Implementing Typographic controls in Dreamweaver

Michael Geraci
Pacific University

Follow this and additional works at: http://commons.pacificu.edu/inter07

Recommended Citation

This Article is brought to you for free and open access by the Interface: The Journal of Education, Community and Values at CommonKnowledge. It has been accepted for inclusion in Volume 7 (2007) by an authorized administrator of CommonKnowledge. For more information, please contact CommonKnowledge@pacificu.edu.
Implementing Typographic controls in Dreamweaver

Rights
Terms of use for work posted in CommonKnowledge.

This article is available at CommonKnowledge: http://commons.pacificu.edu/inter07/2
Implementing Typographic controls in Dreamweaver

Posted on February 1, 2007 by Editor

By Michael Geraci

About

Part One: The need for improved typography
Part Two: What the research says about typographical improvements
Part Three: Typographical Control via Cascading Style Sheets
Part Four: Implementing Typographic controls in Dreamweaver

It’s hard to have a prolonged discussion about Web design and development without reaching the subject of Dreamweaver. Dreamweaver is to Web creation as Photoshop is to image editing, it’s the powerhouse application that slices, dices, and enjoys an estimated 80% market share in the industry. Dreamweaver’s similarities to Photoshop don’t end with it’s status in the marketplace, the two also share a daunting feature set that provides one with endless learning opportunities and frustrations alike. That said, when you’re serious about creating Web sites, Dreamweaver is a foregone conclusion. Even die-hard hand-coders have implemented some specialized workflows that harness Dreamweaver’s code constructing prowess.

The matter at hand is web typography — I’ll save the in-depth Dreamweaver discussions for a future 20-part series — and fortunately, setting up and managing your global typographic styles is easy enough in Dreamweaver that I should be able to do it justice in this space.

First let’s look at the options. If you’re content manually editing HTML/CSS code, then Dreamweaver’s code view works like any text editor except you get color coded syntax which makes working with the code a lot easier. Figure 1 shows a portion of some typographical rules in this format. Be warned that Dreamweaver is a stickler for CSS, so if you’ve already made some edits to the current page, there’s a good chance that there are already some CSS rules present in the <head> of the document. For example, formatting some text via the properties palette will result in the creation of a CSS class called .style#, where # is a unique number based on the order in which this rule was created (see figure 1). We’ll talk about CSS classes later, but...
for now, you need to know that some of Dreamweaver’s generated rules may affect the ones you want to establish for your type. If you find that Dreamweaver’s rules are conflicting with yours, simply delete the rule from the stylesheet, or if you’re up to it, delete just the declaration that is in conflict. For example, a .style rule might be controlling the typeface and weight (boldness) of some text. If you want the boldfacing style but not the typeface setting, just highlight and strike the font-family declaration from the rule and keep the one for font-weight.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Good Typography Rules</title>
<style type="text/css">
body, td, th {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 11px;
    color: #000033;
    width: 500px;
    line-height: 1.5;
    text-indent: 20px;
}

h1, h2, h3, h4, h5, h6 {font-family: Georgia, Times New Roman, serif;}

h1 {font-size: 16px;}
h2 {font-size: 14px;}
h3 {font-size: 12px;}
</style>
</head>
```

*figure 1: Dreamweaver’s code view*

If you’d rather not edit in code view directly, you can let Dreamweaver write your CSS rules by making selections in the appropriate dialogue boxes. Probably the most important place to set page-wide typographic styles is in the Page Properties dialogue. This easy-to-use set up panel allows you to specify many of the more important type settings for your entire document. To access the Page Properties dialogue, select it from the Modify menu. Once it’s open, click on the appropriate category at the left (e.g., Appearance), and enter your settings in the series of input fields and pull-down menus on the right (e.g., Page font). Even color selection is simplified through the use of swatch palettes that enter hexadecimal values for you. Once you’ve made all your selections hit the OK button and your style rules are written for you in the document’s `<head>`.

Finally, there’s the CSS Styles panel, which can be accessed via the Window menu. This multi-faceted panel provides you with a command center for all your CSS needs. I find that the CSS panel is more useful as a means to edit and append existing rules in your document, rather than creating them anew. Once you have some rules established for your document via any of the above means, you can select their associated selectors in the CSS panel and alter the existing values or add new properties via the “add property” button at the bottom. For example, let’s say that I had set my page font to 11px Verdana via the Page Properties dialogue, but I want to also set my line-height (leading) to 1.5. The `<body>` selector appears in the CSS panel as soon as I establish any rules for it, so I can select it and go about adding the line-height declaration. Selecting the “add property” link in the panel will cause an input field to open up in the panel.
where you can either type the property name in directly or select it from a pull-down menu that appears next to the field. Once a property is entered, the corresponding input fields will appear for you to enter your value and the appropriate units. This is a great way to manage your typographical styles without having to work directly in the code. If you want to delete a declaration, simply select it and click the tiny trash icon at the bottom-right of the panel.

One of the advantages of using Dreamweaver to create and manage your CSS styles is the instant gratification of seeing the changes render in the document as soon as you make them, provided you’re not in code view, of course. This saves you from having to save the file and view it in a browser every time you want to inspect the current status of the page. However, you should get in the habit of inspecting the page in a browser every so often. While Dreamweaver’s design mode is pretty accurate, especially with text presentation, you’d be surprised how often things can be different in the browser. Once again, Dreamweaver expedites this process by giving you a Preview in Browser button at the top of every document. though it appears like a standard button, clicking this object will pop open a menu that you can populate with any and all of the browsers resident on your computer.

Another Dreamweaver feature that can make your life easier is the option to Highlight Invalid Code. When turned on via the View Options menu at the top of your document, this will underline all HTML/CSS that does not conform to the Web standards as defined by the \(<a href=\&gt;W3C</a>\), and will likely cause some error in the display of your page.

The next and final part of this series will cover two slightly more advanced techniques in CSS/Dreamweaver. I’ll cover the creation and use of linked stylesheets, where one stylesheet governs an entire site’s presentation; and I’ll conclude with a discussion of CSS classes, which allow specialized rules to be applied to specific objects on the page.

**Further reading:**

CSS Styles in Dreamweaver 8 at Community MX
Intro to CSS with Dreamweaver 8 at Senocular
An overview of CSS in Dreamweaver at Adobe’s Developer Center

This entry was posted in Uncategorized by Editor. Bookmark the permalink [http://bcis.pacificu.edu/interface/?p=3329].