



## Roadmap Series

Issue 3

# Styling Forms with CSS

by Joseph Lowery

03-12-09



**WEBASSIST**<sup>®</sup>  
Build better websites, faster.

Excerpt from the Dreamweaver 8 Bible by Joseph Lowery, published by Wiley Publishing, Inc.

ISBN 0-471-76312-8

# Styling forms with CSS

Produced by Joseph Lowery

In many ways, forms are the real workhorses of the Web-but that doesn't mean they have to be plain. Until CSS use became prevalent, little could be done to alter the way forms and form elements looked on the Web. Standardizing text field sizes between PC and Macintosh was a problem because the different operating systems interpreted character width differently; moreover, the field sizes may vary from browser to browser.

CSS gives the form designer much more flexibility, both to integrate and isolate the form and form elements. Text fields, for example, can take on a shade of a site's background color or adopt the same typeface used on the page. Similarly, you can draw attention to the form itself by giving it a contrasting background; this enables you to format lengthy drop-down lists for easy reading.

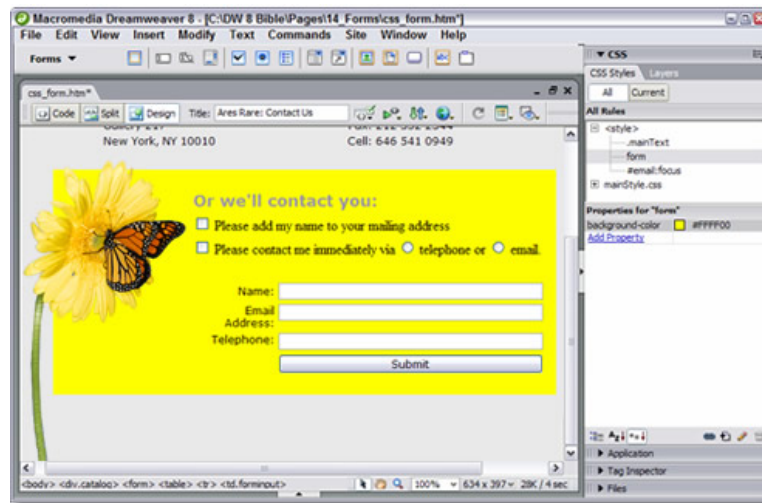
## Encompassing the Form

The `<form>` tag is a containing element that, like the `<div>` tag, is not rendered by default. Both tags, in fact, can be styled with CSS-you can even position a form on the page via CSS declarations. Browser support varies for some of the more esoteric CSS properties applied to the form tag, but more common attributes such as background color and border are rendered properly in most cases. Best of all, if CSS does not support certain attributes, these attributes are just ignored and the form renders plainly.

Frequently, a Web page only contains a single form. In these situations, styling the `<form>` tag itself will have the desired results. For example, this CSS rule gives the entire form a bright orange background and a blue border:

```
form {
background: #FF9900;
border: thin solid #0000FF;
padding: 10px;
}
```

Padding is added to move the form elements in from the edges, as shown in Figure 1.



**Figure 1:** Apply CSS coloring to the form to make it stand out on the page.

Should your page contain more than one form and you want to style each one differently, create a CSS ID selector for each form. In this situation, choose the Advanced selector in the New CSS Rule dialog box and enter a unique ID - such as `form#topform` or `form#bottomform` - in the Selector field. Also set the ID of the form tag when using this method.

## Roadmap Series

Issue 3



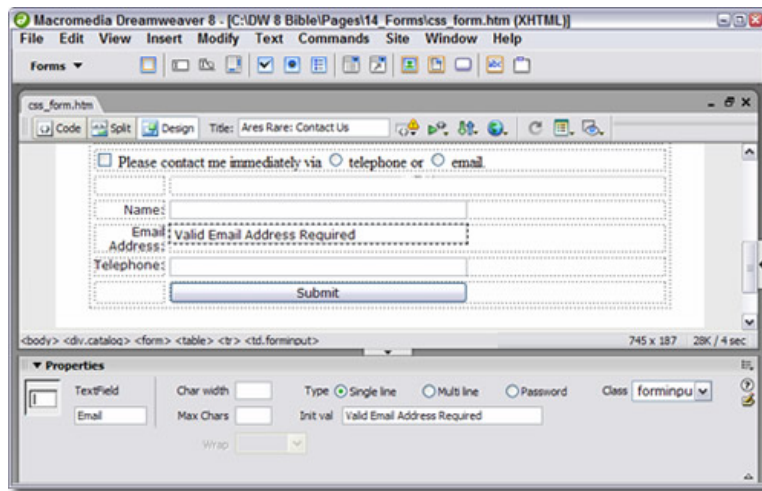
## Altering Input Fields

One way in which the `<form>` and `<div>` tags differ in regard to CSS is in the matter of inheritance. Elements within a form do not inherit the CSS properties of the form, but elements within a `<div>` tag do inherit the div's CSS attributes. You must, therefore, take another route for styling all the text fields in a given form. The best way to affect multiple form elements all at once is to style the `<input>` tag. You may recall that the `<input>` tag is used to create text fields, radio buttons, checkboxes, and Submit buttons.

For example, this CSS rule gives all the input elements a uniform background color as well as a specific color, font, and size for the text fields:

```
input {
    background-color: #F5F5F5;
    color: #660099;
    font: 10px Verdana, Arial, Helvetica, sans-serif;
}
```

CSS styles the text fields for initial text as well as text entered by the user, as shown in Figure 2.



**Figure 2:** Keep the text in your text fields looking like the rest of your page through CSS styling of the input tag.

**Tip** Remember that a multiline text field is really a different tag `<textarea>` than the single-line text field. You have to create a CSS rule for both input and textarea tags.

## First steps

One of the best uses for CSS and text fields is setting the width. This method is far more flexible and responsive than using the Char Width field on the Property inspector for each individual text field. It is best to set the width on a CSS class rather than alter it directly in the CSS rule for the input tag. Why? The width setting not only affects all the single-line text fields, but it also alters checkboxes and radio buttons - which are also input tags. After the CSS rule is defined, set the class of a selected text field from the Property inspector.



## Distinguishing Lists and Menus

The select list/menu object is composed of two tags: `<select>` and `<option>`. The `<select>` tag is the overall container for the list items; use `<select>` to style the width, typeface, and font size of all drop-down lists on the page uniformly. Individual list items can be styled by setting a class on the separate option tags. Although this operation must be performed by hand and is somewhat tedious, it does open the door to many possibilities.

If you have a very long drop-down list that includes a wide-range of items organized by category, with judicious CSS styling, main category headings can be in one color and subitems in another, as shown in Figure 3.

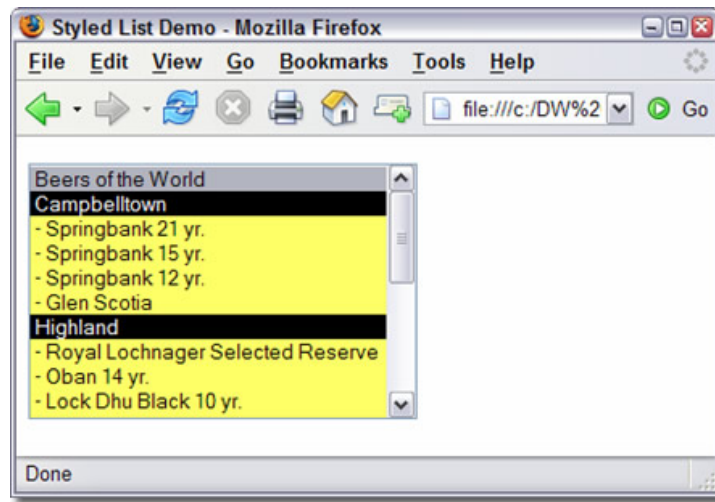


Figure 3: CSS classes for menu items must be applied by hand to separate `<option>` tags.

## Roadmap Series

Issue 3

## Changing Labels and Legends

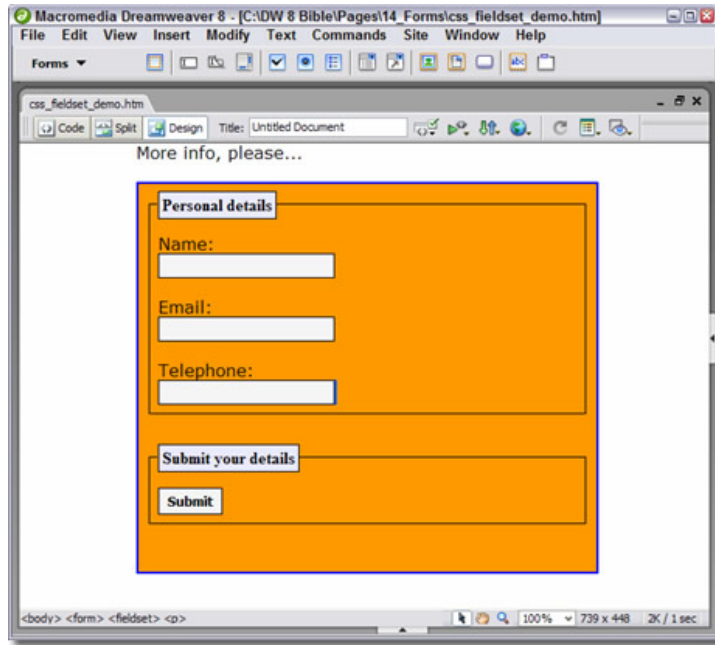
A form is more than a collection of text fields and checkboxes; labels play an equally important role in form organization and usability. Form labels are often applied in one of two ways. The standard technique is to place most of the labels in a single column of a table, with the form elements in another. Designers are also increasingly using the `<label>` tag as a means of enhancing accessibility. A Dreamweaver CSS methodology is available for whichever route you take when labeling your forms.

In a situation where all the labels are arranged in a table column, it's best to create a CSS class for your labels and apply it to the `<td>` tags. The most efficient way to do this is to first select the column containing the labels and then choose the desired class from the Style list on the Property inspector. Dreamweaver applies the selected class to each of the `<td>` cells in the column.

If your layout uses `<label>` tags, CSS control is even easier. Add a specific CSS style for the `<label>` tag to create a uniform appearance for all your labels. Note that you may still need to adjust the dimensions of the label column separately because setting the width in CSS for the `<label>` tag has no effect.

Two other form-related tags - `<fieldset>` and `<legend>` - are available for CSS styling. As described earlier in this chapter in the sidebar "Grouping Form Controls," the two are used together to visually associate related form elements. Style the `<fieldset>` tag to alter the outlining border or add padding from the edge of the border. Change the `<legend>` style when you want to give it a separate background color and/or border, as shown in Figure 4.





**Figure 4:** Style the `<fieldset>` and `<legend>` tags to really make them stand out from a form with a colored background.

## Highlighting Focus

Want to spotlight the interactivity of a form? CSS includes a pseudo-element selector (so called because it is valid only when an element is in a particular state) that takes effect when a form element is selected. The CSS selector is `:focus` and it works with `<input>`, `<select>`, and `<textarea>` tags.

To create a `:focus` selector, follow these steps:

1. Select New CSS Rule from the CSS Styles panel.
2. In the New CSS Rule dialog box, select the Advanced selector option.
3. Enter the name of the tag you want to affect followed by `:focus` in the Selector field. For example, if you wanted to alter all the text fields, radio buttons, checkboxes, and buttons when they receive focus, enter `input:focus`.
4. Click OK to close the New CSS Rule dialog and open the CSS Rule Definition dialog.
5. Choose the desired styles from the various categories and click OK when you're finished.

Preview the page in a compatible browser such as Mozilla Firefox, Netscape 6 or better, Safari, and so on to experience the CSS changes (Figure 5).

## Roadmap Series

Issue 3



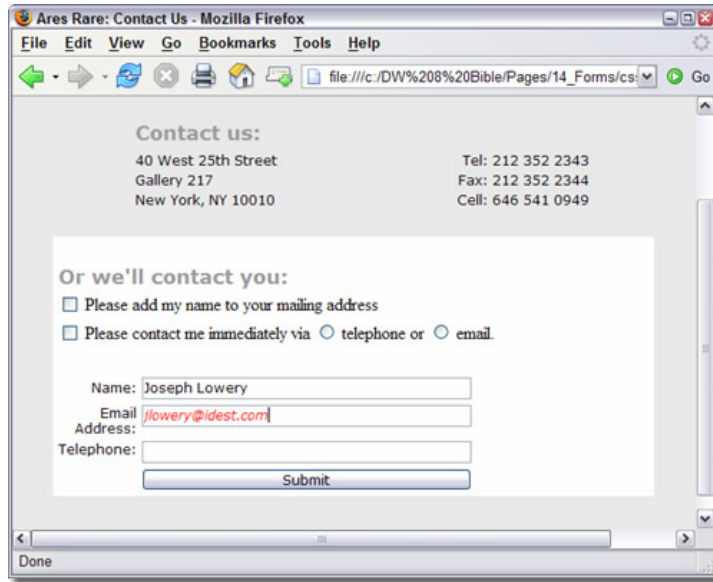


Figure 5: Text in a selected field is shown in bright red and bold, thanks to the `:focus` selector.

## Roadmap Series

Issue 3

**Caution** Unfortunately, the `:focus` selector is not supported in any version of Internet Explorer, as of this writing. You can, however, simulate the effect by triggering JavaScript functions with the `onFocus()` and `onBlur()` events that manipulate the class attributes.

## Join our Community

From free software and training, to special offers, there is something for everyone. Come [take a look](#) at what is waiting for you and [sign up](#) today to gain access to all of the valuable benefits awarded to the members of our community.

## About WebAssist

WebAssist helps you build better websites faster by offering software, solutions, and training needed to succeed on the web. WebAssist.com hosts a thriving community of over 300,000 designers, developers, and business owners. WebAssist's partners include Adobe, Microsoft, and PayPal.



WebAssist.com Corporation

8899 University Center Lane  
San Diego, CA 92122

USA

[www.webassist.com](http://www.webassist.com)