

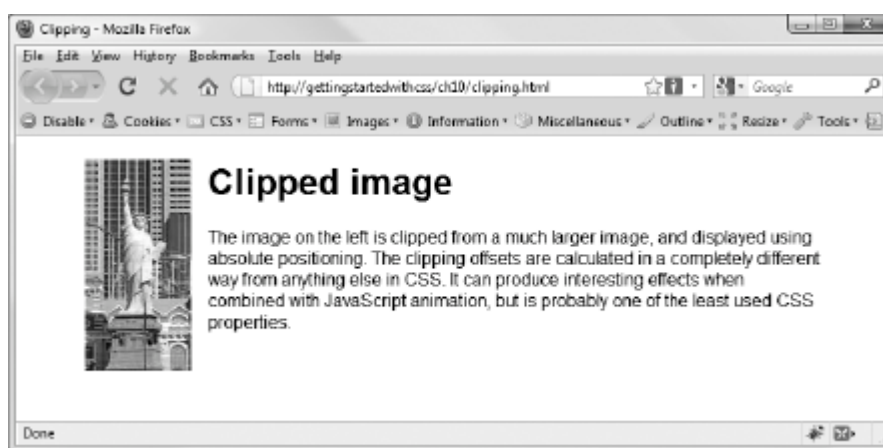
Cropping an image nondestructively with CSS

The `clip` property is similar to a mask or cropping tool in that it creates a window through which part of an element is displayed. The best way to understand it is to compare [Figures 10-23](#) and [10-24](#). [Figure 10-23](#) shows a photo of the Strip in Las Vegas with just the Statue of Liberty highlighted. [Figure 10-24](#) shows the same image, `lasvegas.jpg`, displayed in `clipping.html`. If you check the download files, you'll see that `lasvegas.jpg` hasn't been cropped. It's the same image as in the scaling image exercise in [Chapter 6](#). The `clip` property simply masks the sections of the image dimmed in [Figure 10-23](#).

Figure 10.23. The `clip` property lets you select a rectangular area to be displayed while the rest is hidden.



Figure 10.24. Only the selected area of the image is displayed, even though it hasn't been physically cropped.



The `clip` property works *only* with absolutely positioned elements, and the only shape you can use is a rectangle. You define the area that you want displayed by supplying four offsets in clockwise order starting from the top like this:

```
clip: rect(top, right, bottom, left);
```

However, *all the offsets are calculated from the top-left corner* of the element you want to mask, and each must be defined as a length. You cannot use percentages.

The top edge of the highlighted section is 146px from the top of `lasvegas.jpg`; the right side is 140px from the left; the bottom is 336px from the top; and the left side is 45px from the left. So, to clip the image, this is the style rule I created:

```
#clipped_image {  
  position: absolute;  
  clip: rect(146px, 140px, 336px, 45px);  
}
```

Just to make life even more difficult, the clipped element doesn't move—it is, after all, absolutely positioned. So, in `clipped.html`, the Statue of Liberty was 146px too far down the page and 45px too far right. To compensate for that, I needed to use negative `top` and `left` offsets like this:

```
#clipped_image {  
  position: absolute;  
  clip: rect(146px, 140px, 336px, 45px);  
  top: -146px;  
  left: -45px;  
}
```

Oh yes, another snafu—IE6 and IE7 won't play ball if you separate the offsets with commas, which is the correct way of doing things. Fortunately, other browsers accept leaving the commas out like this:

```
clip: rect(146px 140px 336px 45px);
```

With all these problems, it's small wonder that `clip` is one of the least used CSS properties. However, it can be combined with JavaScript to create animated effects, revealing and concealing images.