerous edits and quickly create numerous versions to return to if necessary as you work.

With the EditCSS panel, you can modify properties in your stylesheet in real time and see the results instantly, which is invaluable when tweaking and debugging. To use just one aspect of CSS coding as an example, many page layouts these days rely on the complex interaction of positioning and box model properties such as margin and padding. It can be difficult to get these

Tweak and Troubleshoot CSS Designs

layouts to work right. With the EditCSS panel, you can change property values and instantly see the effect on a layout.

—*John Allsop and Maxine Sherrin*