



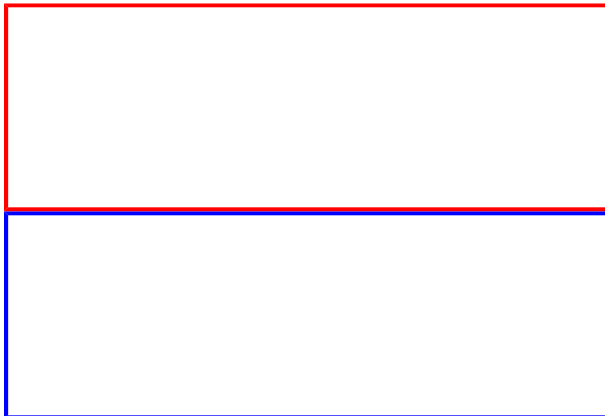
CSS Negative Margins - Part One

By: [Zoe Gillenwater](#)

Of all the pieces of the [CSS box model](http://www.communitymx.com/abstract.cfm?cid=20B41) (<http://www.communitymx.com/abstract.cfm?cid=20B41>), margins are probably the easiest to grasp. The margin is the transparent space outside the borders of a box that pushes other elements away from that box. But this is really only part of the definition. When you give a margin a negative value, it's a whole other ball of wax! This article will explain how negative margins work and begin to explore some of their uses. A following article will explain how negative margins can be used in float-based layouts to to achieve some interesting effects.

How Negative Margins Work

Margins are one of the few properties that can be given a negative value. A negative value essentially works in the reverse way that a positive value does. It's easiest to see visually, so check out these two divs:

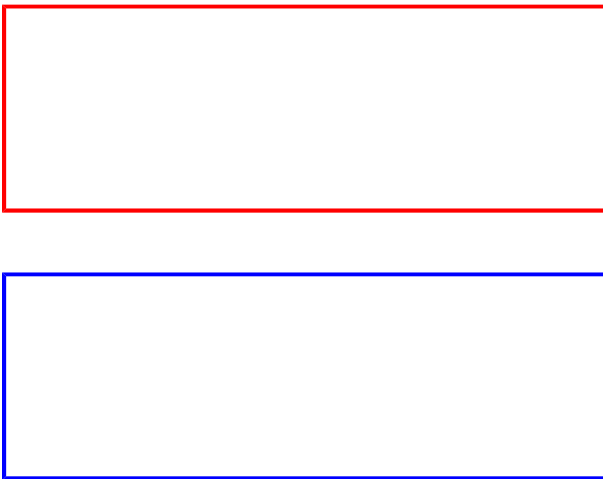


These divs have the following CSS properties:

```
#div1 {
  width: 300px;
  height: 100px;
  border: 2px solid #f00;
}
#div2 {
  width: 300px;
  height: 100px;
  border: 2px solid #00f;
}
```

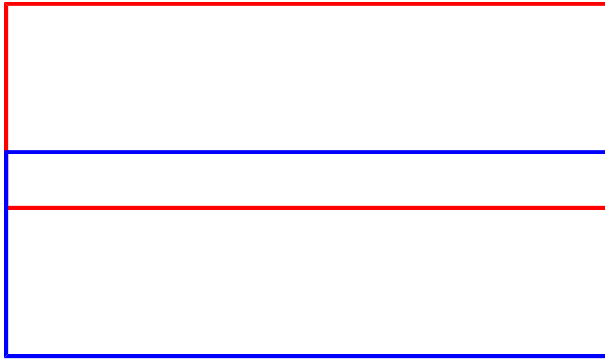
Note: If you look at the source code for this page, you will notice that there **are** left margins set on the two divs. This was done simply to make sure the boxes do not overlap the left CMX sidebar in later examples. The example code I've provided is for the hypothetical situation where the boxes are the only elements on the page.

Note that there is no margin set, so the margin values for each of the divs default to zero. This means the divs are moved zero pixels apart – they touch each other. Adding a positive margin value moves the divs apart by that amount:



```
#div1 {
  width: 300px;
  height: 100px;
  margin-bottom: 30px;
  border: 2px solid #f00;
}
#div2 {
  width: 300px;
  height: 100px;
  border: 2px solid #00f;
}
```

So if the blue div moves down from its original position by 30 pixels when the div above it has a bottom margin of 30 pixels, it follows that a value of -30 pixels will move the blue div **up** from its original position. Indeed, this is what happens:

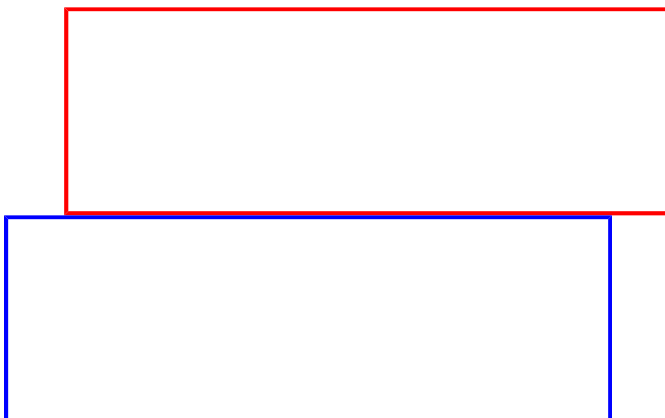


```
#div1 {  
  width: 300px;  
  height: 100px;  
  margin-bottom: -30px;  
  border: 2px solid #f00;  
}  
#div2 {  
  width: 300px;  
  height: 100px;  
  border: 2px solid #00f;  
}
```

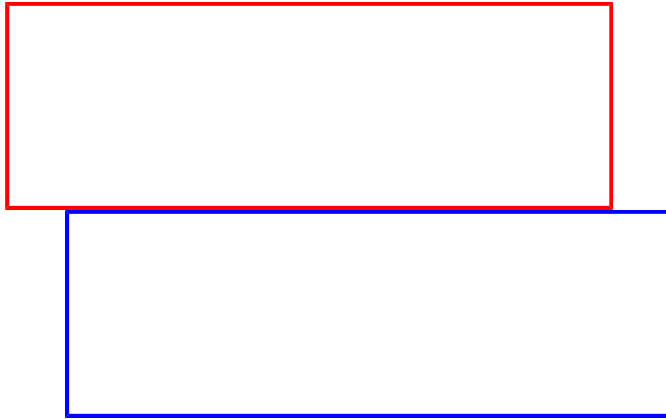
Note that the blue div now overlaps the div above it by 30 pixels! It's pretty simple: a box's positive margin value pushes surrounding content away from the box in the same direction as the margin, and a negative margin value pulls surrounding content over it.

I've used pixel values here for my example, but a negative margin value can take any sort of value, just like a positive margin.

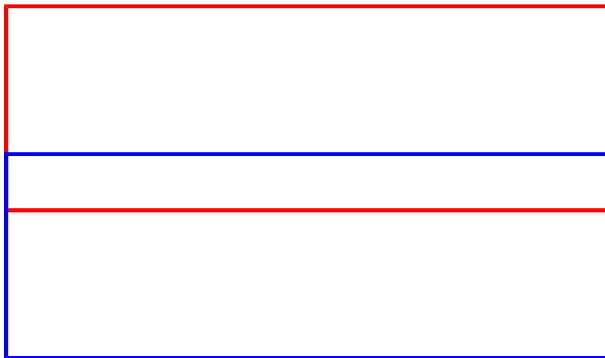
That's not all that negative margins do, however. Think back to positive margins on a box – they not only push surrounding content away from the box, they can also move the box itself. Going back to our original two boxes, what happens if we add a 30px positive margin to the left side of the red box?



The positive 30 pixel **left** margin on the box moves it to the **right** by 30 pixels, away from the edge of its containing div. So I bet you can guess what happens when we change that 30 pixels to a negative value?



With the **negative** 30 pixel **left** margin, our box has moved to the **left** by 30 pixels. Once again, a negative margin is simply the reverse of a positive margin. A positive margin pushes a box in the opposite direction of the margin (a left margin pushes it right, a top margin pushes it down, etc) while a negative margin pulls a box in the same direction as the margin. This means that we could have caused that overlap between the red and blue boxes in another way: by adding a negative top margin to the blue box instead of a negative bottom margin to the red box.



This produces the same visual effect as before, but with different CSS:

```
#div1 {
  width: 300px;
  height: 100px;
  border: 2px solid #f00;
}
#div2 {
  width: 300px;
  height: 100px;
  margin-top: -30px;
  border: 2px solid #00f;
}
```

Summary

Just remember: a negative margin is just the reverse of a positive margin.

- A box's positive margin either pushes content away in the same direction as its margin, or pushes the box itself in the opposite direction of the margin.
- A box's negative margin either pulls content over it in the opposite direction as its margin, or pulls the box itself in the same direction as the margin.

Using Negative Margins

You might be thinking, "Ok, I get it, but why would I ever want my content to overlap?". Surprisingly, negative margins have many uses.

Bug Squashing

One of their most practical uses is in fixing bugs. For instance, Windows Internet Explorer, including IE 6, inserts an extra three pixels of space between the edge of a floated block and the edge of the following content. If the following content has a width or height assigned, the three pixel gap shows up as a space between the blocks. If it does not have a width or height, the blocks seem to sit next to each other, but the content *within* the following box is pushed over by three pixels for as long as the float extends beside it, shifting back over to the edge where it belongs after the end of the float. Setting a margin of zero does not get rid of this mysterious gap, but setting a margin of -3px pulls things back into their proper position. Here's an example of CSS for a floated sidebar and the content block that sits next to it:

```
#sidebar {
  float: left;
  width: 300px;
}
#content {
  margin-left: 300px;
}
/* hide from MacIE */
* html #sidebar {
  margin-right: -3px;
}
* html #content {
  height: 1%; /* this line is needed only if you have not declared a width
or height on #content above */
  margin-left: 0;
}
/* end hide */
```

For a detailed description of this bug and how the fix works, see [Float: The Bugs \(Part Three\)](http://www.communitymx.com/abstract.cfm?cid=1BF53) (<http://www.communitymx.com/abstract.cfm?cid=1BF53>) by John Gallant and Holly Bergevin. There is a similar IE and Opera bug where negative margins come in handy, as described in [Float: The Bugs \(Part Two\)](http://www.communitymx.com/abstract.cfm?cid=D5416) (<http://www.communitymx.com/abstract.cfm?cid=D5416>). WinIE5.x, MacIE, and Opera 6 add an extra three pixels of space to both sides of a floated image. This can be fixed with the following CSS:

```
img {
  float: left;
  width: 100px;
  height: 100px;
  margin: 0px; /* fixes Opera 6 */
}
* html img {
  margin-left /**/: -3px; /* fixes MacIE and WinIE5.x */
  margin-right /**/: -3px; /* fixes MacIE and WinIE5.x */
}
```

Creating Cool Visual Effects

In addition to squashing bugs, negative margins can also be used to create cool CSS effects. In the free article [Do You Want To Do That With CSS? - Multiple Column Lists](http://www.communitymx.com/abstract.cfm?cid=27F87) (<http://www.communitymx.com/abstract.cfm?cid=27F87>), John and Holly show how creative use of margins can turn a single `ul` element into a three column list. Basically, a class is added to each list item specifying which column you want it to be in (col1, col2, or col3). Positive margins are used to move each of these "columns" over to the right, so they no longer line up vertically. Then, a second class is added to each list item that is the first in its column. These list items are given negative top margins to move them up, so all three columns are now on the same level and truly appear as columns! Check out the article for examples and the full CSS.

There are also times where overlapping may be an attractive visual effect. Let's take some text and a simple floated image:

Sheri's kitten
jumped into the
box and
enjoyed a nice
nap, but when
she awoke, try
as she might,
she couldn't
climb out!



Pretty standard stuff. Using a negative margin, however, can help us break up the boxiness.

Sheri's kitten jumped
into the box and
enjoyed a nice nap, but
when she awoke, try as
she might, she couldn't
climb out!



Now that kitten's getting somewhere! If you happen to have the misfortune of viewing this page in IE, however, you'll see that the photo has moved over, but the right side has been cut off. IE will cut off the portion of an object that overflows its parent container due to negative margins. Luckily, there's a quick fix: apply `position: relative` to the float, and the cut off portion appears!

Sheri's kitten jumped into the box and enjoyed a nice nap, but when she awoke, try as she might, she couldn't climb out!



The complete CSS used to create this effect is quite simple:

```
p {  
  float: left;  
  width: 300px;  
  padding: 10px;  
  border: 2px solid #999;  
}  
img {  
  position: relative;  
  float: right;  
  width: 200px;  
  margin: 0 -50px .5em .5em;  
}
```

You'll notice that I have not only floated the image, but the paragraph containing the image as well. This is a quick and easy way to ensure that a parent element contains its child floats. A more robust method is described in John Gallant and Holly Bergevin's [Clearing Floats, the Easy Way](http://www.communitymx.com/abstract.cfm?cid=93C7A) (<http://www.communitymx.com/abstract.cfm?cid=93C7A>), but I did not need to use it for this simple example.

It's not only floats that can be negatively margined – any block can be. Consider an unstyled heading and paragraph:

About CMX

At Community MX, you'll find fresh content daily for Web designers and developers. We focus on the most powerful and popular development tools available: the Macromedia Studio suite. From around the world, we've brought together 30 of the top experts in their fields. Together, we offer premiere content, such as articles, tutorials and extensions, as well as support service in our forums. There you'll find timely help for questions relating to Studio products, as well as other web development topics such as programming languages, CSS, and accessibility, among others.

Once again, negative margins can make things look a little more dynamic:

About CMX

At Community MX, you'll find fresh content daily for Web designers and developers. We focus on the most powerful and popular development tools available: the Macromedia Studio suite. From around the world, we've brought together 30 of the top experts in their fields. Together, we offer premiere content, such as articles, tutorials and extensions, as well as support service in our forums. There you'll find timely help for questions relating to Studio products, as well as other web development topics such as programming languages, CSS, and accessibility, among others.

Here's the CSS used to create this effect:

```
div {
  margin-top: 30px;
  padding: 20px;
  border: 1px solid #ccc;
  background: #eff;
}
div h2 {
  position: relative;
  float: left;
  margin: -30px 0 15px 0;
  padding: 8px;
  background: #ccc;
  color: #036;
  font: bold 1em Arial, Helvetica, sans-serif;
}
div p {
  clear: left;
  color: #036;
  font-size: 0.8em;
  line-height: 130%;
}
```

The div containing the heading and paragraph is given extra margin on top to make room for the overflowing heading. The heading's negative top margin moves it up, overlapping the border of the div. The `position: relative` on the heading keeps it from getting cut off in IE. To create the block of color behind the heading, it was not only assigned a background color, but floated as well. This is because headings are block elements and expand to 100 percent of their containers. Floating the heading constrained its width to the width of the text inside (this is known as "shrinkwrapping"). This also meant that the paragraph below needed to have a clear applied so it would begin on a new line under the heading.

Note: MacIE does not shrinkwrap floats as others browsers do. If you wish to support MacIE on this effect, you need to set `display: inline-block` on the floated heading, but hide this value from other browsers. For hacks to do this, see page three of [\(Not So\) Common Coding Problems with HTML and CSS — Part Six](http://www.communitymx.com/abstract.cfm?cid=A37EE) (<http://www.communitymx.com/abstract.cfm?cid=A37EE>) by John Gallant and Holly Bergevin.

Conclusion

You've learned how negative margins can move blocks around, squash bugs, and create interesting visual effects. In an upcoming tutorial, I'll describe how you can use negative margins to create a columnar CSS layout that is resistant to IE's tendency to drop content next to floated columns and that features proper source order.

You can download an XHTML file containing the two examples above in the support files for this article.

Keywords

negative margins, margining, overlapping, overflowing, 3px bug, three pixel bug

All content ©CommunityMX 2002-2004. All rights reserved.