

# Mastering HTML5: part 3

CSS3 is an integral part of coding with HTML5, and builds on the functionality and flexibility of its predecessor. **Nik Rawlinson** explains how to style up your pages and add some great effects

Ten years ago, we'd have thought nothing of peppering our HTML with styling information. Tags that defined fonts, sizes and colours would break up the flow of words and pictures. This made the raw source of the page hard to follow, so any third party who needed to edit your code would have had to pick through confusing, fractured HTML. Also, if you later wanted to change the style of your headings, text or titles, you'd have to do it several times throughout each document, every time one appeared.

Hiving off the styling to an external, referenced file (a style sheet) lets you apply styles to particular elements on the page. Define the look of a heading tag, such as `<h1>` or `<h2>`, within the CSS and it's applied every time those tags appear on your page, without you having to define them again wherever you use them. As the style is defined once in the attached file, you have to change it only that one time, and the change is reflected through your document. It's an enormous time-saver.

With the advent of HTML5, this is more important than ever. Fully validated HTML5 makes no provision for styling content within the pages. If you've never tried your hand at CSS, now is definitely the time to start.

## BORN THREE

Just as HTML5 improves on HTML4.01, CSS3 takes the best of what was already in place in CSS2 and adds new

elements such as drop shadows, native rounded corners and animations that allow designers to build more flexible pages in native code, rather than resorting to third-party tools, image editors and plug-ins. CSS3 will let you lay out textual content in columns, as you might in a magazine, specify where breaks fall and how wide the gaps should be between columns. You can use images as borders, stretching them as appropriate, and target elements specifically on the page.

It is already possible for an action to be carried out only when the visitor's mouse is in a particular position in relation to the content on your page; using the `:hover` modifier. However, new modifiers such as `:checked`, `:enabled` and `:not` mean you can apply styles to selected radio buttons, enabled page elements and even objects that are not of a kind specified.

Work started on CSS3 in mid-1998, and the first elements were published in 1999, but it's only now that browser support has advanced sufficiently to allow us to use it with confidence on a live site. However, the degree to which different browsers render the various elements varies greatly. Luckily, you can target specific browsers with fixes by isolating each rendering engine. CSS3 allows for rounded corners, for example, which we could apply to a container called `boxout` as follows:

```
.boxout {border-radius: 20px; }
```

This isn't universally supported, and won't render properly on older browsers. You must make allowances for this by targeting the earlier Webkit (Safari 3 and earlier) and Mozilla (Firefox) engines as follows:

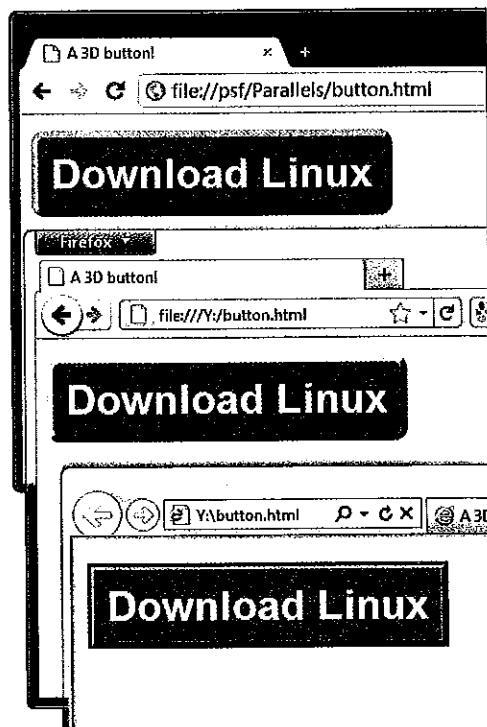
```
.boxout {
  border-radius: 20px;
  -moz-border-radius: 20px;
  -webkit-border-radius: 20px;
}
```

Browsers that don't need the specific `-moz` and `-webkit` targets will ignore them. This workaround is easy to implement, compliant and won't throw up an error.

## ANOTHER DIMENSION

We'll put this into practice by creating a simple CSS3 button using code instead of graphics. You may wonder why we'd do this when we could just knock up a button in Photoshop, compress it and embed it within the page. There are two very good reasons. First, it makes lighter demands on your server. The code below that defines the two button states is 285 bytes in length. Doing the same with images would require two graphics – one for the up state and one for the down – and that would total several kilobytes each. Savings of this type soon pay dividends when implemented across all pages of a busy site.

Second, the content of the button itself – the words 'Download Linux' – is already search engine-optimised.



▲ Firefox and Chrome render the button well, but IE9 can't handle the rounded corners without extra code

## INTRODUCTION

In the previous two issues, we explored the ways in which HTML5 lets you code more engaging web content in less time. All styling elements must be removed the main flow of code. From now on, pure CSS is used to define the layout and look of your pages.

This issue, we're turning our attention to CSS3, which is being developed alongside HTML5. We'll explore two of the most radical effects applied with CSS – shadows and rounded corners – and how CSS3 addresses these commonly requested features of modern web design in a clean and simple manner.

**Nik Rawlinson**

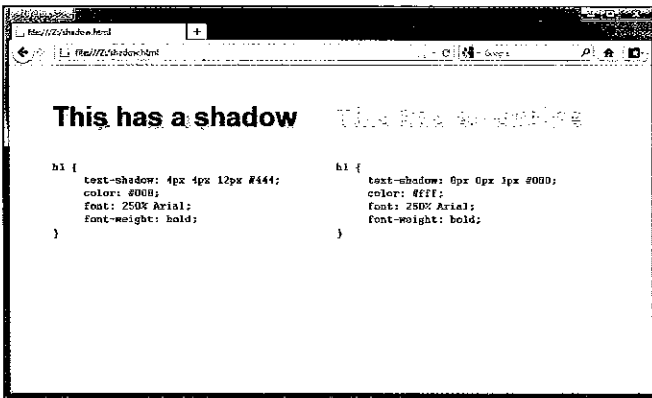
Web developer



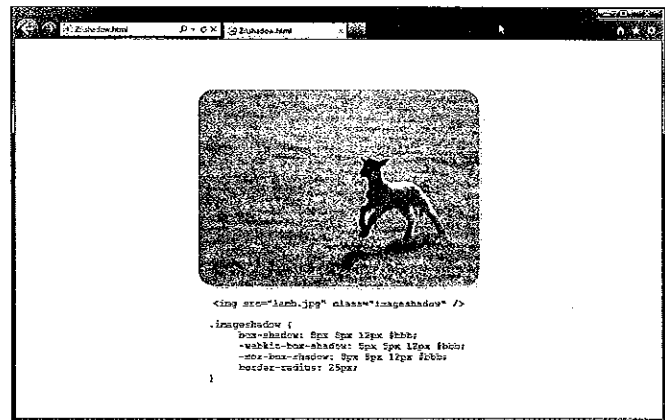
letters@computershopper.co.uk



Don't miss our step-by-step guide, on sale now for £6.96 at [Magbooks.com](http://Magbooks.com)



▲ By changing the offset of your shadow, you can quickly and easily create outlined text using CSS3



▲ Shadows follow the contours of the box object to which they are applied

If you were using a graphic, you would need to use `<alt>` tags to describe the content to search engine crawlers, and possibly also apply a 'title' attribute to the `<a>` tag to further improve its indexing. Screen readers will see our button as nothing more than a standard link to which we've applied some styling, which will greatly ease navigation for visually impaired users:

```
<!DOCTYPE html>
<head>
<title>A 3D button!</title>
<style type="text/css">
#button a {
background-color: #39F;
border: 5px outset #6CF;
border-radius: 10px;
font: 24pt Arial;
font-weight: 900;
color: #fff;
padding: 10px;
text-decoration: none;
}
#button a:hover {
border: 5px inset #6cf;
padding: 9px 11px 11px 9px;
}
</style>
</head>
<body>
<br />
<div id="button"><a href="download.
html">Download Linux</a></div>
</body>
</html>
```

The button has no fixed dimensions, so will always be 10 pixels larger than its contents in each direction, allowing us to re-use the code wherever we want to create a button by simply placing a link inside a `<div>` tag called 'button'. By combining the 10-pixel padding with CSS3's new border-radius attribute, we've given our button rounded corners and then we've given it a sense of depth by extruding the whole thing by five pixels, using the border outset and inset attributes on the up state and down state buttons respectively.

One further change made to the downstate button – which is displayed when the visitor hovers their mouse over the link – is to shift the text one pixel up and to the left. We have done this by reducing the top and left padding to 9px in each case, and increasing the right and bottom padding to 11px to maintain the overall size of the button. The text now follows the motion of the button in stepping back into the page.

When specifying the padding on each of the sides of our button, we are able to use just '10px' once on the up state button as each is the same, but on the down state button we must specify the padding on each side individually ('9px 11px 11px 9px'), starting with the top and working around in a clockwise direction. Previewing the results in Chrome, Firefox and IE, we can immediately spot a problem here. Chrome (which uses Webkit) and Firefox (which uses the Gecko) perform as we would expect but IE9, which uses the Trident rendering engine, does not, despite supporting the border-radius class.

To overcome this problem, we must notify IE9 of the document coding by adding the following line to the coding's `<head>` area:

```
<meta http-equiv="X-UA-Compatible"
content="IE=9" />
```

Re-loading the page, we can now see that it correctly renders with rounded corners in all three browsers. Although IE9's support for CSS3 is better than that of its predecessors, it's by no means perfect, and it lacks support for many elements that remain draft proposals. Microsoft maintains a list of the current supported CSS classes in each version of its browser from IE6 to IE9, which you can find at <http://bit.ly/lBbsvV>.

## CASTING SHADOWS

Our button is complete, but rounded corners are just one of the new features open to us by switching to CSS3. Firefox, Safari and Chrome support drop shadows on both text and graphics, allowing you to apply these effects to your content, again without needing a graphics tool to render them as a JPEG image. The formula for applying shadows is as follows:

```
text-shadow: x-position y-position
blur-amount colour;
```

To apply a mid-grey shadow to the bottom right of a heading (as though cast by a light positioned above and to the left of the text), you'd use the following code:

```
h1 {
text-shadow: 2px 2px 4px #bbb;
}
```

To cast your shadow upwards, as though lit from below, you would change the second pixel measurement (the y-position) to a negative figure, such as -2. However, by removing both offsets and darkening the shadow colour, we can

modify this effect to produce a keyline around only text. Changing the font colour to white (`#fff`) would then result in outlined characters. So for example:

```
h1 {
text-shadow: 0px 0px 4px #000;
}
```

IE9 doesn't currently support text shadows rendered in CSS3, so it ignores these commands when parsing the stylesheet. The result will be solid text without a drop shadow or an outline.

You can apply a shadow to any onscreen element within a virtual bounding box, such as an image or a div layer. Here we'll define a class called 'imageshadow' to do just that:

```
.imageshadow {
box-shadow: 8px 8px 12px #ccc;
-webkit-box-shadow: 8px 8px 12px
#ccc;
-moz-box-shadow: 8px 8px 12px #ccc;
}
```

IE9, Opera 10.5 and Firefox 4 natively support the box-shadow attribute, but Safari, Chrome and earlier editions of Firefox (up to 3.5) do not, and so they need the `-webkit` and `-moz` variants detailed above. To apply this effect to an image, simply call the class from the image tag:

```

```

As a live effect, we can combine our shadow with other effects, such as the border radius that we used earlier, and it will follow the contour of the element to which it's attached.

## SAVING GRACES

CSS3's enhancements will considerably reduce the amount of work that's involved in building a website. What would once have involved the use of graphics and animation tools can now be achieved natively using nothing but code, thereby tightening the development cycle, reducing server load and ensuring that pages load faster for visitors. ☑

## Next month

### GOING FURTHER WITH CSS3

More new tools in CSS3 to make onscreen text more attractive and appealing.