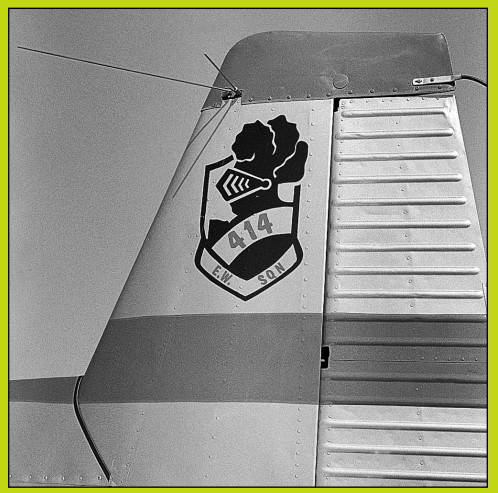
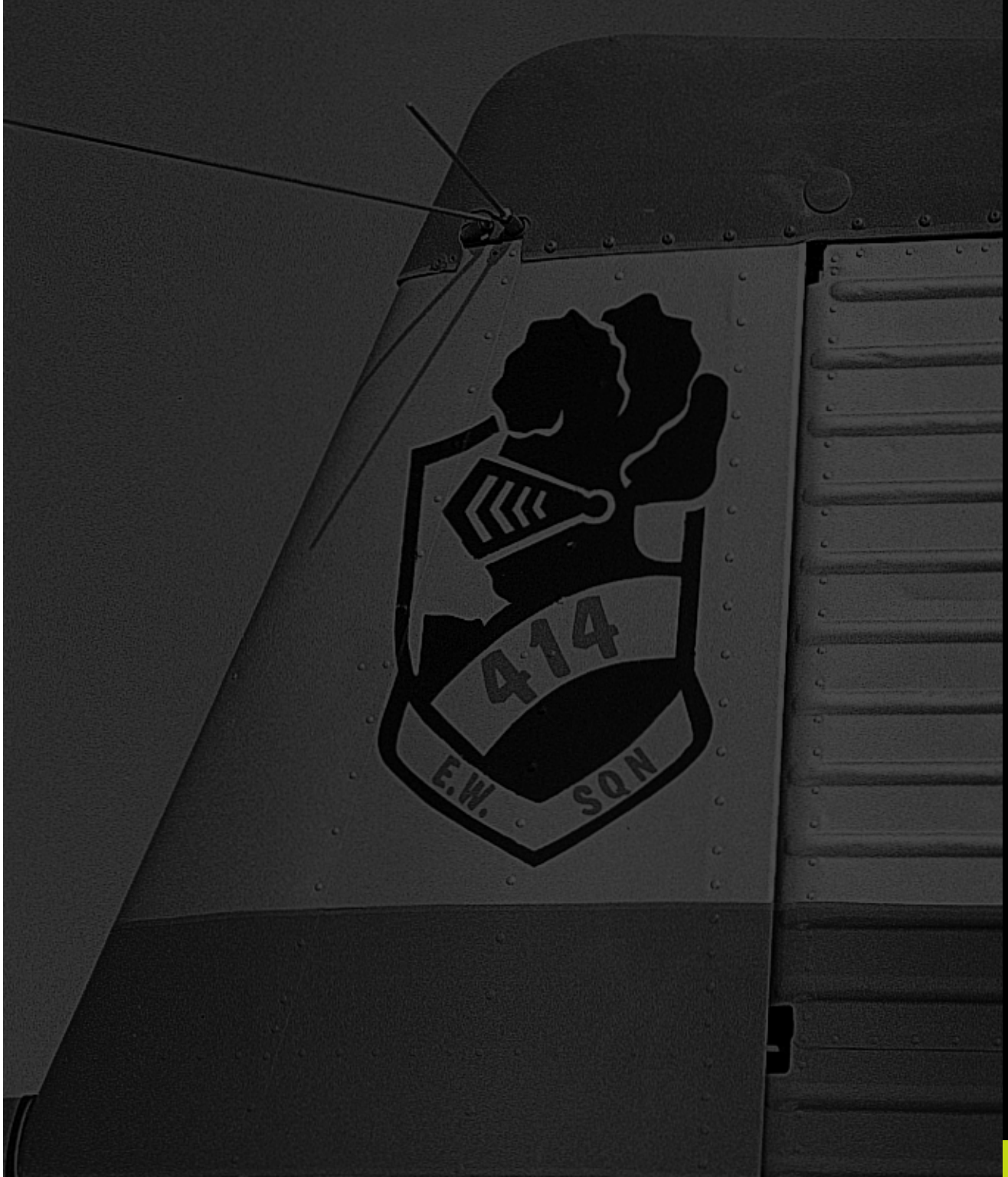


USING CASCADING STYLE SHEETS TO MAKE SELECTABLE THEMES

"While browsing through the Web one day, it occurred to me that I was not alone. Then it all made perfect sense."

—ANCIENT PVII PROVERB



The Incredible CSS Cookie Monster

Wouldn't it be cool if you could offer your site's visitors a choice of design themes? What if you could make it so that whatever theme a visitor chose would stick with him throughout his session? And maybe it would be totally cool if a visitor's chosen theme could be remembered so that if he returned, say, a week later, the same theme would be selected.

All in all, what we have here is a technique with an off-the-scale wow factor. Imagine that

- You have a persnickety client who can't decide whether he wants blue text or green text (yuck).
- Your boss wants dark text on a white background, but her boss wants light text on a dark background.
- Your husband wants your family business Web site to look like the Cleveland Browns team uniform, but you prefer something a bit more subtle.

Well, you could use this technique to make everyone happy. Just give them all a link to their own preferred style (they can draw straws over whose style is the default). Or you could use it as the ultimate prototyping tool to quickly show your clients different versions of the same page with just a click of a link.

The Incredible CSS Cookie Monster is a peacemaker of grand proportions. We rank it up there with Roosevelt, Churchill, and Ghandi. Of course, they never had this technique available to them or history might well have taken a completely different course.

Project 1

Using Cascading Style Sheets to Make Selectable Themes

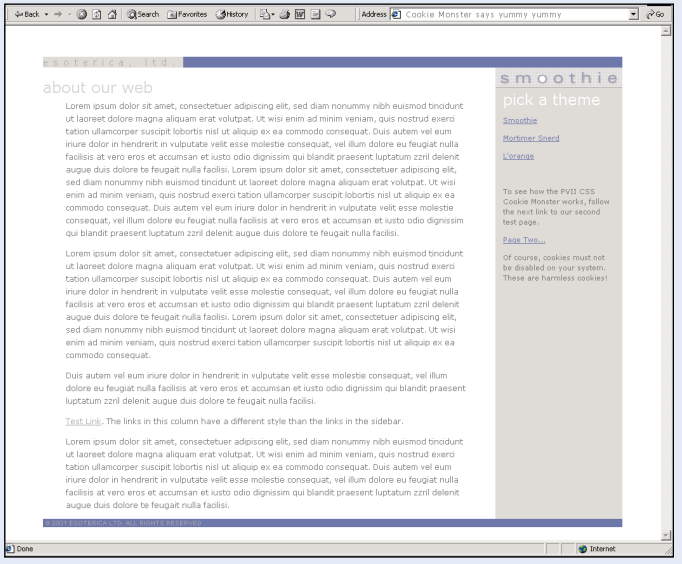
by Al Sparber and Gerry Jacobsen

It Works Like This

A visitor is presented with a heading that says something like "Choose a Theme." Under the heading is a set of links. Each link describes a theme. For example, Soft Blue, Lemon Yellow, Happy Holidays, and Large Text.

Upon clicking one of the links, a small JavaScript in the page writes a little cookie to the visitor's system. The cookie simply states the name of a style sheet to use and how long the cookie will remain on the visitor's system. It takes a millisecond to write the cookie, after which the script reads back the cookie, tells the page which style sheet to use, and reloads the page to make the changes take effect.

The magic begins when the visitor goes to another page. Our little script reads the cookie as it's loading the new page and voilá! The new page loads with the selected theme. But the magic doesn't stop here.



Our page is rendered with its default style sheet.

The same page after selecting the Mortimer Snerd theme.

The cookie contains not only the style sheet name, but also an expiration date (that you set). So long as the cookie has not expired, your visitor's selected theme will be chosen every time he visits your site. This is one smart cookie.

What happens if a visitor has cookies disabled? That's simple, we are going to place a default style sheet link in our page. If cookies are not allowed, the default theme (based on the default style sheet) will work perfectly.

A Class Act: CSS is the straw that stirs it all.

Our JavaScript works by reading a cookie that tells the browser which style sheet to use. The style sheet used determines the theme. Here's a little example:

Theme1.css

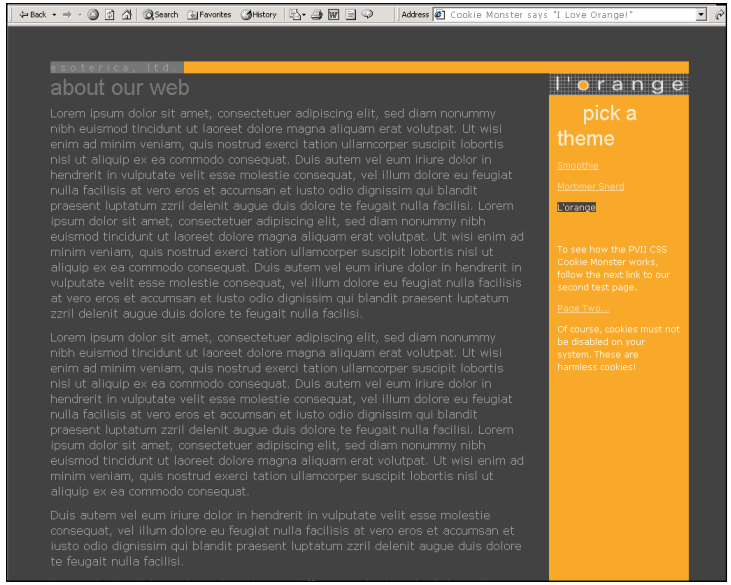
```
.examplestyle {  font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px; color: #000000}
```

Theme2.css

```
.examplestyle {  font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 16px; color: #FFFFFF}
```

The secret is to make sure that each CSS file contains the same classes. If, for example, a paragraph on your page is assigned the .examplestyle class, it will reflect the rules contained in the CSS file relating to the chosen theme. If a visitor has chosen Theme1, the text in the paragraph will be 12px black. If Theme2 is chosen, the same paragraph will be 16px white. Why? Because both style sheets contain the same class (.examplestyle) with different rules. Obviously, then, Cascading Style Sheets are vital components of this project!

The three CSS files necessary for this project are included in the assets/style_sheets folder. We're going to set up our home page now, but don't worry, we will have a full style sheet discussion later.



The same page after selecting the L'Orange theme.

Driven by Styles

The CSS classes assigned to your page elements are contained in multiple style sheets. Each style sheet contains the same named classes; we just change the style rules in each sheet. This causes our page to transform.

Note: You may notice that the code in this book sometimes wraps to the next line. That wrapping is caused by the narrowness of the column within this book. Dreamweaver does not restrict the length of the line; the code appears on one line.

Preparing to Work

Let's go through the routine. You need to install some extensions, copy the project folder from the CD to your hard drive, and then fire up Dreamweaver and define a new site using the copied project folder as the local root folder.

1. Install the P7 CSS Styles Cookie Monster Suite:
   - Browse to the Extensions folder on the CD.
   - Open the Extensions\Commands subfolder.
   - Double-click p7setCSS.mxp to install the suite. The suite contains both a command and a behavior.

2. Install the NN4 Return False Fix and the Scrubber commands:

   The NN4 Return False Fix command eliminates Netscape showing the hourglass cursor when clicking a null link to fire a JavaScript. This command must be applied for our project to function correctly in Netscape. The Scrubber can be used to eliminate the focus lines around clicked links in MS Internet Explorer.
   - Browse to the Extensions folder on the CD.
   - Open the Commands subfolder.
   - Double-click the file called N4 Return False Fix.mxp to install the command.
   - Repeat to install scrubber.mxp.

3. Copy the projects folder:
   - Browse to the projects folder on the CD.
   - Copy the project_one folder to a convenient location on your hard drive.

4. Define a new Dreamweaver site using the project_one copy as your local root folder.

Browser Compatibility

We have tested this interface and found that it is fully functional in the following browsers:

MSIE 4 (Windows and Mac)

MSIE 5 (Windows and Mac)

MSIE 5.5 (Windows)

NN4.08–4.76 (Windows)

NN4.5 (Mac)

Opera 5.01 (Windows)

Netscape 6 (Windows and Mac)

Note: If you want to see how the completed site looks in either Dreamweaver or your browser, you can find all the files in the test_site folder.

Tip: At Project Seven we use the PC platform to develop extentions. Macintosh systems will sometimes not recognize the MXP file format. However, double-clicking the file will still execute a proper installation of the extension.

Special Notes

This project presents a very powerful tool. To implement it correctly, you need to remember these points concerning both the design and the underlying code. The points raised here are covered in detail during the course of the project. This section serves only to emphasize their importance.

Previewing

The site you build will only function on a remote server (to ensure Netscape 4 compatibility). If, at any time, you wish to preview a locally working version of the site, use the index page inside the test_site folder. The test files have been specially prepared for local previewing in MS Internet Explorer and Netscape 6. The test files are linked to your real working style sheets, so this is a good place to do your editing, too.

Absolute Paths

Netscape 4 requires you to use absolute paths to your CSS files because of the way the script works. There is no other way. An absolute path looks just like a normal hyperlink. In a normal page, a linked style sheet would have a document relative path and might look like:

../assets/styles/mystyles.css

If your Web domain is www.boris-natasha.com, then the absolute path to the above style sheet would be:

http://www.boris-natasha.com/assets/styles/mystyles.css

However, you do use a document relative style sheet link in the head of your working page. It's there so that you see a styled page as you work in Dreamweaver. Once the page is up on the Web, your JavaScripts take over and actually write the CSS links on the fly.

Style Sheet Position

The position of the style sheet in your page is important, too. Even though you are using absolutely pathed CSS files written on the fly by our scripts, you still have a presentational style sheet relatively pathed in your page (so you see something pretty as you work!). If you attempt to have Dreamweaver link a style sheet to a page that contains any JavaScript, Dreamweaver will write the link below the closing **</script>** tag. Under normal circumstances this wouldn't matter. But this technique requires that the style sheet link be in the proper position—above the opening **<script>** tag. If you follow the project step-by-step, this won't be an issue. It will only be a problem if you use Dreamweaver's CSS panel to change the CSS link after you've inserted the JavaScripts. Why is this a problem? Our scripts write the style sheet link to the proper position above the opening **<script>** tag and below where a normally linked style sheet would be declared. If there is a style sheet linked to below the closing **</script>** tag, then it will be given precedence to any style sheets listed above it. We have included instructions in the project that explain how to manually change the position in your source code.

The NN4 Return False Fix

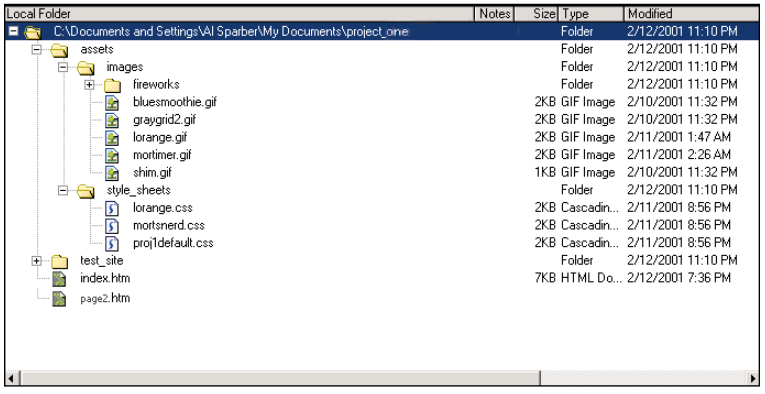
This command places return false; after the onClick events that fire the CSS Cookie Monster scripts. The command must be applied to your page or the script will not work correctly in Netscape browsers. The NN4 Return False Fix can be applied as many times as you please. It will only fix those events that need fixing! So it's a good idea to apply it after an editing session just in case!

Now, on with the show!

Creating the Home Page

In this section, we add three tables to form the structure of our design. If at any time you want to see the finished pages, you will find them in the test_site folder. The reason we've used test_site rather than the project folder will become apparent as we proceed.

1 Open the file named index.htm in the root of your project_one site.



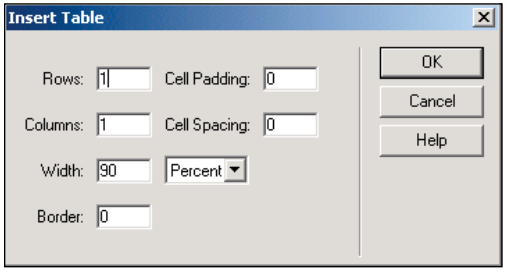
Open the file named index.htm in the root of your project_one site in the Dreamweaver Site window.

2 Insert a new table:

- In the Insert Table dialog, enter these settings:

  Rows: **1**
  Columns: **1**
  Width: **90%**
  Border: **0**
  Cell Padding: **0**
  Cell Spacing: **0**

- Use the property inspector to set Align to Center.



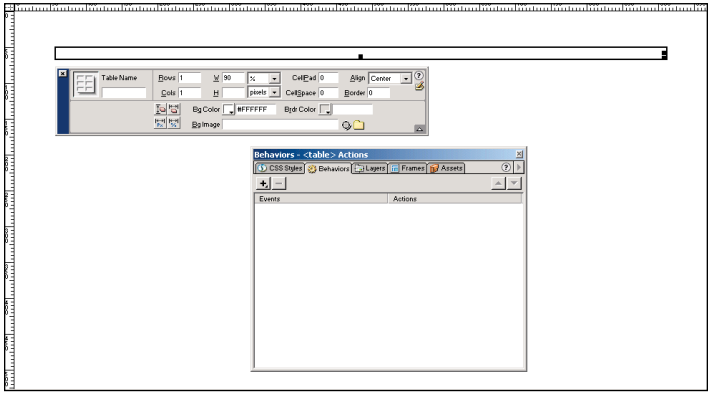
In the Insert Table dialog, enter the settings for the new table.

3 Insert a second table:

- In the Insert Table dialog, enter these settings:

  Rows: **1**
  Columns: **2**
  Width: **90%**
  Border: **0**
  Cell Padding: **0**
  Cell Spacing: **0**

- Use the property inspector to set Align to Center.



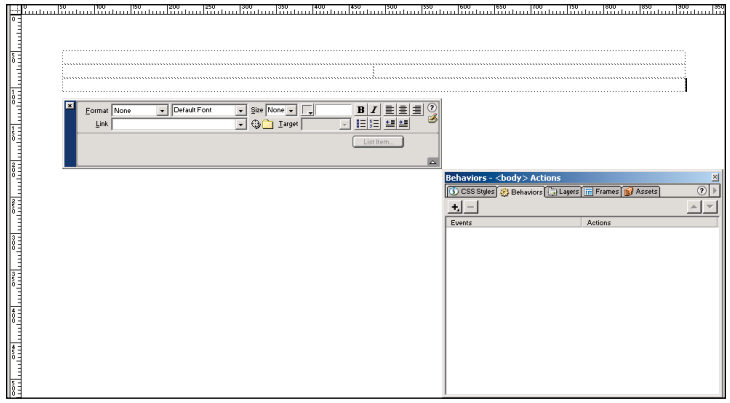
The first table should look like this after you've inserted and centered it.

4 Insert a third table identical to the first:

- In the Insert Table dialog, enter these settings:

  Rows: **1**
  Columns: **1**
  Width: **90%**
  Border: **0**
  Cell Padding: **0**
  Cell Spacing: **0**

- Use the property inspector to set Align to Center.



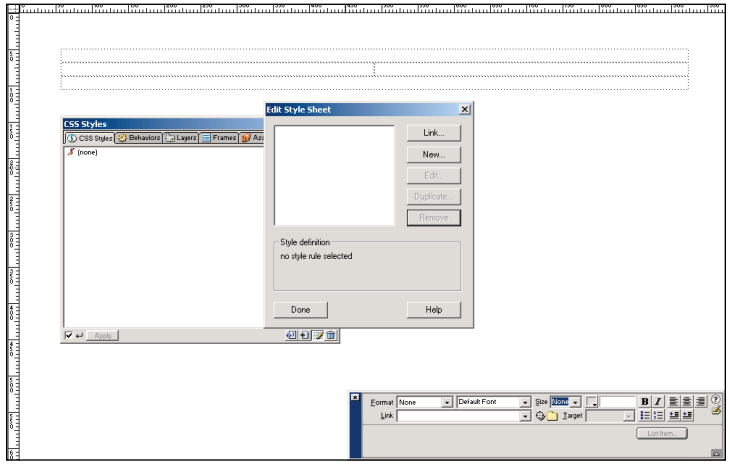
Your page should look like this when you have all three tables inserted and ready to be styled and populated.

Populating the Tables on the Home Page

We've included all the style sheets you'll need and all are preformatted for you. Before we can apply styles to the tables, we need to link the default CSS to the home page.

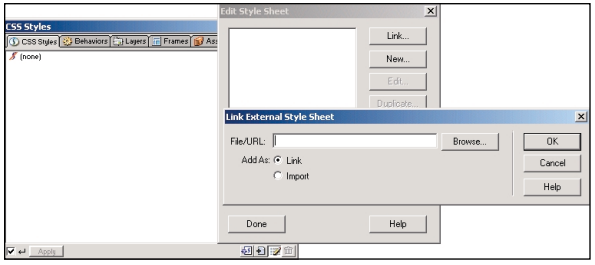
1 Link the default style sheet:

- Click the Pencil icon at the bottom of the CSS Styles panel to open the Edit Style Sheet dialog.



Click the Pencil icon at the bottom of the CSS Styles panel to begin linking a CSS file.

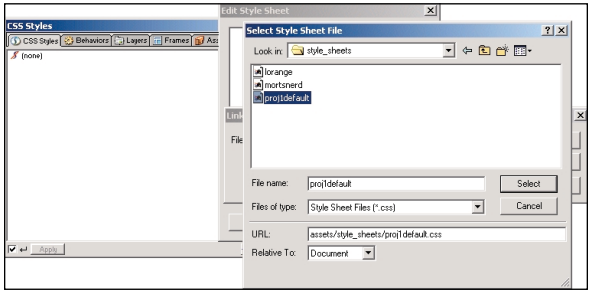
- Click the Link button on the Edit Style Sheet dialog, and the Link External Style Sheet dialog opens.
- Browse to proj1default.css in the style_sheets sub-folder of the assets directory, off the root of your project_one defined site.



Click the Link button on the Edit Style Sheet dialog and then click the Browse button on the Link External Style Sheet dialog.

- Click the Select button to execute the link and then click the Done button on the Edit Style Sheet dialog.

  The CSS file is now linked to the home page.



Browse to the style_sheets folder in the assets directory, choose proj1default.css, and click the Select button.

2 Enter the header text in the top table:

- Click inside the top table and type **esoterica, ltd.**

3 Enter the middle table header and placeholder text:

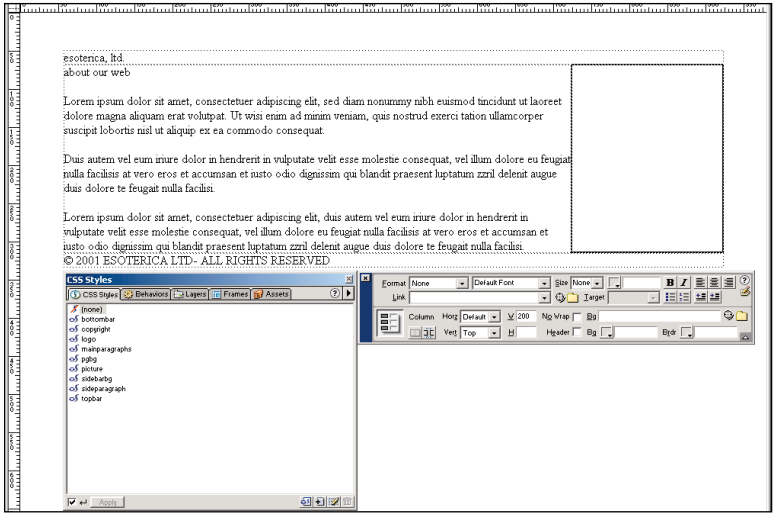
- Click inside the left column of the middle table and type **about our Web**.
- Press Enter and type a few paragraphs of placeholder text.

4 Enter the copyright in the bottom table:

- Click inside the bottom table and type **© 2001 ESOTERICA LTD- ALL RIGHTS RESERVED**.

5 Size the right column of the middle table:

- Click inside the right column of the middle table and set its width to **200** on the property inspector.



After typing in the headings, copyright, and placeholder text, set the right column of the middle table to a width of 200 on the property inspector.

Tip: Sometimes tables will collapse when you enter text inside one cell. To get to the next cell to the right, press your Tab key while in the left cell. To move to the next cell left, press Shift+Tab.

6 Enter the sidebar heading, links, and text:

- Click inside the right column and type the following:

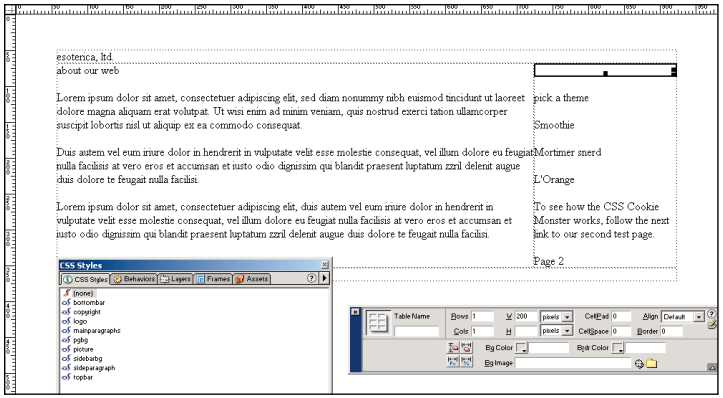
  pick a theme

  Smoothie

  Mortimer snerd

  L'Orange

  To see how the CSS Cookie Monster works, follow the next link to our second test page: Page 2



After typing in the sidebar content, insert a 200-pixel wide table inside the right column of the middle table.

7 Insert a nested table inside the right column:

- Place your cursor just to the left of the "p" in pick a theme and insert a table inside with these settings:

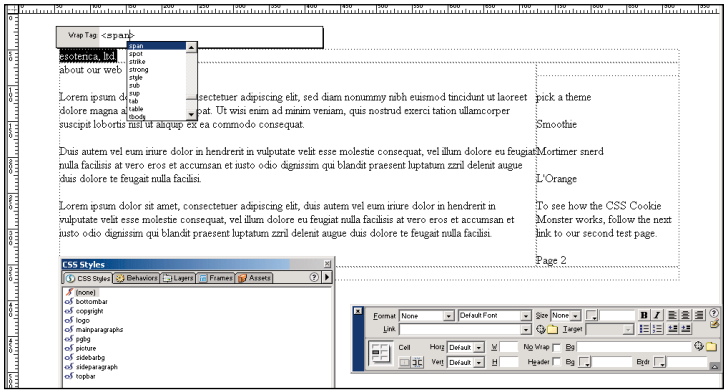
  Rows: **1**
  Columns: **1**
  Width: **200 pixels**
  Border: **0**
  Cell Padding: **0**
  Cell Spacing: **0**

Assigning CSS Classes to the Home Page Elements

By assigning classes to headers, text, and table cells, we'll see the home page begin to take shape.

1 Assign a class to the header in the top table:

- Select the text esoteric, ltd.
- Right-click (Ctrl+click) and choose Wrap Tag.
- Click inside the little box that pops up with an empty tag <>and type **span** so that it looks like ****.
- Press Enter to close the box.
- Insert your cursor somewhere inside the text esoterica, ltd.

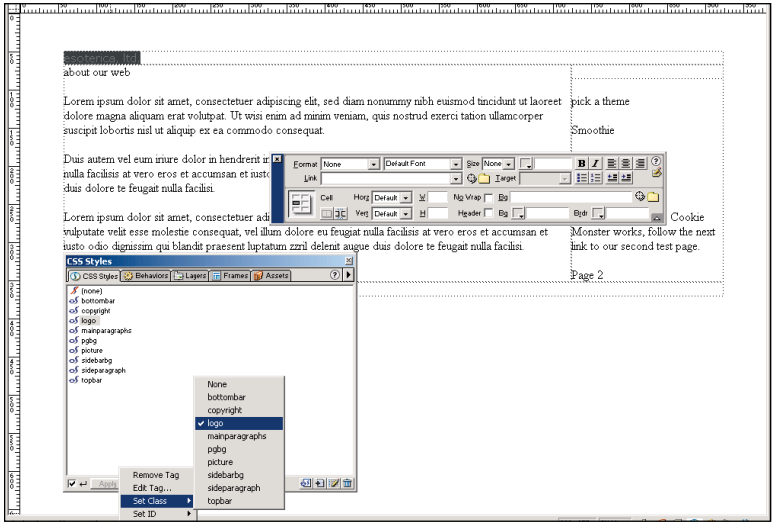


Use Dreamweaver's powerful Wrap Tag feature to insert a **** tag pair.

- Select the **** tag on the Dreamweaver status bar (lower-left window border) and right-click (Ctrl+click).
- Choose Set Class/logo.

  The text is now dark gray on a light gray background.

Note: We are using several methods of applying CSS classes because Dreamweaver behaves differently under certain situations, and we have chosen the methods that work most accurately in each situation.

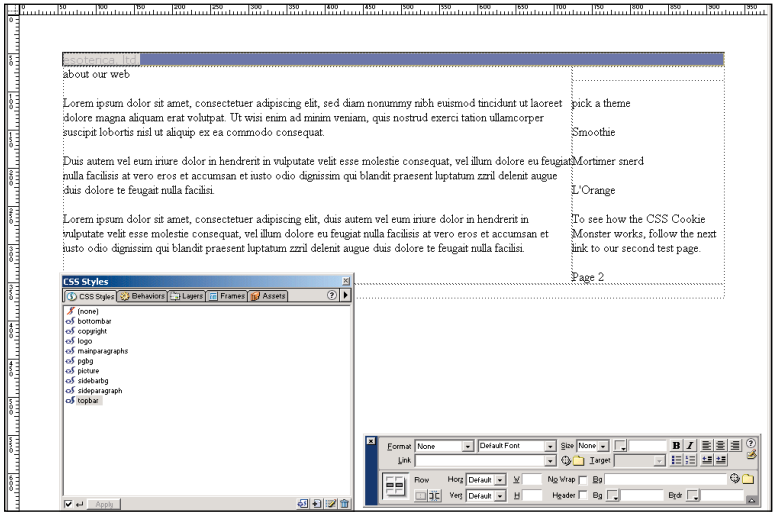


Set the CSS class of the **** tag via the status bar context menu.

2 Assign a class to the top table's row:

- Insert your cursor in the top table row (next to esoterica, ltd.) and right-click (Ctrl+click) the **<td>** tag on the status bar.
- Choose Set Class/topbar.

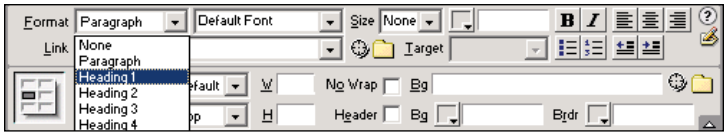
  The table row is now colored blue.



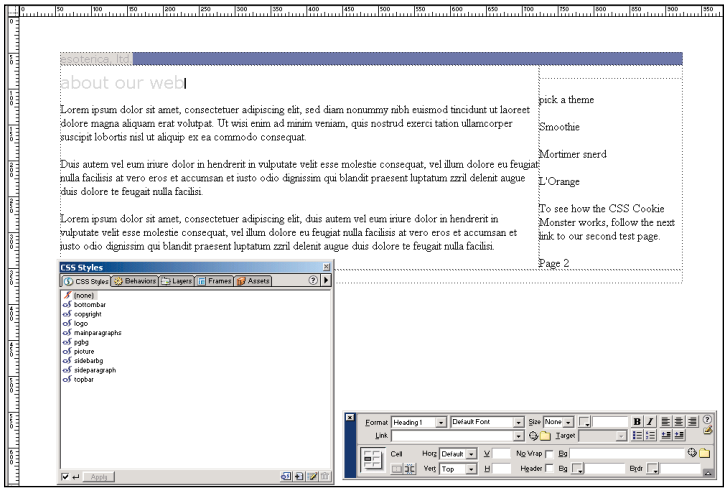
Apply the topbar class to its **<td>** tag to make the top table row turn blue.

3 Assign a class to the heading in the left column-middle table:

- Place your cursor on the same line as the heading "about our Web."
- Drop down the format menu on the property inspector and choose Heading1.



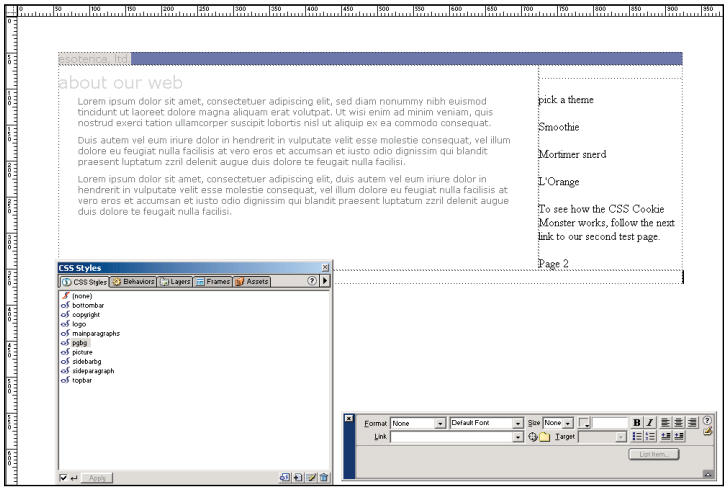
Apply the H1 format to the headings.



When the H1 format is selected, the heading becomes instantly styled.

4 Assign a class to the paragraphs in the left column-middle table:

- Place your cursor inside the first paragraph below the "about our Web" heading.
- Right-click (Ctrl+click) the **<p>** tag on the status bar.
- Choose Set Class/mainparagraphs.
- Repeat this class for the remaining paragraphs in the column.

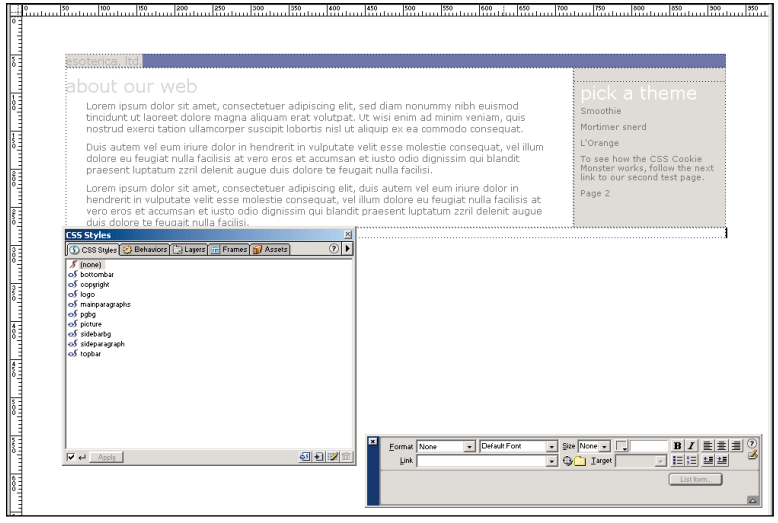


Apply the mainparagraphs class to each **<p>** tag in the first column.

5 Assign classes to the sidebar and its content:

- Place your cursor on the same line as the heading "pick a theme" in the right column of the middle table.
- Drop down the format menu on the property inspector and choose Heading2.

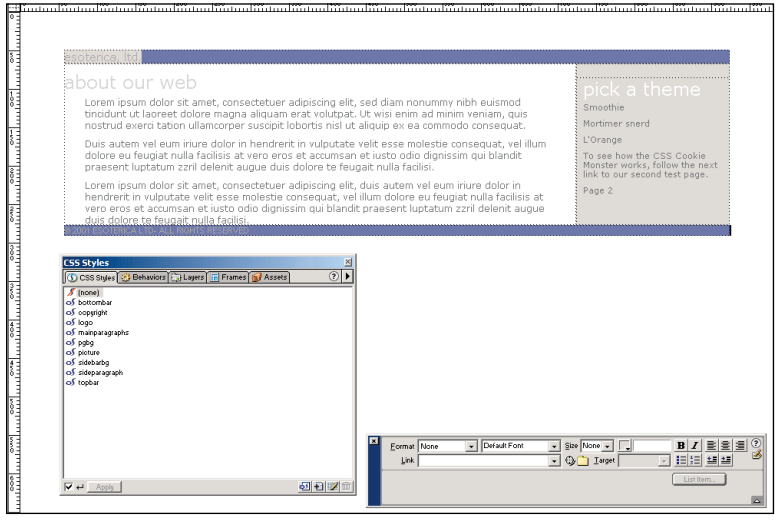
  The text will seem to disappear. We'll fix that in a moment.
- Place your cursor inside the first paragraph below the "pick a theme" heading.
- Right-click (Ctrl+click) the **<p>** tag on the status bar.
- Choose Set Class/sideparagraphs.
- Repeat this class for the remaining paragraphs in the column.
- Click inside any of the sidebar paragraphs we just styled.
- Right-click (Ctrl+click) the **<td>** tag on the status bar.
- Choose Set Class/sidebarbg.



With paragraphs, headings, and the sidebar styled, our page is really taking shape.

6 Assign classes to the lower table:

- Select the text © 2001 ESOTERICA LTD- ALL RIGHTS RESERVED.
- Right-click (Ctrl+click) and choose Wrap Tag.
- Click inside the tag markers and type **span** so that it looks like ****. Press Enter to close the box.
- Insert your cursor somewhere inside the text © 2001 ESOTERICA LTD- ALL RIGHTS RESERVED.



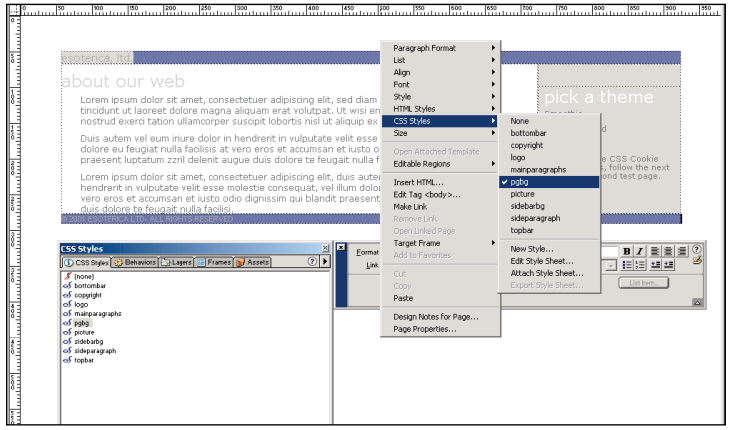
Wrap a **** tag around the text to style the copyright notice and then set the copyright.

- Select the **** tag on the status bar and right-click (Ctrl+click).
- Choose Set Class/copyright.

7 Assign a class to the lower table's row:

- Insert your cursor in the bottom table row and right-click (Ctrl+click) the **<td>** tag on the status bar.
- Choose Set Class/bottombar.

  The table row is now colored blue.



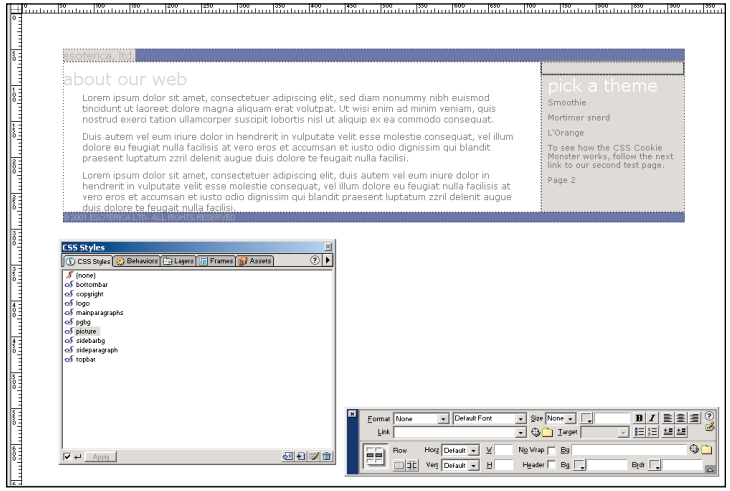
Right-click (Ctrl+click) anywhere on the page below the bottom table to set the body class.

Setting Up a Swap Image with CSS

Now the real magic starts! Remember the little nested table we placed at the top of the sidebar? We have a special CSS class for that baby.

1 Assign a class to the nested table's row:

- Click inside the nested table that is inside the sidebar.
- Right-click (Ctrl+click) the **<td>** tag on the status bar.
- Choose Set Class/picture.

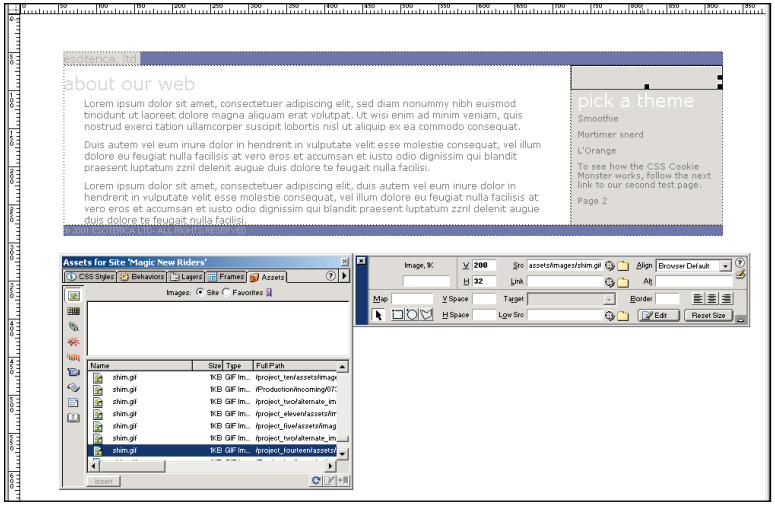


Set the **<td>** of the nested table to the picture class.

2 Insert a transparent GIF inside the nested sidebar table:

- Click inside the nested table.
- Open the Assets panel and scroll down to the image called shim.gif.
- Click the Insert button on the Assets panel to insert the image inside the table.

  The image we've provided is already properly sized at 200 pixels wide by 32 pixels high.



Insert shim.gif inside the nested table.

Note: The transparent GIF serves to hold the table open to allow a background image to show through. The CSS class .picture assigns the background image to the table cell. Each of our three CSS files sets a different image, so when a user chooses a theme, the image changes along with the style sheet. However, if you preview the page at this stage, you will not see the image. Don't worry; this is perfectly normal. We'll fix that in a moment.

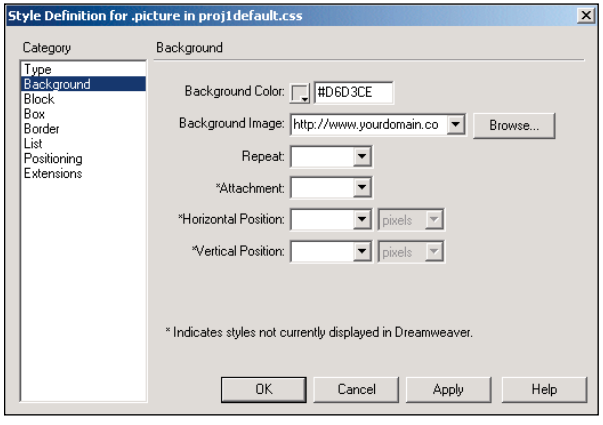
Setting the Background Image URL in Style Sheets

We'd love for this part to be just a little easier, but we are committed to supporting Netscape 4, and that browser needs a little workaround to render images declared in externally linked style sheets.

1 Enter an absolute path for our CSS background images:

- Click the Pencil icon at the bottom of the CSS Styles panel to open the Edit Style Sheet dialog.
- Double-click proj1default.css to edit the CSS file.
- Double-click the .picture class.
- Choose the Background category.
- In the Background Image field, type the fully qualified path to the default background image relative to your actual Web server, using this format as an example:

  http://www.yourdomain.com/assets/images/bluesmoothie.gif



Enter the absolute URL to the background image of our table cell.

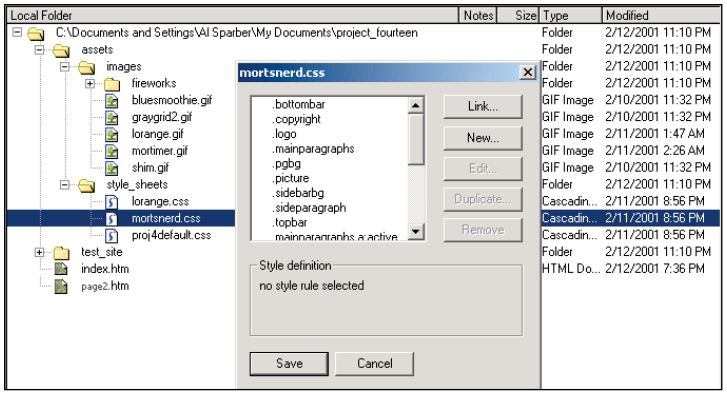
2 Enter an absolute path for the Mortimer Snerd background image:

- Open the site window and double-click mortsnerd.css
- Double-click .picture and repeat the preceding steps to enter the absolute URL for mortsnerd.css, using this path:

http://www.yourdomain.com/assets/images/mortimer.gif

3 Open the l'orange.css file and enter an absolute path for the L'Orange background image:

http://www.yourdomain.com/assets/images/lorange.gif



Double-click mortsnerd.css in the site window to open the CSS Editor.

Tip: If you are unsure of your actual domain, ask your network administrator, ISP, or hosting company to give you the proper path.

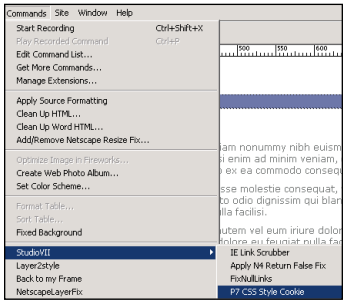
Note: By default, Dreamweaver opens the CSS dialog unless you have customized your settings in the Preferences window.

Applying the Cookie Script Command and Behaviors

We're now ready to insert the script that writes the CSS cookie. Once again, you'll be setting the absolute URL to the CSS files. This is vitally important for Netscape 4 compatibility (and also to enable you to use a Dreamweaver template to manage a multiple-folder site).

1 Apply the P7 CSS style cookie:

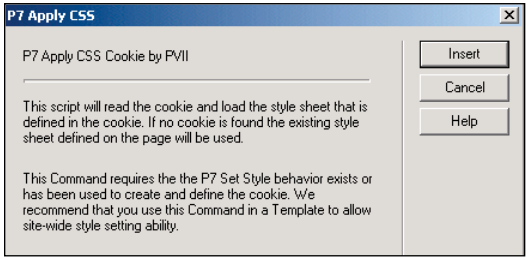
- From the Commands menu, choose Studio VII/P7 CSS Style Cookie.



Choose Commands/Studio VII/P7 CSS Style Cookie.

- Click the Insert button.

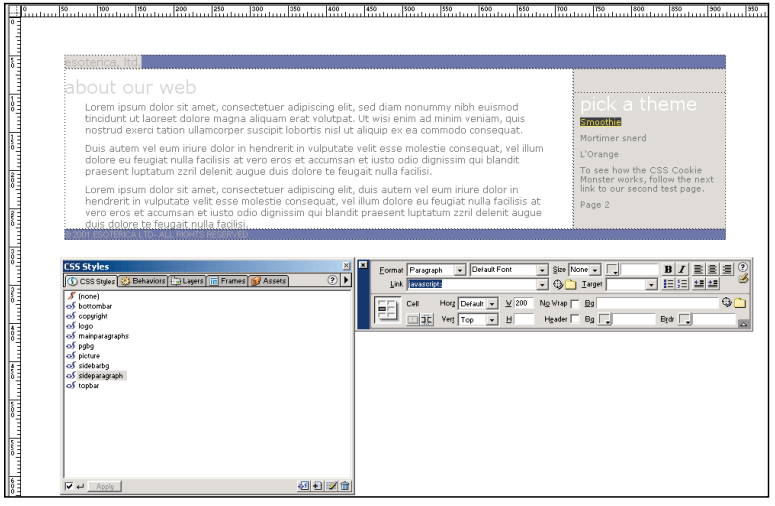
  The cookie script is now in the document.



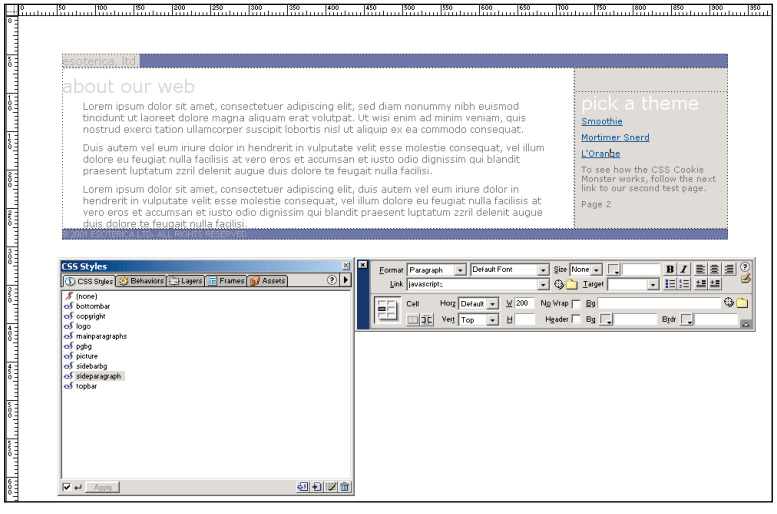
Click the Insert button to apply the P7 CSS Cookie script.

2 Prepare the text links to accept the P7 Apply CSS behaviors:

- Select Smoothie from the sidebar. Type **javascript:;** into the Link field on the property inspector.
- Repeat the preceding steps to apply JavaScript links to Mortimer Snerd and L'Orange.



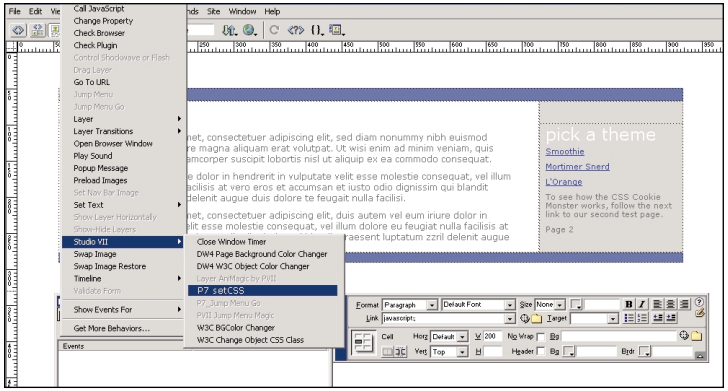
Select the word "Smoothie" in the sidebar and type **javascript:;** into the Link field on the PI.



Add null links to the remaining two theme names in the sidebar.

3 Apply the P7 CSS behaviors:

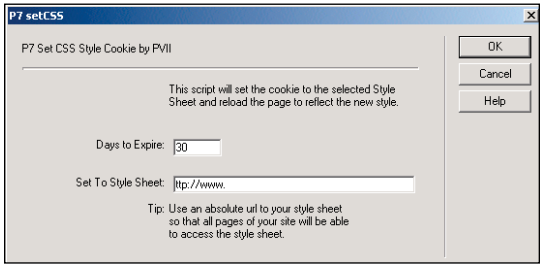
- Select the Smoothie link.
- Open the Behaviors panel menu, click the + sign to open the Behaviors menu, choose Studio VII, and then choose P7 setCSS.



Select the Smoothie link and apply the P7 setCSS behavior.

- Enter an expiration date into the Days to Expire field or keep the default setting of 30 (days).
- Type the fully qualified absolute URL to the proj1default.css style sheet in the Set To Style Sheet field. For example:

  http://www.yourdomain.com/assets/style_sheets/proj1default.css



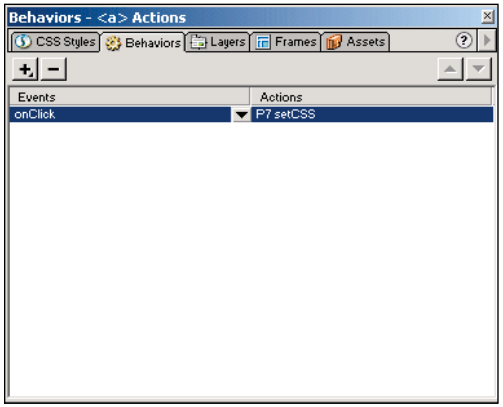
Enter an expiration date for your cookie and the absolute URL to the default style sheet.

- Make sure that the event is set to onClick.
- Repeat the preceding steps to apply the P7 setCSS behavior to the Mortimer Snerd and L'Orange links. Remember to set the absolute URLs:

  http://www.yourdomain.com/assets/style_sheets/mortsnerd.css

  and

  http://www.yourdomain.com/assets/style_sheets/lorange.css



Make sure that the event is set to onClick.

Applying the NN4 Return False Fix and the Scrubber Commands

The NN4 Return False Fix must be applied for our project to function correctly in Netscape. The Scrubber can be optionally used to eliminate the focus lines around clicked links in MS Internet Explorer. Apply these commands to both the index.htm and page2.htm pages.

1 Apply the NN4 Return False Fix:

- Choose Command/Studio VII/Apply N4 Return False Fix.
- Click the Apply button.

2 Apply the Scrubber:

- Choose Command/Studio VII/IE Link Scrubber.
- Click the Scrub Em! button.

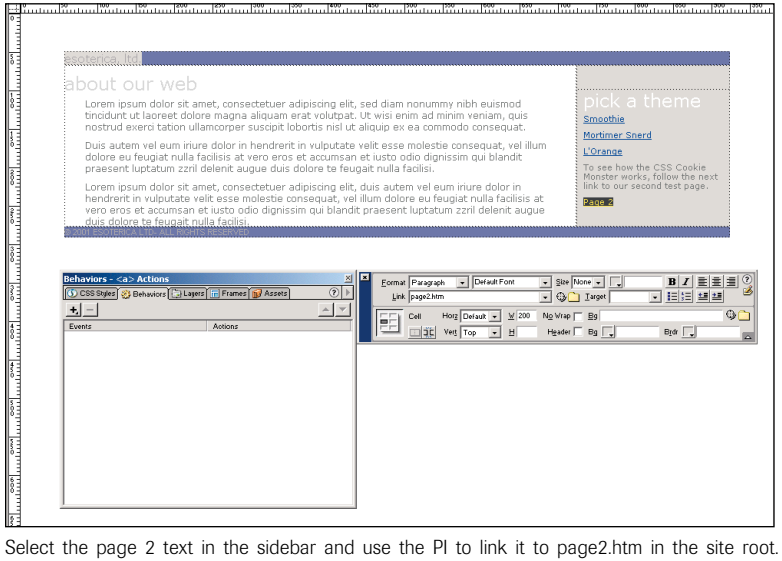
Using a Second Test Page to See if the Browser Is Reading the Cookie

It's a good idea to create a second page to test the scripts on your local system. The page should have the P7 Apply CSS command and it should be styled using the CSS classes from our style sheets. We've taken the liberty of including this page in the root of your project folder. Test the cookie, using the provided second page as described in the next section.

1 Select the text page 2 in the sidebar of the home page.

- Use the property inspector to create a normal hyperlink to page2.htm.
- Open page2.htm, select the text back to the index in the sidebar.
- Use the property inspector to create a hyperlink back to index.htm.

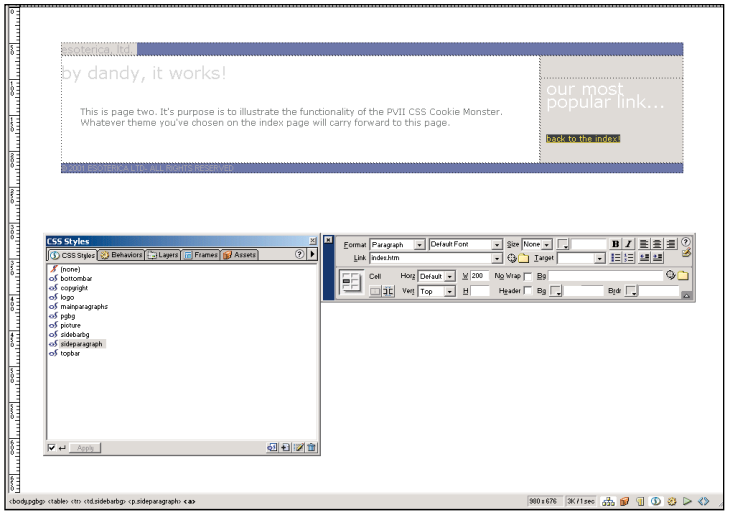
Note: Please read this paragraph. Do not overlook it. It is so important that we set it up as a separate section.



Select the page 2 text in the sidebar and use the PI to link it to page2.htm in the site root.

2 Save your work and close index.htm and page2.htm.

Now you're ready to begin working on the pages in the test site folder.



Open page2.htm, select the text "back to the index," and use the PI to link it to index.htm.

Using the Supplied Test Site

Because we need to use absolute URLs to link our style sheets, local previewing and style sheet edits would be difficult without a dedicated test page or two. The test page functions locally to permit previewing through Dreamweaver. We've created two test pages for you and placed them in a folder called test_site.

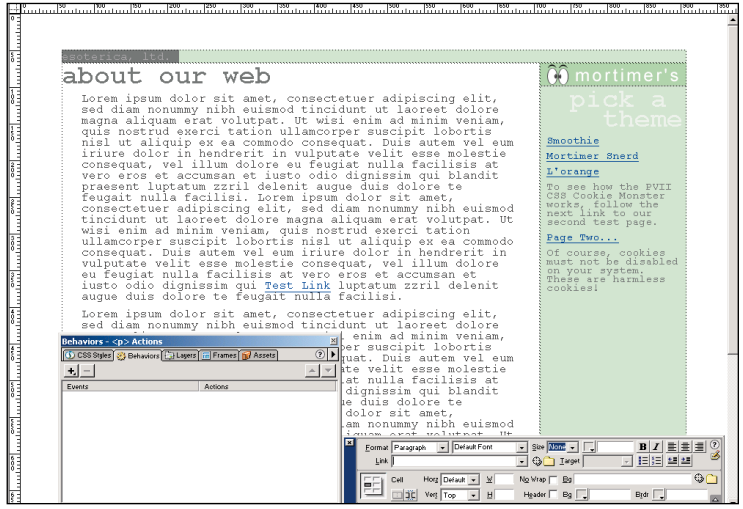
We've linked the test page to all three style sheets. **Do not link the second and third style sheets to your actual site pages, only to the test pages!**

By linking all three CSS files in the test pages, we made it possible for you to toggle the active style sheet so that you can edit the styles *and* see your changes in Dreamweaver.

The mortsnerd.css is being used as the active sheet because we've placed it below the links to the other style sheets in the document's head. The word "cascading" in Cascading Style Sheets means (among other things) that you can link multiple style sheets in a document and the browser places precedence on the one listed last.

Note: When a page contains JavaScript, Dreamweaver places a newly linked style sheet below the closing **<script>** tag, just above the opening **<body>** tag. This is incorrect and prevents the cookie script from executing properly. This is not a problem if you link a style sheet before adding scripts to the page. If you encounter this problem, simply cut the style sheet link and paste it just above the first **<script>** tag at the top of the document.

Note: This section is all about managing the finished site by using test pages that remain on your local system and do not get uploaded to your actual Web server.



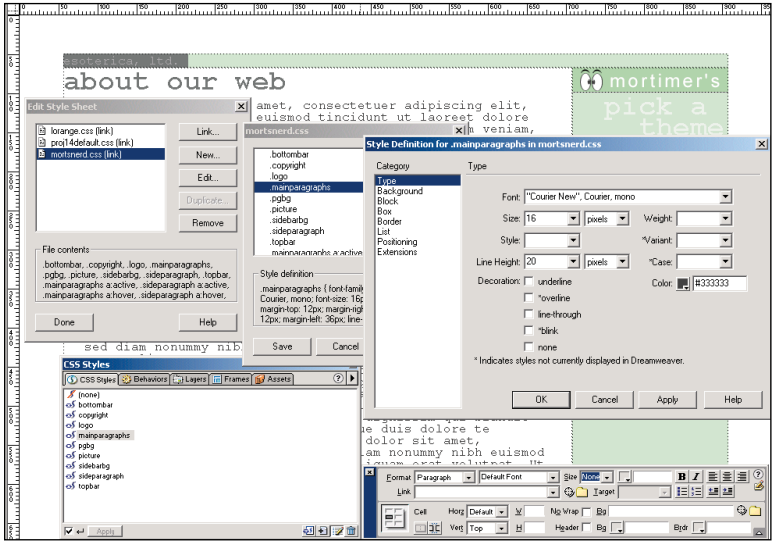
The index page in the test_site folder is set to show the mortsnerd styles by default.

We've set the source of the image atop the right sidebar to mortimer.gif by simply changing the source on the property inspector so you can visualize it onscreen. If you wanted to make L'Orange the active style sheet for your test page, you'd change the image source to lorange.gif to get an accurate preview.

```
<html>
<head>
<title>Chapter 14 - Test Site</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href="../assets/style_sheets/lorange.css" type="text/css">
<link rel="stylesheet" href="../assets/style_sheets/proj14default.css" type="text/css">
<link rel="stylesheet" href="../assets/style_sheets/mortsnerd.css" type="text/css">
<script language="JavaScript">
<!--
function P7_getCSS(a) { // v1.6 PVII
  var i=0,j=0,e=0,theSheet="";
  a=a+"=",t="<",u="LINK";
  var alen=a.length,clen=document.cookie.length;
  while (i < clen) {
    j=i+alen;
    if(document.cookie.substring(i,j)==a) {
      e = document.cookie.indexOf(";",j);
      if (e==-1) (e=clen)
      theSheet = unescape(document.cookie.substring(j,e));
      document.write(t+u+' rel="stylesheet" type="text/css" href="' + theSheet + '">');
      break;}
    else {i=document.cookie.indexOf(" ",i)+1;
      if (i==0) break; } }
}
P7_getCSS('p7styler');

function P7_setCSS(d, theStyle) { // v1.6 PVII
  var expDays=parseInt(d),path = "/";
  var expdate = new Date();
  expdate.setTime(expdate.getTime() + (expDays*24*60*60*1000));
  document.cookie = "p7styler=" + escape (theStyle) + "; expires=" + expdate.toGMTString() +'; path=' + path;
  top.location.reload();
}
//-->
</script>
</head>
<body bgcolor="#FFFFFF" text="#000000" class="pgbg">
<p> </p>
```

The style sheet links must be above the opening **<script>** tag in your document's **<head>**.



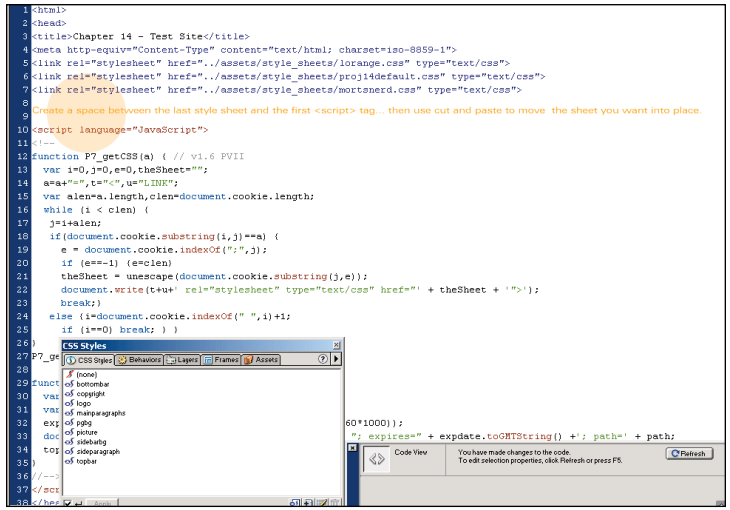
Use Dreamweaver's CSS Editor to customize the style sheets.

If you want to edit a style sheet in the test site, there are some things you need to know. The index.htm page in the test site is linked to the same style sheets used by the real index page. As mentioned earlier, it is actually linked to all three style sheets. This is the most efficient approach to take in a test scenario because if you used Dreamweaver to re-link multiple style sheets, it would place the CSS link under the JavaScript, and the script would not work.

Editing the CSS in the test pages is fine because you are using the same CSS files linked to the real site. The following is the process for editing:

1. Click the Pencil icon on the CSS panel to open the Edit Style Sheet dialog, which lists all three linked sheets.
2. Double-click the sheet you want to edit.
3. Double-click a class to open Dreamweaver's Style Definition window.

   The style sheet listed last in the Edit Style Sheet dialog is the active style sheet. Editing that sheet enables you to see your changes immediately.



The style sheet listed last in the Edit Style Sheet dialog is the active style sheet. Changing the order of your linked style sheets sets their precedence.

Tip: To make a style sheet active, go into the source code and move it to the bottom of the list.

An In-Depth Look at Project One's CSS Deployment

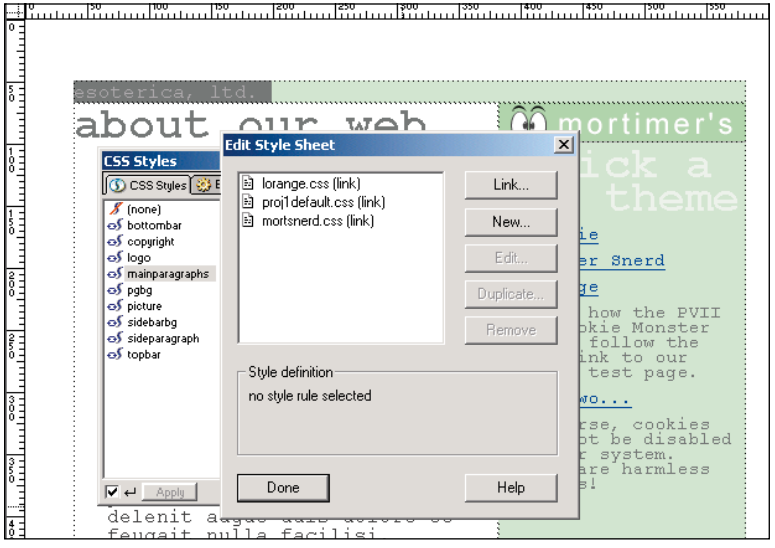
We know that our cookie script sets one of three style sheets as the active one. To make this work properly, we assigned classes to our page elements. We also redefined some tags so that any element within a redefined tag is automatically styled. We created and edited the styles in Dreamweaver's CSS Editor. If you want to follow our tracks and then play around with changing our styles, complete the steps in this section.

1 Access the CSS Editor:

- Open index.htm in the test_site folder.
- Open the CSS Styles Panel and click the little Pencil icon along its bottom border to open Edit Style Sheet dialog and see the list of linked CSS files.

Click the CSS panel's Pencil icon to open the Edit Style Sheet dialog.

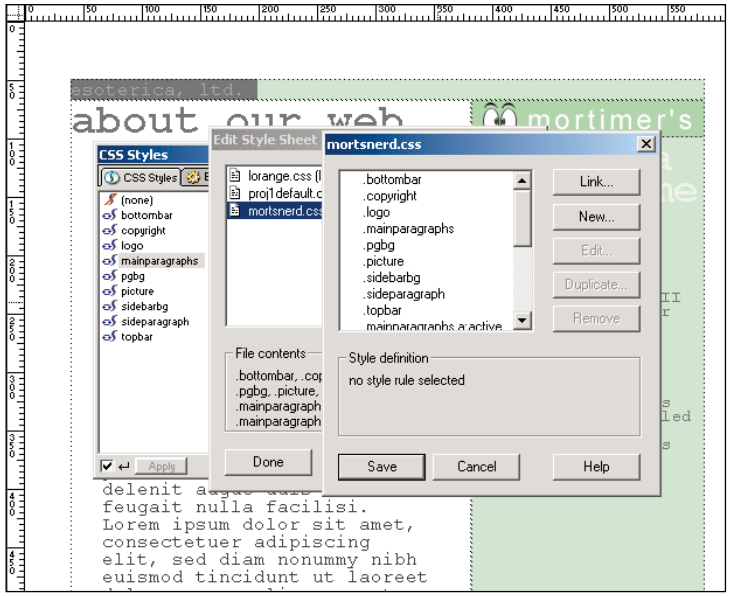
- Double-click the active (bottom) style sheet file mortsnerd.css to open a window that lists the styles in the mortsnerd.css file.



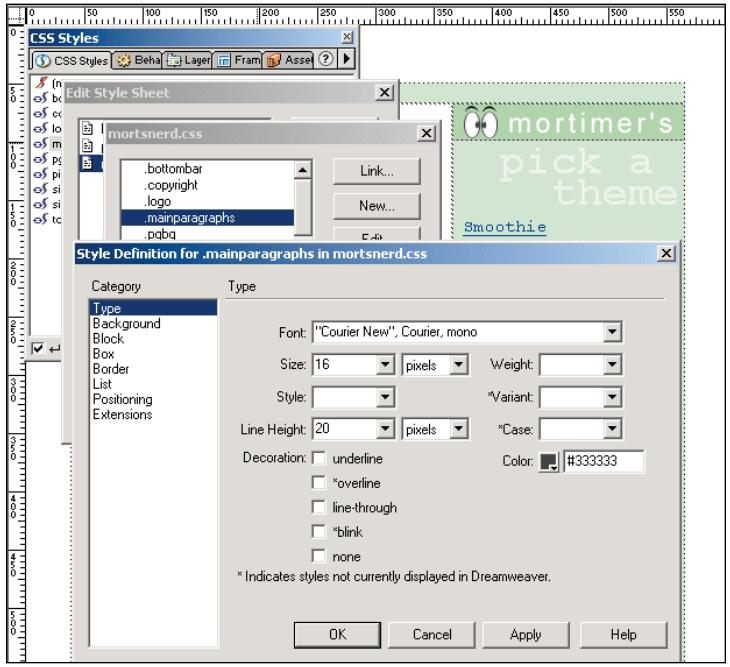
Double-click the active (bottom) style sheet to open the styles list.

- Double-click any style to open the CSS Editor window.

  The CSS Editor title bar displays Style Definition for .whateverstyleyouchoose in mortsnerd.css. The Editor window displays these categories: Type, Background, Box, Border, List, Positioning, and Extensions.



Double-click any style to open the Editor window.



In the Editor window, select the category you want to work with.

2 Look at the styles one-by-one:

- Double-click the style called h1.

  h1 { font-family: "Courier New", Courier, mono; font-size: 36px; font-weight: bold; color: #666666; line-height: 36px; margin-top: 6px; margin-right: 0px; margin-bottom: 12px; margin-left: 0px}

  This is a redefinition of the h1 (Heading 1) tag. Font-family tells the browser to use Courier New. If Courier New is not present on the client system, the choice cascades to Courier. If Courier is not present, the browser looks for Mono (the default mono-spaced font on the system). If Mono isn't found, the browser chooses the client system's default monospaced font. Font size, font weight, color, and line-height are all selected in the Type Category of the Editor window. Margin settings are made in the Box category.

- Double-click .topbar.

  .topbar { background-color: #C6E3C6}

  .topbar is a custom style (denoted by a leading period). Custom styles are always applied manually, via the CSS panel or by right-clicking (Ctrl+click-ing) the tag on the status bar. Both the .topbar and .bottombar styles are applied to the **<td>** tags in the top and bottom table rows of our page. We have made two styles to give us the option of making each cell a different color.

- Double-click .bottombar.

  .bottombar { background-color: #C7E2C7}

  Background colors and images are entered in the Editor's Background category.

- Double-click .mainparagraphs.

  .mainparagraphs { font-family: "Courier New", Courier, mono; font-size: 16px; color: #333333; margin-top: 12px; margin-right: 36px; margin-bottom: 12px; margin-left: 36px; line-height: 20px}

  The custom style .mainparagraphs is used to style the paragraphs in the main content column of our page.

- Double-click .sideparagraph.

  .sideparagraph { font-family: "Courier New", Courier, mono; font-size: 14px; color: #666666; margin-top: 12px; margin-right: 12px; margin-bottom: 12px; margin-left: 12px; line-height: 16px}

  The custom style .sideparagraph is used to style the paragraphs in the sidebar column of our page.

- Double-click .sidebarbg.

  .sidebarbg { background-color: #C7E2C7}

  The custom style .sidebarbg is applied to the **<td>** tag of the sidebar column.

- Double-click .h2.

  h2 { font-family: "Courier New", Courier, mono; font-size: 36px; font-weight: bold; color: #E2F1E2; line-height: 36px; margin-top: 6px; margin-right: 6px; margin-bottom: 0px; margin-left: 12px; text-align: right; text-indent: 0px}

  The redefined h2 tag has a text-indent, which is applied in the Block category. It is set to 0 because other style sheets do have an indent set. If a particular property is used in any of the three style sheets, it must be present in all three, even if it is 0. Why? Because if not explicitly stated, the browser continues to use the last declared value for that property until we tell it otherwise.

- Double-click .logo.

  .logo { font-family: "Courier New", Courier, mono; font-size: 16px; color: #999999; background-color: #666666; letter-spacing: 2px}

  This custom style is the one we applied to the **** tags surrounding esoterica, ltd. in the top bar of our table.

- Double-click .copyright.

  .copyright { font-family: "Courier New", Courier, mono; font-size: 10px; color: #333333; text-indent: 3pt}

  This custom style is the one we applied to the **** tags surrounding the copyright notice in the bottom bar of our table.

- Double-click .sideparagraph a:link.

  .sideparagraph a:link { color: #0000CE; text-decoration: underline}

  This is a contextual selector. Huh? Any link inside an element that has the .sideparagraph custom style applied to it will automatically render according to this style. In other words, the browser selects and accordingly styles all links (**<a>** tags) in the context of elements that have a class of .sideparagraph.

- Double-click .sideparagraph a:visited.

  .sideparagraph a:visited { color: #993399; text-decoration: underline}

  This is the style selected for all visited links according to the contextual selector.

- Double-click .sideparagraph a:hover.

  .sideparagraph a:hover { color: #0000CE; text-decoration: underline; background-color: #C6E3C6}

  This is the style selected while hovering the mouse over the links styled with the .sideparagraph selector. Hover is a CSS-2 property and is currently supported by MSIE4, 5, and 6 series browsers, Netscape 6, and Opera 5. We don't want the Mortimer theme to have a changed background on hover, but because the other themes have this effect, we need to declare a background color; otherwise, the browser would render the background from the last selected theme. So, in this case, we set the background color to the same color as the table cell background in .sidebarbg.

  However, if you compare the two, you'll note that the color numbers are slightly different. This seems to be a bug (at least on our three test systems) in Dreamweaver's color picker tool. The colors are similar and render fine in the browser. Feel free to change them, but we wanted to document the actual results of our page-building process!

- Double-click .sideparagraph a:active.

  .sideparagraph a:active { color: #333333; text-decoration: underline}

  This is the style selected while the mouse is being pressed down on .sideparagraph links.

- Double-click .body.

  body { background-color: #FFFFFF}

  This is the redefined body tag. It simply sets the background color.

- Double-click .mainparagraphs a:link.

  .mainparagraphs a:link { color: #0000CE; text-decoration: underline}

  This is a contextual selector for all links within paragraphs to which the .mainparagraphs custom style is applied. So, we can have more than one style of link on our pages. Cool!

- Double-click .mainparagraphs a:visited.

  .mainparagraphs a:visited { color: #996699; text-decoration: underline}

  This is a contextual selector for all visited links within paragraphs to which the .mainparagraphs custom style is applied.

- Double-click .mainparagraphs a:hover.

  .mainparagraphs a:hover { color: #0000CE; text-decoration: underline; background-color: #FFFFFF}

  This is the style selected while hovering the mouse over the links styled with the .mainparagraphs selector.

- Double-click .mainparagraphs a:active.

  .mainparagraphs a:active { color: #333333; text-decoration: underline}

  This is the style selected while the mouse is being pressed down on .mainparagraph links.

- Double-click .picture.

  .picture { background-color: #99CC99; background-image: url(http://www.yourdomain.com/assets/images/mortimer.gif)}

  Shssh… keep this one to yourself. It's really a cool trick. This is the custom style that's applied to the **<td>** tag in the nested table atop the sidebar. When a visitor chooses a theme, a different background image is loaded because each of our three style sheets has different images declared.

Note: The image path must be absolute to accommodate a Netscape 4 bug.

3 Edit a style:

It's real easy to manage and edit the style sheets using the test_site index page technique. We already linked all three style sheets to the test page, and you've learned how to manipulate the order of a sheet (in the source code) to make it the active sheet. So, you're ready to edit a theme—if you want.

- Make the theme style sheet the active sheet.
- Use Dreamweaver's CSS Editor to make changes.

  As you make a change, Dreamweaver renders the style on the page.

Automatically Applying a Link Style to All Links on a Page

If you want to have a link style that automatically applies to all links on your page (except those to which a contextual selector applies), then you would define them using the CSS selector in the New Style dialog:

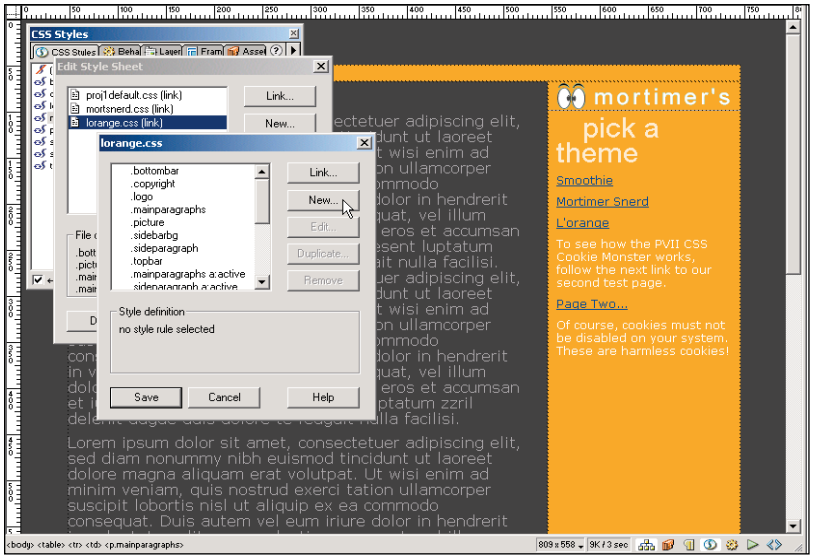
- a:link { blah; blah; blah}
- a:visited { blah; blah; blah}
- a:hover { blah; blah; blah}
- a:active { blah; blah; blah}

Note: Dreamweaver renders most styles in the document window. The styles it is incapable of rendering are marked with asterisks in the CSS Editor. However, those styles are rendered when you preview in your browser (if the browser supports them!).

4 Add a new style:

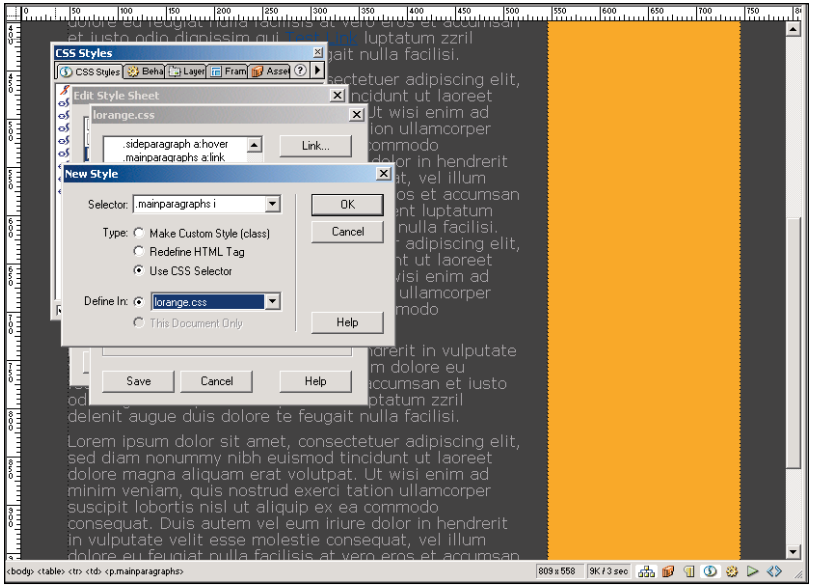
If you want to add a new style, there are some steps you should take to make sure you get the results you want. Let's say that you want a certain paragraph in the main body to stand out.

- Open the CSS panel.
- Click the Pencil icon.
- Double-click the active (bottom) style sheet. In this case, I've made the active sheet lorange.css because I'm tired of looking at green screen captures!
- Click the New button.



Open the CSS panel, click the Pencil icon, double-click lorange.css, and then click New.

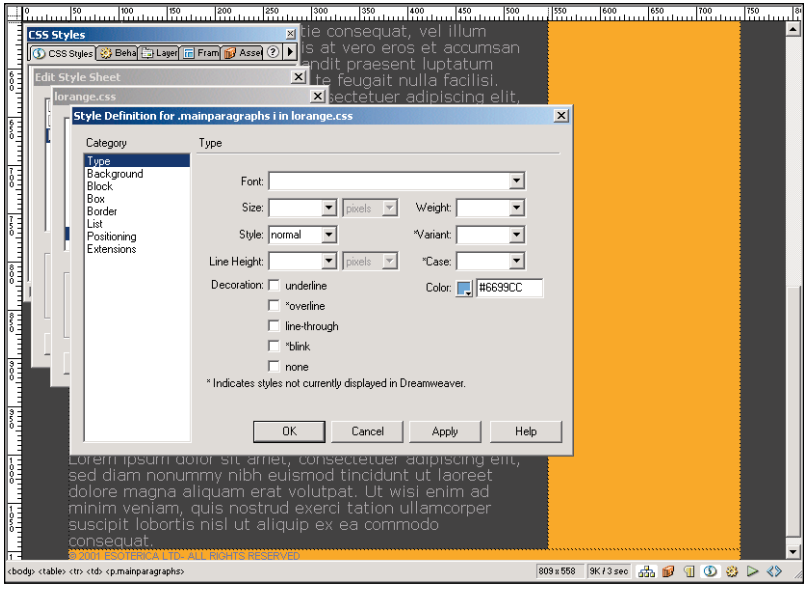
- Make sure that Type is Use CSS Selector and Define In is lorange.css.
- Type **.mainparagraph i** into the Name field.
- Click OK and the Editor window opens with the Type category selected.



Name the new style and make sure that it is defined in lorange.css.

- Enter **#6699CC** in the Color field and normal in the Style field.
- Leave all other fields blank.

Tip: To create a new style, you can simply click the + icon on the CSS panel's bottom (to the left of the Pencil icon). This opens the New Style dialog. We took a purposefully circuitous route to make sure that you are familiar with all the CSS dialogs.



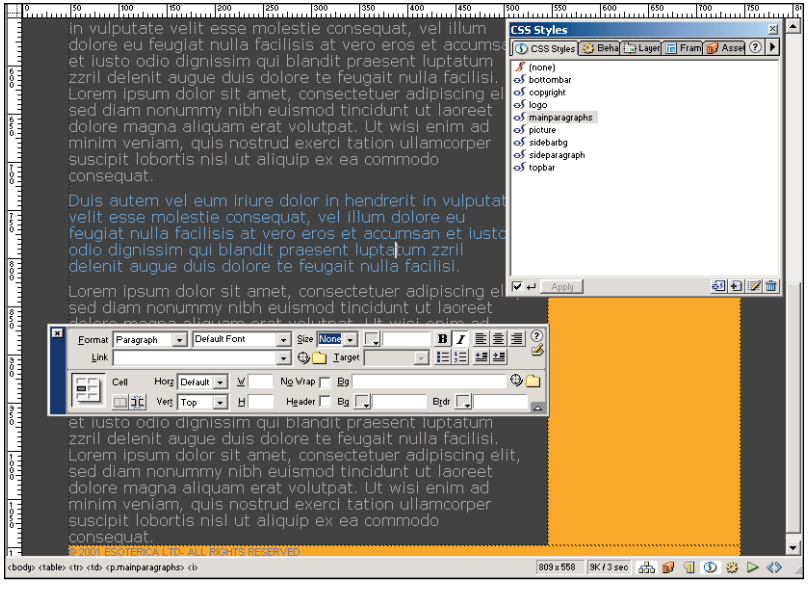
Set the style and color.

5 Apply the new style:

We have a special new style. A Contextual Selector just like our special link styles. But this Selector does not do links. It styles text with an assigned class of .mainparagraph that has also been set to italic (**<i>**).

Tip: I use this technique often because I never use italic text. It tends to look pretty ugly on the Web. So why waste a perfectly good tag!

- Select any block of text within a paragragh that has the .mainparagraph class applied.
- Click the Italic button on the PI.



Apply the style by selecting a block of text in the main table column and then clicking the Italic button in the PI.

Note: This style is not included in the supplied style sheets. But do feel free to make your own selectors. If you do, make sure that you make identically named selectors in your other style sheets.

Modifications

The nature of this project leaves room for easy and efficient modifications. You can change an existing theme in just a few minutes. You can add additional themes just as easily by adding new style sheets.

The easiest way I've found to add more themes is to open one of the style sheets in my default text editor (Notepad, for example). Use the text editor's File menu to save the .css file with a new name, thereby creating a duplicate.

Link the new style sheet to your test page. Make sure that you go into Code view and move it above the opening script tag (this is explained in greater detail earlier in the project).

Edit the style sheet in the Dreamweaver CSS Editor, from within the test page; and then create a new link beneath the last theme (L'Orange) and apply the P7 setCSS behavior. Set this instance of the behavior to "call" your new style sheet. It's that simple!

Test it and then place the link and behavior on your real index page, using the techniques you've learned in this project!

Don't forget! The CSS links in your test site can be document relative, but the links from your real site must be absolute (http://don't_forget!).

And always use the test page technique to edit and preview your themes locally! If you try to preview your real pages, your browser will attempt to connect with your remote server via the Internet. So it's usually best to use the test page.