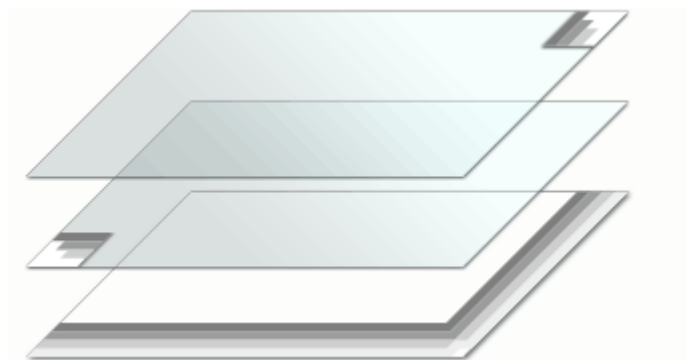

This article was originally published at *A List Apart*. You can find it online at its permanent URL: <http://www.alistapart.com/articles/onionskin/>

Onion Skinned Drop Shadows

Yes, Onion skinned. Animators use onion skinning to render what is impossible to see otherwise: a snapshot of motion across time. Now, web designers can use it to render another seeming impossibility: the truly extensible CSS-based drop shadow.



Enough Already

Tired of reading technique after technique outlining the many ways to skin this cat? Just consider this a demonstration of the capacity of CSS to surprise. Just when it looks as though the method is clear and the way is defined, one more wandering mind reveals another creative approach. If all you glean from this article is a deeper appreciation for the diversity of solutions CSS affords, then you've benefited far more than any one technique could deliver. Read on.

What's different

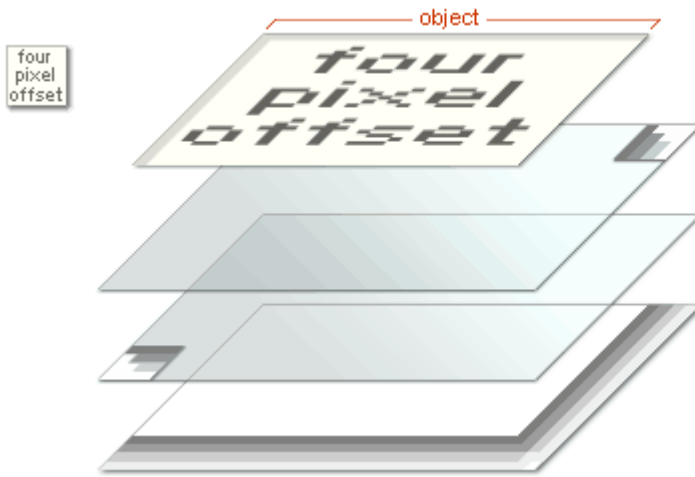
There are several differences between this technique and those offered previously, both here and elsewhere. Some may find it simpler than others, some more stable; but no matter how you look at it, it is more extensible. With just one set of simple rules, these new drop shadows:

- ◆ Automatically expand and contract to fit any object, without specifying widths
- ◆ Allow you to modify the shadow depth with *no image manipulation*
- ◆ Render the same across all browsers (Appendix A) without cutting any corners

Peeling The Onion

Onion skinning refers to a technique in time-based arts of overlaying several frames in a sequence to discern the subtle differences between them. The animator is simply looking through a stack of layers to see one whole motion through a composite of its parts. What is opaque on one layer can be seen through the transparent areas on a layer above it. We can do the same thing with `divs`. Using CSS, we can stack `divs` together in concentric fashion, like the layers of an onion, to form a composite of anything we like. Today, we are making drop shadows.

Isometric view of an onion skinned drop shadow:



Three divs, each containing a different shadow image component (Appendix B), overlay one another to form a composite drop shadow. The markup couldn't be simpler. Just wrap your object inside three `div`s, one inside the other, and give each a class name which corresponds to its role. Since we're "wrapping" the object, the classes we'll assign for demonstration purposes are named `wrap1`, `wrap2`, and `wrap3` (`.wrap1` assigned to the outer, and `.wrap3` the innermost `div`).

```
<div class="wrap1">
  <div class="wrap2">
    <div class="wrap3">
      
    </div>
  </div>
</div>
```

The CSS is a bit more complicated – but not much. There are basically three things that the style rules must accomplish for the shadow:

- ◆ **Make It Show.** Assign each of the three shadow image components (shadow + two corners) to a different layer in the onion skin stack.
- ◆ **Make It Drop.** Create an offset which moves the drop shadow down, and to the right of the object.
- ◆ **Make it shrink-to-fit.** Force the `div`s surrounding the object to collapse upon it.

Step One: Rendering The Shadow

The basic idea is to assign one of the three shadow image components to each of the `div`s through its corresponding `class`. The sequence of which `div` gets which component is crucial, though. The purpose for having three image components is to use the two small ones at the top-right and bottom-left positions to mask the clipped edges of the large shadow revealing only soft, rounded edges all around.

In order for one image to mask another, it must reside above it in the z-index. We don't have to worry so much about indexing the stack, since the nested structure of our `div`s carry's an intrinsic stacking order that works for us. We simply assign the class with the shadow to the outermost `div` at the bottom of the stack. Since the `div`s inside it will naturally sit on top, we'll give the classes with the corner images to them.

```
.wrap1 {
  background-image:url(shadow.gif);
}
.wrap2 {
  background-image:url(corner_bl.gif);
}
.wrap3 {
  background-image:url(corner_tr.gif);
}
```

Once we've assigned the right images to the right classes, we need to give them position - `background-position`, that is. If we left the rules alone at this point, all we would see is a tiled background of 'corner_tr.gif' since it's sitting highest in the stacking order. Remember, onion skinning requires transparency in the upper layers so that we can see what's beneath. To do this, we'll cancel the repeating property of all the background images with `no-repeat`, and position them where they belong to form the composite of our drop shadow.

Common sense tells us that an image named 'corner_tr.gif' should reside in a corner - probably the top-right; ditto for 'corner_bl.gif' (bottom-left). But, what about the large shadow image? Does it even need position? Yes-more than any other, in fact. If we want our shadow to fall to the bottom-right of whatever object we put it under, we must specify that direction in our `.wrap1` rule. Otherwise, it will automatically fill the `div` relative to its top-left corner, the opposite of what we need.

The Base Layer: `div.wrap1`

The outermost `div` holds the largest of the three shadow components, which is positioned `right bottom`.

```
.wrap1 {
  background:url(shadow.gif) right bottom no-repeat;
}
```



The Middle Layer: `div.wrap2`

The second div, nested inside the first, masks the clipped lower-left corner of the shadow underneath, giving it a rounded appearance.

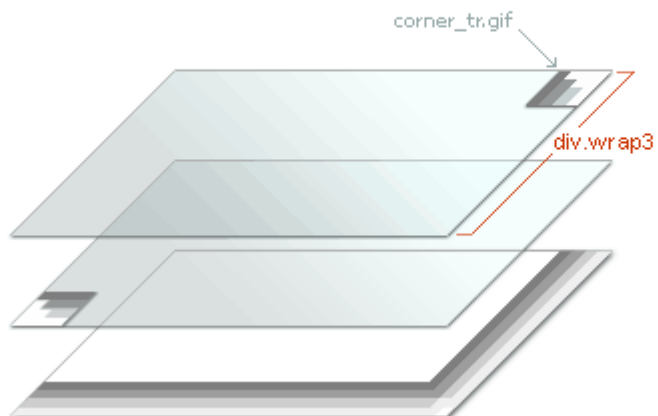
```
.wrap2 {  
  background-image:url(corner_bl.gif) left bottom no-repeat;  
}
```



The Inner Layer: `div.wrap3`

The third div, nested inside the second, takes care of the clipped corner in the upper right.

```
.wrap3 {  
  background-image:url(corner_tr.gif) right top no-repeat;  
}
```



Step Two: Dropping It Down

The next step for the CSS is to create the offset that makes the drop shadow drop. This could not be more simple. All it takes is a little bit of `padding` on the right and bottom of the innermost `div`. When the padding causes this `div` to expand away from the object inside it, the two outer `divs` expand with it. The result: all three shadow components, positioned along the right and bottom sides of their `divs` move in tandem, and can now be seen through the gap created by the padding.

```
.wrap3 {
  padding:0 4px 4px 0;
  background:url(corner_tr.gif) right top no-repeat;
}
```



Modifying The Offset

Changing the amount offset for your shadows is *almost* as easy as simply changing the `padding` values on the `.wrap3` rule. We say "almost" because adjusting the `padding` merely moves the shadow while the corners continue to hug the edges of their containers. To accurately simulate a shift in offset, you'll need to tweak the `background-position` of both corners relative to the `padding`.

Some would say that it's good enough to simply adjust the padding, and leave it at that. No sense complicating things to achieve a nuance that is barely discernable for most people anyway. Others would argue that it cheapens the effect to cut corners this way. If a method will support better aesthetic and technical fidelity, as designers we're obligated to use it to its full potential. It's likely that most, however, could go either way.

Judge For Yourself.



This image is used below to demonstrate two sets of drop shadow styles. In the first set of examples, the offset is modified using `padding` values in the `.wrap3` rule, which moves only the shadow-not the corners. In the second set, both the `padding` and `background-position` values of the corners are modified so that the entire shadow, corners and all, moves as the offset changes.

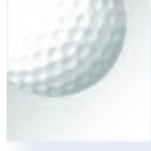
Set 1. Padding Only



8 pixels



12 pixels



18 pixels

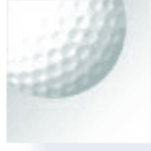
Set 2. Padding and Background-Position



8 pixels



12 pixels



18 pixels

If you can discern the difference between the two sets, and you prefer set two, then you'll need to adjust the `background-position` of your corner images to compensate for the offset in `padding`. Further, you'll need to add additional white pixels to the outside edges of your corner images. This allows them to move away from the boundary of the `div`, without losing their ability to mask the clipped edge of the shadow underneath. Each shadow style is a little different, and as you begin experimenting with offsets, your particular corner white space requirements will become clear.

Step Three: Making it Shrink-To-Fit

A little sleight of hand was necessary to coax every browser that was tested into conformity with this requirement, a potential deal-breaker. But, without this behavior, each and every instance of the shadow style would require the foreknowledge of the object's width in order to specify the width of the shadow. Obviously, that wasn't going to cut it.

Most of the browsers tested would allow the `divs` to collapse when they received the `float` property. This would have sufficed, if it weren't for those of you who use IE5 on the Mac. The mere fact that you exist was reason enough to explore alternatives. Unfortunately, none were found. At least, none that worked universally.

Some research and experimentation (<http://www.ploughdeep.com/dropshadow/ala/macfix.html>) eventually revealed that `inline-table`, an obscure CSS display property value, would be the saving grace of this technique. Right, wrong, or indifferent, this was all that could be found, and it would have to do. So, using the commented backslash method (http://www.sam-i-am.com/work/sandbox/css/mac_ie5_hack.html) to isolate Mac IE5, we give it `display:inline-table`, and all the rest receive `block`. So there it is, shrink-to-fit-everything, everywhere, in every browser.

The Composite Result:



With all three steps in place our CSS looks like this:

```
.wrap1, .wrap2, .wrap3 {
  display:inline-table;
  /* \*/display:block;/**/
}
.wrap1 {
  float:left;
  background:url(shadow.gif) right bottom no-repeat;
}
.wrap2 {
  background:url(corner_bl.gif) left bottom no-repeat;
}
.wrap3 {
  padding:0 4px 4px 0;
  background:url(corner_tr.gif) right top no-repeat;
}
```

One more rule, for good measure

As in the above example, images will often be used with this technique. For those occasions, an additional rule which sets a shadowed image's display property to `block` will help keep unwanted white space from intruding on a good thing. Just add the following rule to your style sheet:














```
.wrap3 img {  
  display:block;  
}
```

Do It Yourself

At this point, you may want to browse the drop shadow gallery (Appendix C) to get a feel for what is possible with this technique. We've made it easy to download a variety of shadow source images so you can get started.

Appendix A:

Successfully tested browsers and platforms

	 98SE	 ME	 2000	 XP	 OS 9.0	 OS X	 Linux
 IE	5.0, 5.5	5.5	5.0, 5.5, 6.0	6.0	5.1	5.2	
 Netscape	6.1, 6.2, 7.1		6.1, 6.2, 7.0	6.1, 6.2, 7.1		6.2, 7.0	7.0
 Mozilla			1.2, 1.5, 1.6	1.6		1.6	1.6
 Firefox			0.8		n/a		
 Safari	n/a	n/a	n/a	n/a	n/a	1.0	
 Opera			6.0, 7.0	7.0			

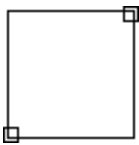
Appendix B:

Anatomy of a Drop Shadow

To complete a drop shadow that is capable of expanding and contracting around an object it must first be broken into three parts:

- ◆ A large image forming the shadow perimeter
- ◆ A small image which forms the top-right corner of the drop shadow
- ◆ Another small image which completes the lower-left corner

Illustration:



Appendix C:

Drop Shadow Gallery



Get source images:

http://www.ploughdeep.com/onionskin/images/shadows/v1/corner_tr.gif

http://www.ploughdeep.com/onionskin/images/shadows/v1/corner_bl.gif

<http://www.ploughdeep.com/onionskin/images/shadows/v1/shadow.gif>



Get source images:

http://www.ploughdeep.com/onionskin/images/shadows/v2/corner_tr.gif

http://www.ploughdeep.com/onionskin/images/shadows/v2/corner_bl.gif

<http://www.ploughdeep.com/onionskin/images/shadows/v2/shadow.gif>



Get source images:

http://www.ploughdeep.com/onionskin/images/shadows/v3/corner_tr.gif

http://www.ploughdeep.com/onionskin/images/shadows/v3/corner_bl.gif

<http://www.ploughdeep.com/onionskin/images/shadows/v3/shadow.gif>



Get source images:

http://www.ploughdeep.com/onionskin/images/shadows/v4/corner_tr.gif

http://www.ploughdeep.com/onionskin/images/shadows/v4/corner_bl.gif

<http://www.ploughdeep.com/onionskin/images/shadows/v4/shadow.gif>



Get source images:

http://www.ploughdeep.com/onionskin/images/shadows/v6/corner_tr.gif

http://www.ploughdeep.com/onionskin/images/shadows/v6/corner_bl.gif

<http://www.ploughdeep.com/onionskin/images/shadows/v6/shadow.gif>