



Don't miss our step-by-step guide, on sale now for £6.96 at Magbooks.com

INTRODUCTION

In the previous issue, we looked at the new styling options offered by CSS3. These allow you to build more engaging layouts and reduce the amount of coding involved in putting a page together. We also took a look at shadows, which can be applied to text and images. Here, we'll look at how you can achieve some fantastic typographic effects. At present, Internet Explorer lacks support for shadows, but it will apply the effect when it catches up.

Finally, we'll show you how to rotate text to create more brilliant effects, including flashes and tables of contents.

Nik Rawlinson

Web developer



letters@computershopper.co.uk

Mastering HTML5: part 4

You can make onscreen text more attractive with the help of shadows, banners and rotations. Here Nik Rawlinson explains how to make some typographical improvements with real style

Shadows can be very useful on web pages, particularly for created pressed and embossed text. Embossed text looks like it is raised from the page, whereas pressed text looks sunk into the page. You could combine the pressing with ink, but here we're going to emulate paper that has been pressed without any ink applied, as it gives a more realistic result.

When working with shadows, your first consideration should be where your virtual light source will be positioned; in most cases, this is above your visitor's monitor. We're going to set ours above and slightly to the left, in which case the light would catch on the inner bevel on the right and bottom of each letter. No light will fall on the other two edges, as they will be facing away from the light source. The formula for casting shadows on text is:

```
text-shadow: x-axis offset, y-axis offset, amount of blur, colour;
```

The amount of blur is optional. To cast a sharp shadow to the lower right of a line of text, you'd use:

```
.shadow {
  text-shadow: 0.2em 0.2em #ccc;
}
```

To soften that shadow for a more realistic result, you'd add a small amount of blur:

```
.shadow {
  text-shadow: 0.2em 0.2em 0.2em #ccc;
}
```

With that in mind, we'll define a new <div> to hold our text, and apply a light shadow to the specified edges to emulate lightfall. The code for the HTML is:

```
<div id="layer">
<p class="shadow">Computer Shopper</p>
</div>
```

And for the CSS, the code is as follows:

```
<style>
#layer {
  background: #900;
}
p {
  padding: 2em;
  color: #600;
  font: 3em Arial;
  font-weight: 900;
}
.shadow {
  text-shadow: 0.03em 0.03em 0.03em #D00;
}
</style>
```

The result is a subtle – but recognisable – sinking of the text into the page; this is helped by the fact that we have darkened the letters slightly by picking a darker tone of the red we used for the background of our <div>. Picking a different colour would have spoiled the effect. Use a colour picker, such as the table maintained at http://en.wikipedia.org/wiki/Web_colors#Color_table.

To create the illusion of embossed text, we only need to swap our text and background colours, and the sides on which our light falls; we don't need to make any changes to our HTML. Our amended CSS looks like this:

```
<style>
#layer {
  background: #600;
}
p {
  padding: 2em;
  color: #800;
  font: 3em Arial;
  font-weight: 900;
}
.shadow {
  text-shadow: -0.03em -0.03em 0.03em #D00;
}
</style>
```

THE REAL DEAL

This is a decent bad start, but if you were to increase the size of your text – say, to 8em – and look closely at the corners of each character, you would see that there is a disconnect between the shadow and the characters that cast it, so the effect works only on smaller text sizes. To create a more believable feeling of depth on larger characters, we must take advantage of CSS3's ability to layer several shadow styles, each one on top of its predecessor, by putting a comma between them.

We don't need to touch our HTML to achieve this, but we will increase the text size in our CSS from 3em to 8em and re-write the .shadow class as follows:



▲ By darkening the letters, the pressed text looks a lot more realistic

```
.shadow {
  text-shadow: -0.01em -0.01em #D00,
    -0.02em -0.02em #D00, -0.03em
    -0.03em #D00, -0.04em -0.04em
    #D00, -0.05em -0.05em #D00,
    -0.06em -0.06em #D00, -0.07em
    -0.07em #D00, -0.08em -0.08em
    #D00;
}
```

Here, we've laid eight shadows on top of each other; each one is offset 0.01em from its predecessor, which gives us a total depth of shadow of 0.08em at the deepest point. Even with large font sizes, this creates a believable sense of depth as the characters recede into virtual 3D space. For clarity, we have changed the colours in the CSS, so that the results in our image appear as grey on white.

PAGE TURNER

Our text stands out more clearly now, but shadows aren't the only tool we can use to draw our reader's attention to our text. CSS3 makes provision for rotating elements on the page, and here Internet Explorer's support is up there with the best of them. Rotation, which can be applied to images and text, needs to be done individually for each browser architecture; the transform command must be preceded with -moz for Firefox, -webkit for Safari and Chrome, -o for Opera and -ms for Internet Explorer 9.

To rotate the text in all the paragraphs on our page 30° counter-clockwise, we add the following code to the previous CSS listing:

```
p {
  -ms-transform: rotate(-30deg);
  -moz-transform: rotate(-30deg);
  -o-transform: rotate(-30deg);
  -webkit-transform: rotate(-30deg);
  color: #800;
  font: 3em Arial;
  position: absolute;
  top: 75px;
  font-weight: 900;
}
```

Because the text rotates around its central point, we have introduced two new lines to our CSS – for position, and for top – to stop it swivelling beyond the edges of the browser window.

As you'll remember, our paragraph is contained within a <div> called 'layer'. We want it to take its measurements with reference to that layer, so we specify 'position: absolute'. If we were to use 'fixed' instead, the element would rotate around a point relative to a position within the



▲ Incorporating a shadow can make your text appear to be embossed



▲ As the image remains untouched, it's easy to enable or disable overlays such as this

browser window rather than the layer that holds it. The rotation point is 75 pixels down from the top of whichever object it is using as its reference point, as denoted by the line 'top: 75;'.

CLASS DISTINCTION

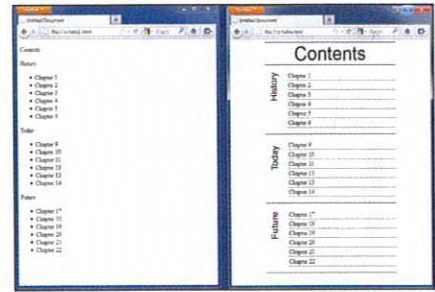
You may think that being able to rotate your text is just a fun novelty, but it does have some very practical real-world examples. Take the following code, for example:

```
<style>
#layer {
  width: 450px;
  height: 320px;
  border: 1px solid black;
}
.sheep {
  background-image: url('lamb.jpg');
  top: 1px;
  left: 1px;
}
.sticker {
  -ms-transform: rotate(-30deg);
  -moz-transform: rotate(-30deg);
  -o-transform: rotate(-30deg);
  -webkit-transform: rotate(-30deg);
  color: #fff;
  font: 2.5em Arial;
  position: absolute;
  top: 55px;
  font-weight: 900;
  background-color: red;
  padding: 0 10px;
  border: 3px dotted white;
}
</style>
```

```
<div id="layer" class="sheep">
<p class="sticker">&pound;25 off
today!</p>
</div>
```

This code could be used as the basis of an online catalogue, with a general '#layer' attribute used to define the proportions of a <div> (450 by 320 pixels with a one-pixel solid border) that can be recycled for each product in the catalogue by applying a class. This time, the catalogue deals with livestock and happens to be selling sheep. We have applied the class 'sheep', which, in our stylesheet, positions it one pixel from the top and left of the browser margins and adds an image of a lamb as the background.

Remember that named <div> elements ('layer' in this case) have their styles defined in the stylesheet with a leading #, whereas classes that are applied to layers, text, images and so on should be preceded by a dot; this is why we are able to apply multiple styles to this <div> while only drawing on a single class.



▲ Building a good-looking table of contents requires a minimum of CSS code

BANNER TIME

We may want to draw our visitors' attention to an issue that affects one of the items in our catalogue by applying text within a banner, overlaid on top of the image as though it is a sticker. In this instance, there's a £25 discount on lambs if bought the same day. We'll apply the class 'sticker' to the text, which is then rendered by the stylesheet in white on a red background, rotated by 30°.

We've also applied padding (padding: 0 10px;) to each end of the sticker, so that the characters don't bump into the edge, and a dotted white border to draw the eye. Consider the following:

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

We could have used this to define the padding applied to the top, right, bottom and left sides respectively, but by cutting it to just '0 10px' we have styled the top and bottom together (0 pixels) and the two sides (10px each), reducing the length of our code.

As well as making our text more dynamic, styling it this way has the added benefit of not requiring the use of an image editor. Because the image of the lamb remains untouched, we can remove the text when the offer expires, without having to swap out the picture. Equally, the sticker contents could easily have been retrieved from a back-end database, allowing any staff who are unfamiliar with HTML to update the image overlays simply by using a text entry field on a web form, rather than having to fiddle with the underlying code.

UNDER THE TABLE

Finally, we can use the idea of rotated text to build a contents table for an online reference book. It will have three sections – past, present and future events – and each will contain a number of individual chapters. We can achieve this by corraling our sections within DIV layers and our chapters in an unordered list.

The results may lack professionalism and do little to mimic a printed table of contents. However, by offsetting the contents so that they sit beside – rather than below – the section headings, and rotating those headings, we can improve the finished layout enormously, without requiring graphics, and with text that remains selectable and editable at all times. ☞

Next month

ANIMATION TOOLS

Discover the animation tools that don't require complex, expensive extras.