PHP and CSS: random images and CSS positioning

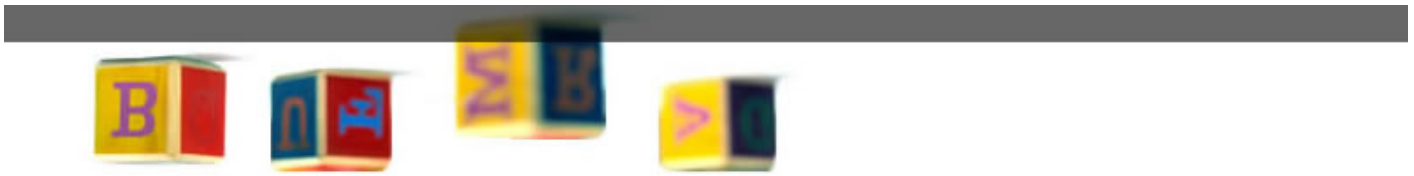
Until recently I was unaware of the power of CSS and Its ability to integrate with other languages. I always thought that styles were static and had to be coded into the style sheet.

It was with a visit to A List Apart when I stumbled across a PHP article about Random Images by Dan Benjamin. As I'm only a newcomer to PHP, I was interested enough to read on and found it to be a great article & script. Another blog style site Relatively Absolute is a perfect working example of the Image rotation script as it uses one for generating random header images much like the ones I'm going to be demonstrating here. In fact Absolutely Relative inspired me to write this article because I liked the transparency illusion so much.

The header that inspired me from Relatively Absolute

I had 2 issues I had to answer.
1. How do I assign a transparent image if IE doesn't support it, and.
2. How do I get my random images into a background-image declaration in CSS.

Answering issue #1 was quite simple. I actually create the black faded bar with my image, hence me calling it an illusion as there is no image transparency. You're simply creating a box in the top layer of your graphic and lowering its Alpha value until it becomes see-thru as you can see below.



One of the example images complete with faded bar.

All the .PNG files for the headers are supplied so you can see exactly how I've done this.

I'll answer issue# 2 when we start looking at the CSS needed for the display.

Our Imaginary Site

For this example we're going to need some images to rotate, the rotate script, a style sheet, and a page to view the images on.

Creating the Images

Let's start by creating our images. I've made 5 header graphics **800x100** in Fireworks and named them **header1.jpg** through to **header5.jpg**
I've saved these to a folder on my server called **headers**

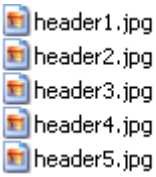


Image 1.1: Listing of header images in folder

The header images are for example purposes only.

Examining the Random Image PHP Script Options

In this example I'm going to be using the **Basic Random Image** script by Dan Benjamin. There is also another excellent script available on A List Apart called **A Better Image Rotator** also by Dan Benjamin, but for this article I will be talking about the **Basic Version** of the script.
For editing purposes I have removed most of the comments, and I will just show the actual PHP for configuration along with chosen comments that can add more functionality.

```
<?php

/* ------------------------- CONFIGURATION ----------------------


      Set $folder to the full path to the location of your images.
      For example: $folder = '/user/me/example.com/images/';
      If the rotate.php file will be in the same folder as your
      images then you should leave it set to $folder = '.';

*/
$folder = '.';

/*

      If you'd like to enable additional image types other than
      gif, jpg, and png, add a duplicate line to the section below
      for the new image type.
      Add the new file-type, single-quoted, inside brackets.
      Add the mime-type to be sent to the browser, also single-quoted,
      after the equal sign.

      Example PDF Files entry:

            $extList['pdf'] = 'application/pdf';

      Example CSS Files entry:

            $extList['css'] = 'text/css';
```

```
      You can even serve up random HTML files:

            $extList['html'] = 'text/html';
            $extList['htm'] = 'text/html';

      Just be sure your mime-type definition is correct!

*/

$extList = array();
      $extList['gif'] = 'image/gif';
      $extList['jpg'] = 'image/jpeg';
      $extList['jpeg'] = 'image/jpeg';
      $extList['png'] = 'image/png';

?>
```

Main options for PHP random image script

This is NOT the full script. The full code listing for the script can be found in the example folder in the file **rotate.txt** or in the headers folder as the file **rotate.php**
Full permission was granted by the author for reprint on DMX Zone.

Breaking apart the code, we have the folder **url** for the script.

```
/* ------------------------- CONFIGURATION ----------------------


      Set $folder to the full path to the location of your images.
      For example: $folder = '/user/me/example.com/images/';
      If the rotate.php file will be in the same folder as your
      images then you should leave it set to $folder = '.';

*/
$folder = '.';
```

Folder Options for the Random Image script

Only the very last line **$folder = '.';** actually does anything. The rest you will notice is commented out with a basic instruction on how to set the folder path if the script is stored outside of the folder where the images are stored.
The path of **'.'** Means the script will call itself from that directory. It is advised that the script be left setup to this default for greater reliability and performance.

Other Options

```
/*

      If you'd like to enable additional image types other than
      gif, jpg, and png, add a duplicate line to the section below
      for the new image type.
      Add the new file-type, single-quoted, inside brackets.
      Add the mime-type to be sent to the browser, also single-quoted,
      after the equal sign.

      Example PDF Files entry:
```

```
            $extList['pdf'] = 'application/pdf';

      Example CSS Files entry:

            $extList['css'] = 'text/css';

      You can even serve up random HTML files:

            $extList['html'] = 'text/html';
            $extList['htm'] = 'text/html';

      Just be sure your mime-type definition is correct!

*/

$extList = array();
      $extList['gif'] = 'image/gif';
      $extList['jpg'] = 'image/jpeg';
      $extList['jpeg'] = 'image/jpeg';
      $extList['png'] = 'image/png';
```

Additional File Formats in the Random Image script.

Looking at the **array();** you can see we have 4 supported File types.
.gif
.jpg
.jpeg
.png

The fact that we can call random **pdf's**, **css** style sheets and **html** documents, means you can really stir things up if you want to go into random overkill. You could even take this further yourself and add any other relevant File Types you want supported. With this in mind I trim the comments so my code block now supports all files mentioned above.

```
$extList = array();
      $extList['gif'] = 'image/gif';
      $extList['jpg'] = 'image/jpeg';
      $extList['jpeg'] = 'image/jpeg';
      $extList['png'] = 'image/png';
      $extList['html'] = 'text/html';
      $extList['htm'] = 'text/html';
      $extList['css'] = 'text/css';
      $extList['pdf'] = 'application/pdf';
```

The Array() with all supported File Types

That's the PHP script set up and configured for our use. Save this as **rotate.php** in the folder where you have your images you want rotating.

The Style Sheet

For the example site, I am going to be using a **Centered 2-column** set up, with a header and a menu div overlapping the header div to form the transparent menu effect.
Setting up the style sheet for the layout is going to consist of 5 **DIVS** and a **BODY** tag. I will also be using a style sheet for styling my fonts and A tags, but I like to keep my layout and styles separate, so these will be in 2 separate files.

Now I know a lot of designers/developers use Dreamweavers built in method for writing CSS, but I much prefer to write my own. For this example, I'll supply the CSS code I have written, and leave the basics of adding a style out as I am assuming you know how to do this already.

The CSS for the layout

```
body{
margin:0px;
background-color:#fff;
}
div#wrapper{
position:absolute;
margin-left: -400px;
top: 0px;
left:50%;
width: 800px;
height: auto;
}
div#header{
height: 100px;
border-bottom: 1px solid black;
text-align:left;
background: url(../headers/rotate.php) #000 no-repeat center top;
}
div#menu{
z-index: 5;
position: absolute;
top:0px;
height:20px;
line-height: 17px;
}
div#leftpanel{
position: absolute;
top:106px;
left:5px;
width: 200px;
border: 1px solid black;
float:left;
}
div#maincontent{
position:relative;
top: 5px;
float: right;
right:5px;
width: 580px;
border: 1px solid black;
}
```

Our layout CSS code for positioning the DIV's

Breaking the CSS for the layout down we start with the **body** element.

```
body{
margin:0px;
background-color: #fff;
}
```

The Body Element in our CSS

This will remove the default **margins** set on the body, and assign a background color of **white**.

Next we have the div to hold and center the main elements.

```
div#wrapper{
position:absolute;
margin-left: -400px;
top: 0px;
left:50%;
width: 800px;
height: auto;
}
```

The Wrapper div to hold or Div's

This has a fixed width and is centered using the **negative margin** technique. You can also center using the auto margin technique but I prefer to use this method when I have fixed width elements. Because this element is fixed width, you don't need to specify a width for any other elements that you wish to be 800px as they will auto span the div they are nested in.
Next we have our Header div which will display the graphics.

```
div#header{
height: 100px;
border-bottom: 1px solid black;
background: url(../headers/rotate.php) #000 no-repeat center top;
}
```

The Header CSS

Here we set a **height** of 100px to accommodate our header graphics which if you remember were created to be 800x100px. No width is needed as a Div will automatically span across the available space. A **bottom border** has been declared just for ease of viewing, so you can see where the Header div ends. And finally we set our **background** image, color, repeat and position using the shorthand method of CSS. The line consisting of :

```
background: url(../headers/rotate.php) #000 no-repeat center top;
```

The background code in shorthand CSS

can also be written like:

```
background-image: url(../headers/rotate.php);
background-color: #000;
background-repeat: no-repeat;
background-position: center top;
```

The background code as Dreamweaver will want to write it.

Notice that our **background-image url** is pointing to a **.php** file and not an a .jpeg or .gif like normally. This is how the randomisation as a background image works. The **.php** file which we looked at earlier automatically generates the image name and file extension for us, allowing us to use it as a background element in CSS as well as in a normal **<img>** tag.
Next we set up our Menu div so it stacks above the header div. We do this using the **z-index** attribute.

```
div#menu{
z-index: 5;
position: absolute;
top:0px;
height:20px;
line-height: 17px;
}
```

The Menu Div

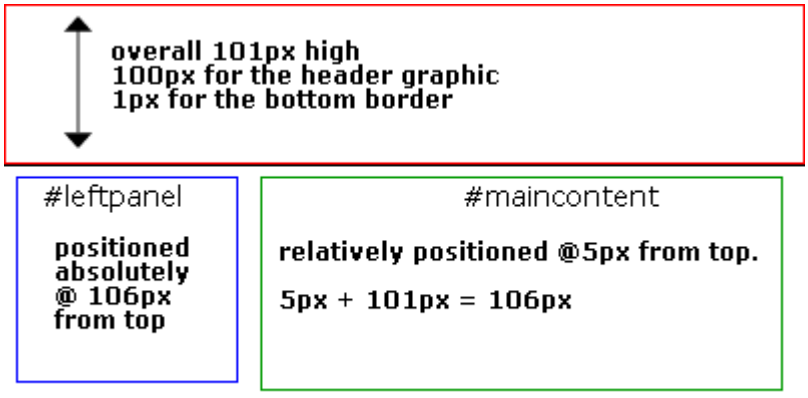
Here we set up a stacking order of 5 using **z-index**, this will keep the Menu div on top of the Header div. Absolutely positioned inside our Wrapper div we set it to **Top** 0px and give a **height** of 20px which will match our graphic's faded black bar.
Again no **width** is needed as the div will auto span across the 800px, and finally by setting a **line height** of 17px we should have any text nicely aligned vertically within the div.

The last 2 div's **#leftpanel** and **#maincontent** are provided as examples on different techniques of using positioning.

```
div#leftpanel{
position: absolute;
top:106px;
left:5px;
width: 200px;
border: 1px solid black;
float:left;
}
div#maincontent{
position:relative;
top: 5px;
float: right;
right:5px;
width: 580px;
border: 1px solid black;
}
```

2 techniques of how to position divs.

You will see that **#leftpanel** is **absolutely** positioned at **top**: 106px & **left**: 5px, and because this div is nested in a wrapper div, the absolute position relates to the 0,0 co-ordinates of the holding div and not the browser window.
#maincontent on the other hand is positioned **relatively**, and because it comes in order after the header div, it will position itself relative to that element. So our top measurement here is **top**: 5px. Which translates to 5px off the bottom of the header div. This will place the div directly in line with the **#leftpanel** at a Y co-ordinate of 106px, which you can easily figure out by looking at the sum below.



Our layout model

All we need to add now to our CSS, is some styling for the text and links. I'll store these in a separate sheet to keep my layout separate. This is good practice if you wish to accommodate older browsers, as you can choose whether to link or import the style sheet at runtime. Now lets add some styles for the **<p>**, **<a>**, and **<h1>** tags. I've saved this as **styles.css**

```
body{font-size:10px;}
p, h1, a {font-family: Verdana, Arial, Helvetica, sans-serif;}
p{
margin: 0px;
padding: 0px;
font-size:1em;
}
div#leftpanel h1{
font-size: 1.8em;
background-color:#CCC;
padding:0px;
margin: 0px;
}
div#maincontent h1{
font-size: 1.8em;
background-color:#666;
padding:0px;
margin: 0px;
}
div#menu a, a:link{
padding: 0px 10px 0px;
border-right: 2px solid #000;
color: #fff;
text-decoration: none;
font-size:0.9em;
font-weight:bold;
}
div#menu a:hover{
color: #00FF00;
}
```

Our styles for the P, H1, and A tags

I start off the styles declaring a font-size of 10px for the body of the document. This allows us to use a relative **em** sized font for our other elements. This is particularly useful for viewers who like to scale their fonts up a little for readability. Using a fixed pixel sized font means that certain browsers will not scale the font size, so all your readers will be stuck with 10px.

Recommended reading: CSS Units by Matt Machel

Calculating **em** values is quite easy. With it being a relative font-size, it takes it value from the parent element, in this case the body tag. So 10px = 1em. 11px would be 1.1em, 12px would be 1.2em and so on. I told you it was easy.

Removing the margins and padding from the **p** tag, makes it act like a shift+enter, rather than a default paragraph block with the empty line underneath, and this can also help with certain layout problems where alignment is off.

Next I add the styles for the **H1** tags in each div, giving each a different background colour so you can categorise them easier.

All we need to add now is the styles for the menu **a** tags.

```
padding: 0px 10px 0px;
```

Left & Right padding set to 10px

```
border-right: 2px solid #000;
```

2px border acts as a separator like you typed the '|' character

```
text-decoration: none;
```

Removes the underlines from the hyperlinks.

The rest of the CSS for the **a** tag is pretty self-explanatory, the only thing we change in the **a:hover** is the **color** of the text.

Our html document

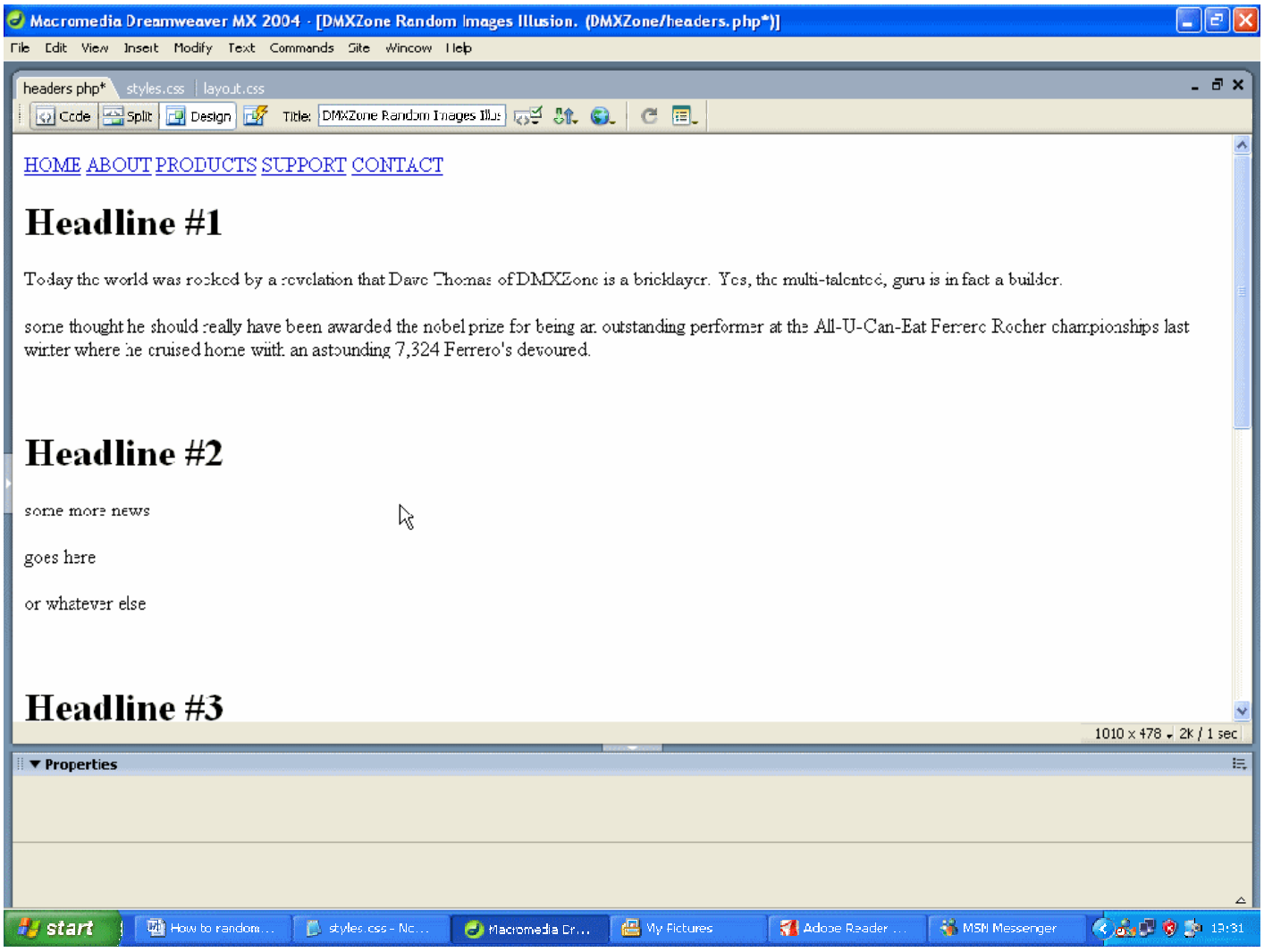
Setting up the display page is a simple matter of placing the div's in the correct order and adding some filler text. Copy and Paste the code below, this can be saved with either a **.html** or a **.php** extension. I've saved mine as **headers.php**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>DMXZone Random Images Illusion.</title>

</head>

<body>
<div id="wrapper">
<div id="menu"><a href="#">HOME</a> <a href="#">ABOUT</a> <a href="#">PRODUCTS</a> <a
href="#">SUPPORT</a> <a href="#">CONTACT</a></div>
<div id="header"></div>
<div id="maincontent">
  <h1>Headline #1 </h1>
  <p>Today the world was rocked by a revelation that Dave Thomas of DMXZone is a
bricklayer. Yes, the multi-talented, guru is in fact a builder.</p>
  <p>some thought he should really have been awarded the Nobel prize for being an
outstanding performer at the All-U-Can-Eat Ferrero Rocher championships last winter
where he cruised home with an astounding 7,324 Ferrero's devoured.</p>
  <p> </p>
  <h1> We depend on you </h1>
  <p>Please rate our articles and other content on DMXZone. It will help us to give a
more rapid and enjoyable experience if we have your feedback.</p>
  <p>  </p>
  <p>  </p>
  <p>  </p>
  <h1>Headline #3</h1>
  <p>and again in here until u get bored with headlines :)</p>
  <p> </p>
  <p>boring.......</p>
</div>
<div id="leftpanel">
  <h1>Title 1 </h1>
  <p>Some text for the title 1 story. lorem ipsum blah blah blah filler text.</p>
  <p>Some text for the title 1 story. lorem ipsum blah blah blah filler text.</p>
  <h1>Title 2</h1>
  <p>Some text for the title 2 story. lorem ipsum blah blah blah filler text. Some text
for the title 2 story. lorem ipsum blah blah blah filler text.</p>
  <p> </p>
```

```
  <p>Etc. Etc... </p>
  <p>  </p>
</div>

</div>
</body>
</html>
```

Our html code for displaying our example page.

Un-styled this will show the following in Dreamweaver.



Our un-styled html document.

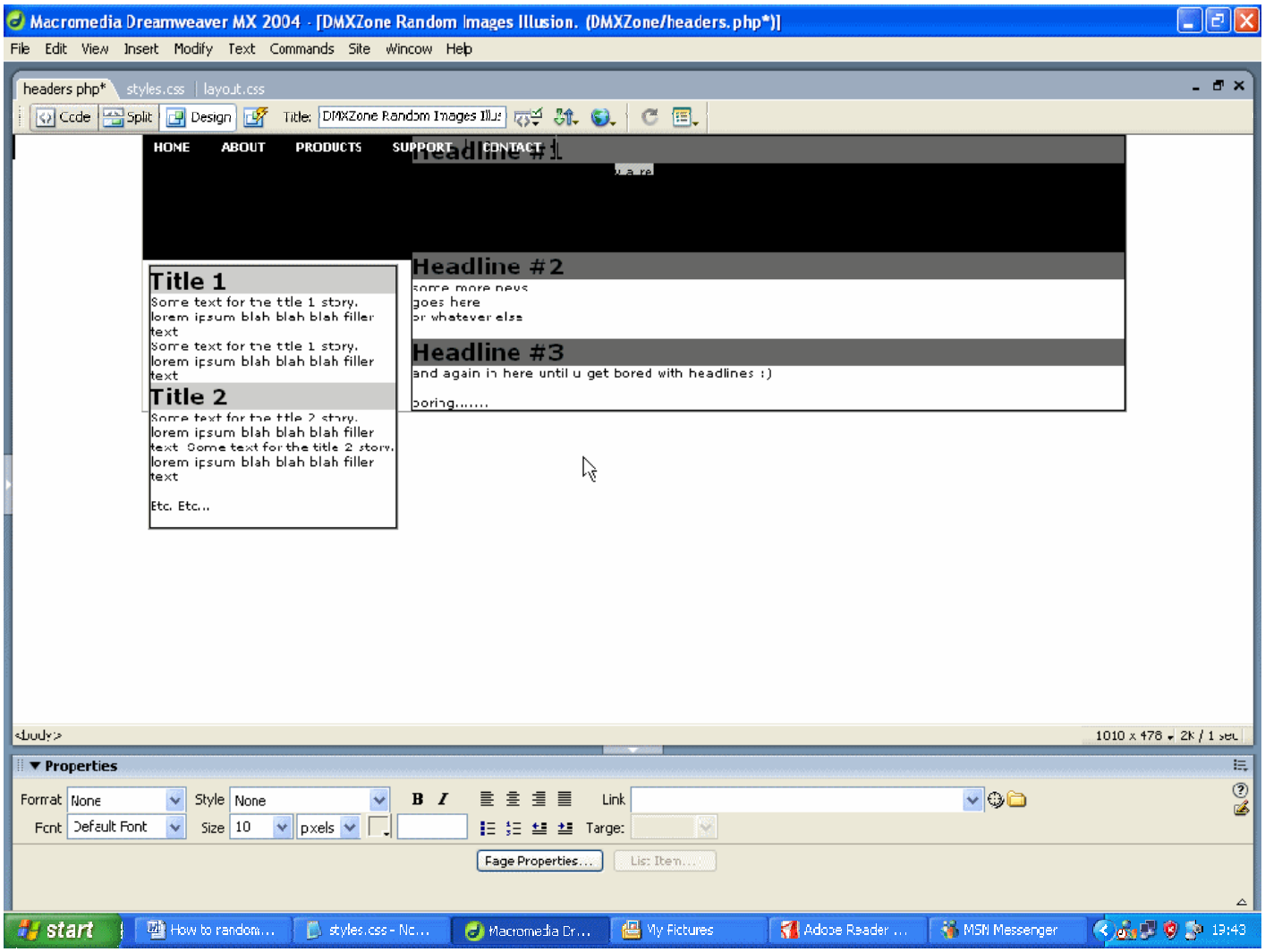
Go ahead and link the style sheets to the document and see it change.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>DMXZone Random Images Illusion.</title>

<link href="css/styles.css" rel="stylesheet" type="text/css">
<link href="css/layout.css" rel="stylesheet" type="text/css">
</head>
```

The HEAD with the 2 linked style sheets

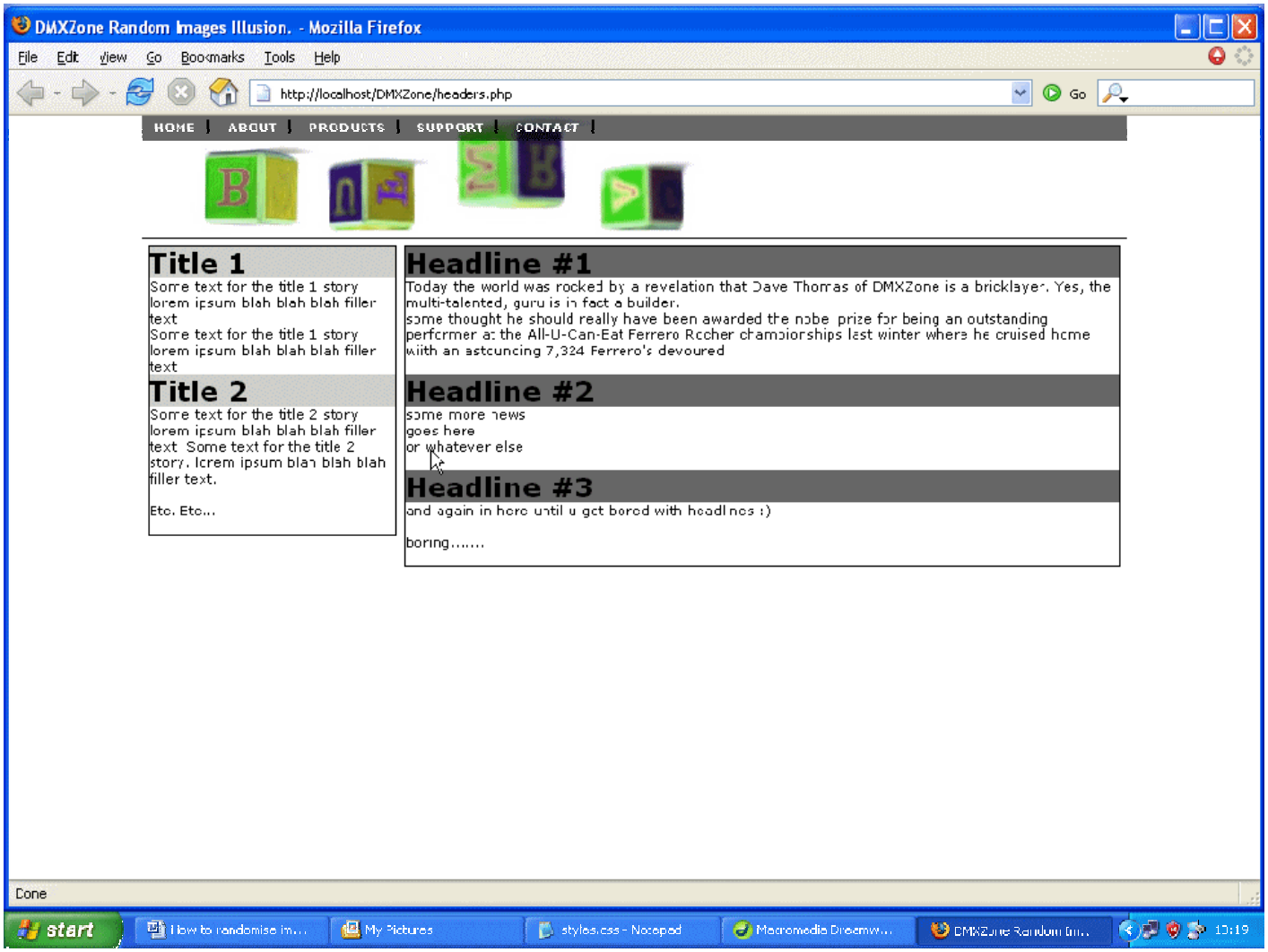
Note: My CSS style sheets are saved in a folder called **css**

You will now see a somewhat distorted view in Dreamweaver. This is due to using the 2 positioning techniques, the *relative* technique renders incorrect, but is absolutely fine when viewed in a browser.

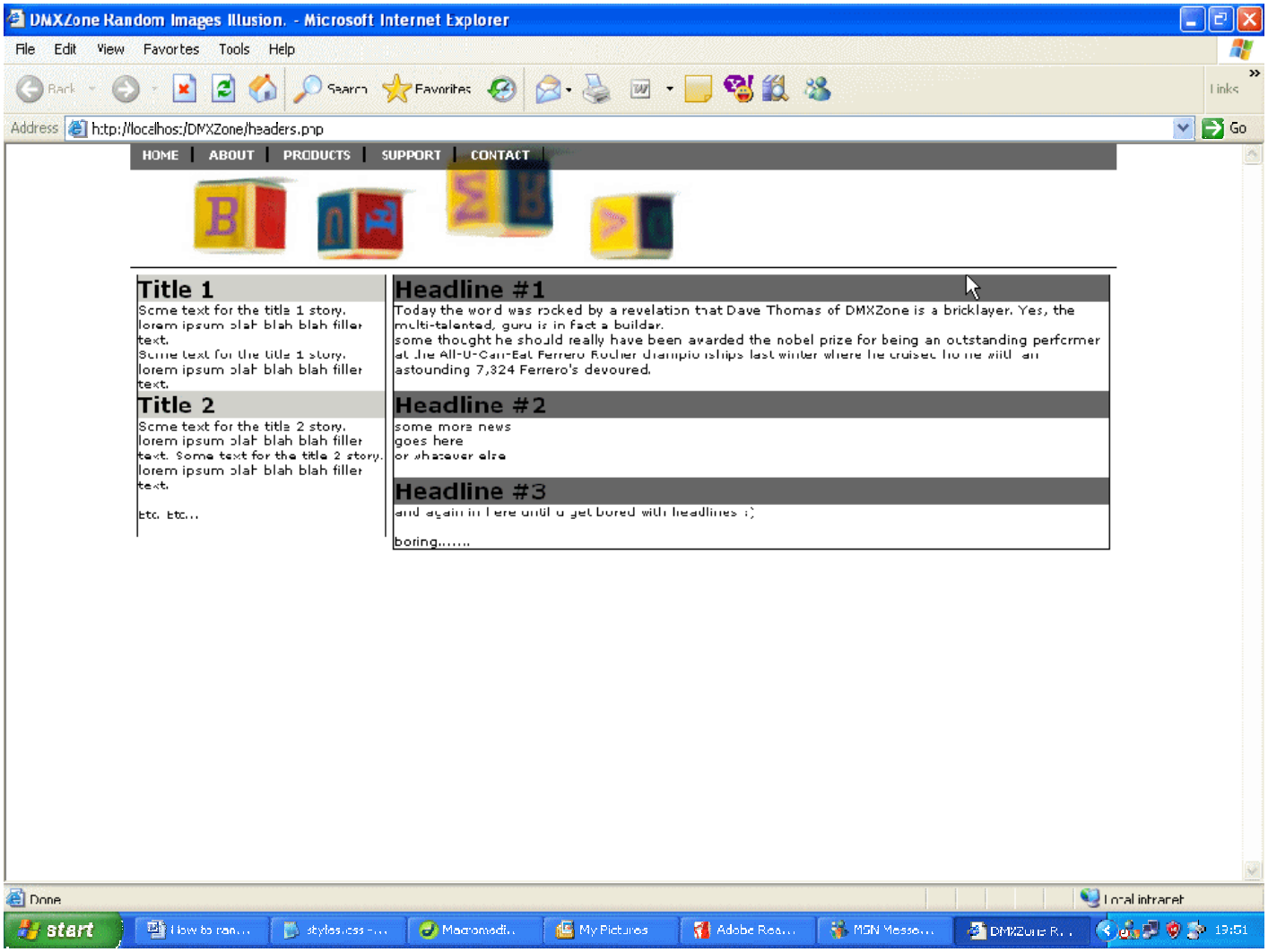


The somewhat distorted view of our CSS in Dreamweaver

But as promised, it will display fine in **Mozilla Firefox** and **IE6** as the screenshots below will show you. If anyone has any problems with any other browsers, please let me know and I will edit the article. As of now, I only have access to the 2 mentioned browsers.



Our display in Mozilla Firefox



Our display in IE6. Notice the header image has changed, but maintains the transparency illusion.

Summary

In this article I have demonstrated how to take CSS a step further by integrating it with a PHP script. This is just the tip of the iceberg if you think about the options we went over in the configuration of the randomise script. You could tweak the code to display a random .pdf in your page, or to call a different style sheet each time the page refreshes. There are many options available.

Author's Notes

I hope you find this article easy to understand and useful for some future design ideas you may have. I've recently 'jumped ship' from ASP over to PHP mainly due to the fact that most hosts now offer mySQL free so I'm busy playing around with the new options PHP has given me.

Many thanks to A List Apart and Dan Benjamin for permission to reprint the Random Image Script.

Enjoy, and please rate our articles if you read them.