# Cross-Browser Inline-Block

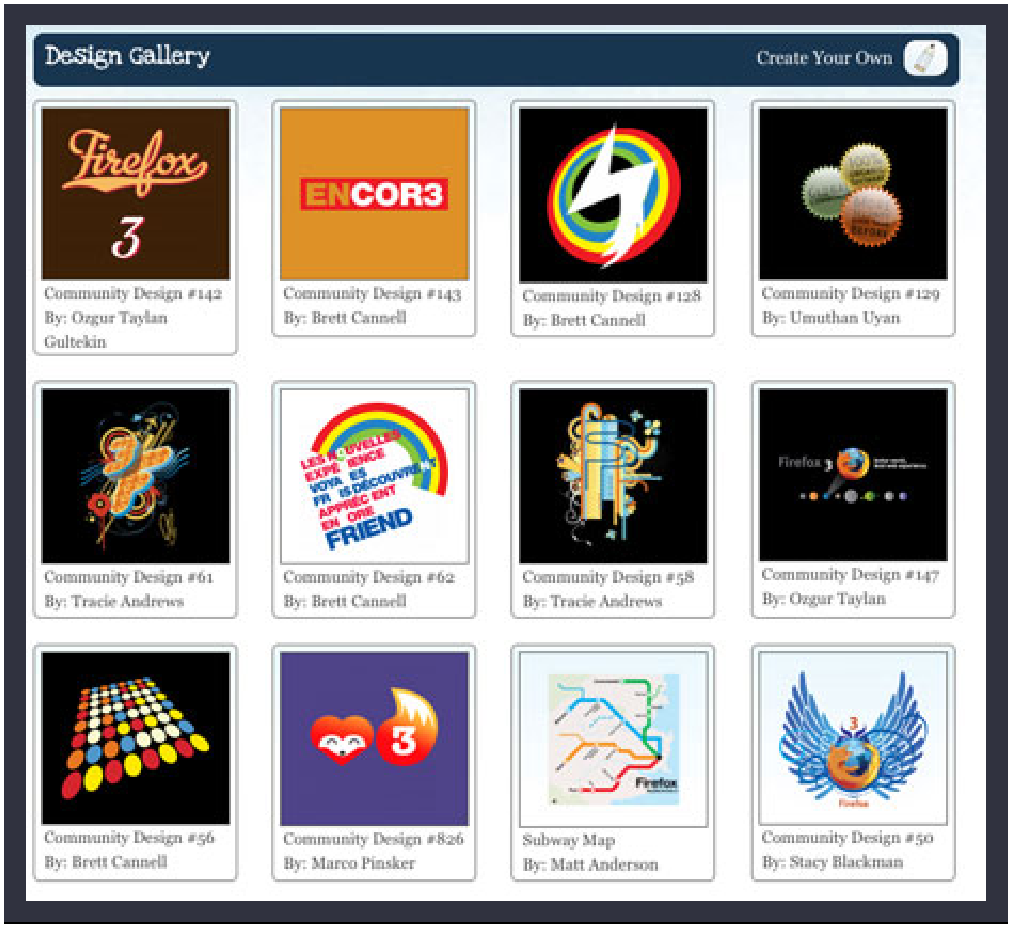
Ryan Doherty

94

Feb 20 2009

Share This

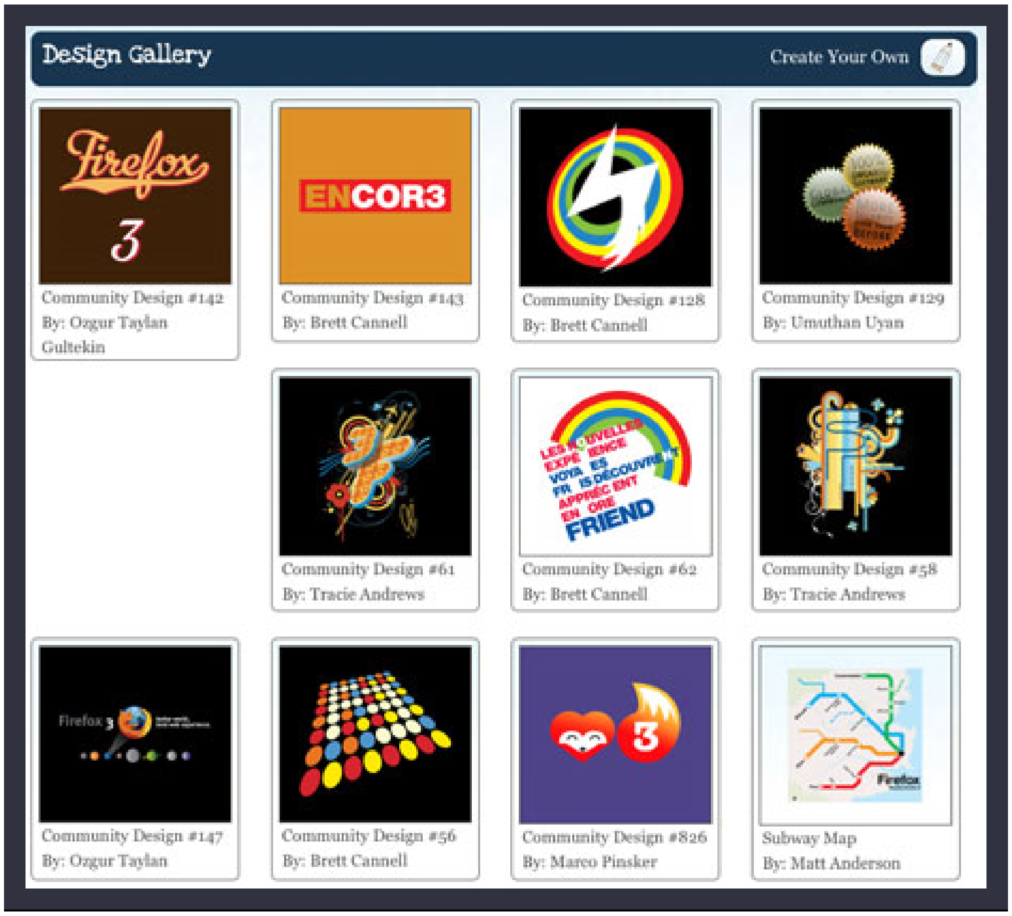
Ah, inline-block, that elusive and oh so tempting display declaration that promises so much, yet delivers so little. Too many times have I received PSD files like this:



Design Gallery
Create Your Own
Community Design #142 By: Ozgur Taylan Gultekin
Community Design #143 By: Brett Cannell
Community Design #128 By: Brett Cannell
Community Design #129 By: Umuthan Uyan
Community Design #61 By: Tracie Andrews
Community Design #62 By: Brett Cannell
Community Design #58 By: Tracie Andrews
Community Design #147 By: Ozgur Taylan
Community Design #56 By: Brett Cannell
Community Design #826 By: Marco Pinsker
Subway Map By: Matt Anderson
Community Design #50 By: Stacy Blackman

and begin to cry.

Normally, this type of layout would be a cakewalk. Fixed width, fixed height, float:left and you're done. Buuuuut, the design needs to work with variable amounts of content, which means if one of these blocks has more content than the others, it will break the layout:



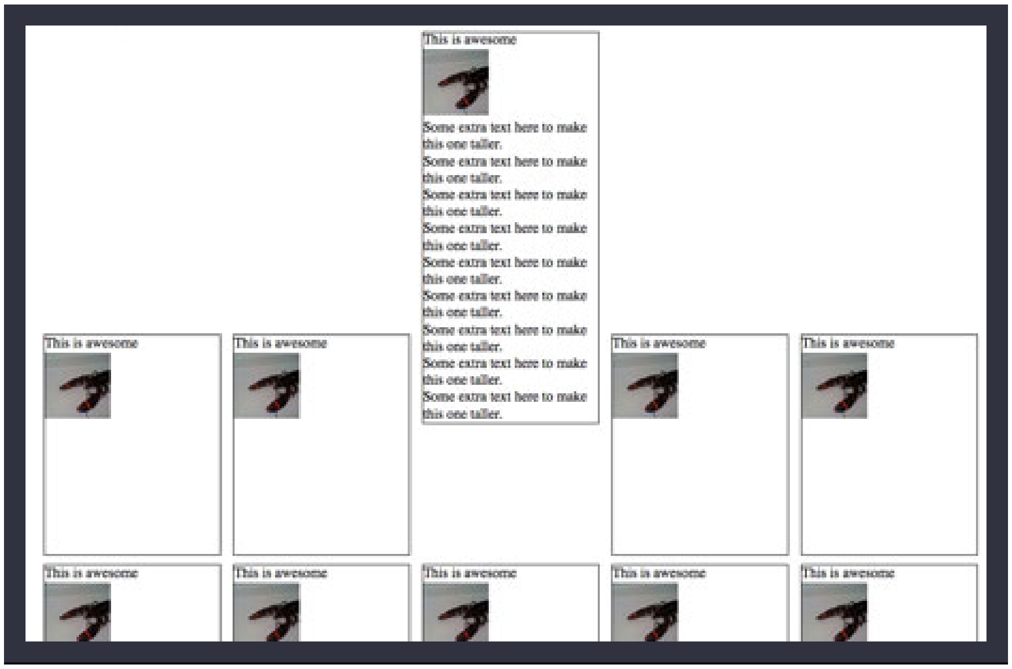
Because the first gallery item is taller than the rest, the 5th item is floated left against it instead of below it. Basically we want a layout with the flexibility of a table, but proper, semantic markup.

We start with a simple page with an unordered list and display set to inline-block:

```
<ul>
    <li>
        <h4>This is awesome</h4>
        <img src="http://farm4.static.flickr.com/3623/3279671785_d1f2e665b6_s.jpg"
        alt="lobster" width="75" height="75"/>
    </li>
...
<ul>

<style>
    li {
        width: 200px;
        min-height: 250px;
        border: 1px solid #000;
        display: inline-block;
        margin: 5px;
    }
</style>
```

And it looks *ok* in Firefox 3, Safari 3 and Opera:



This is awesome

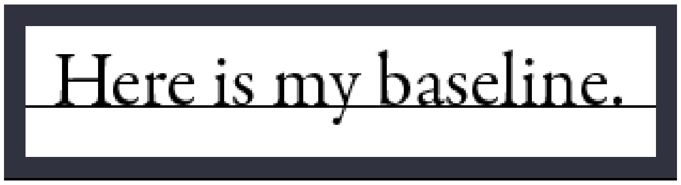
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.

This is awesome
This is awesome
This is awesome
This is awesome

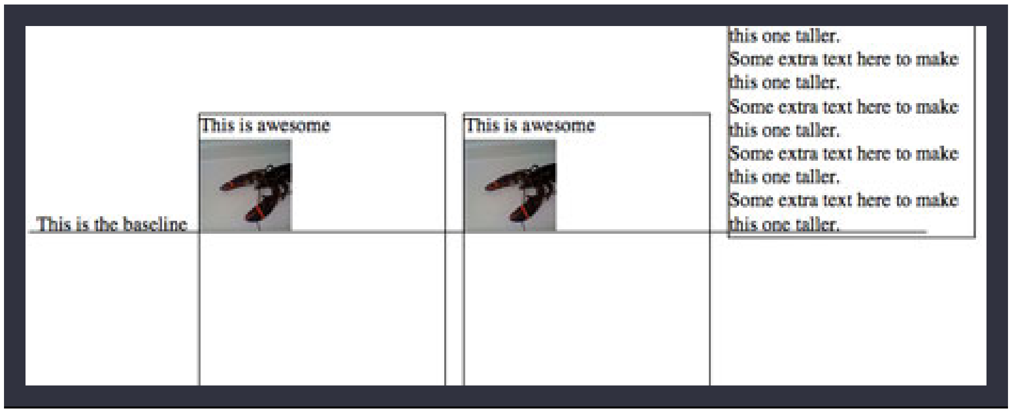
This is awesome
This is awesome
This is awesome
This is awesome
This is awesome

Obviously, something is wrong with the vertical alignment. Well, not exactly wrong, because this is the correct behavior, but it's not what we want.

What's going on here is the baseline of each <li> is being aligned with the baseline of the parent <ul>. What's a baseline, you ask? A picture is worth a thousand words:

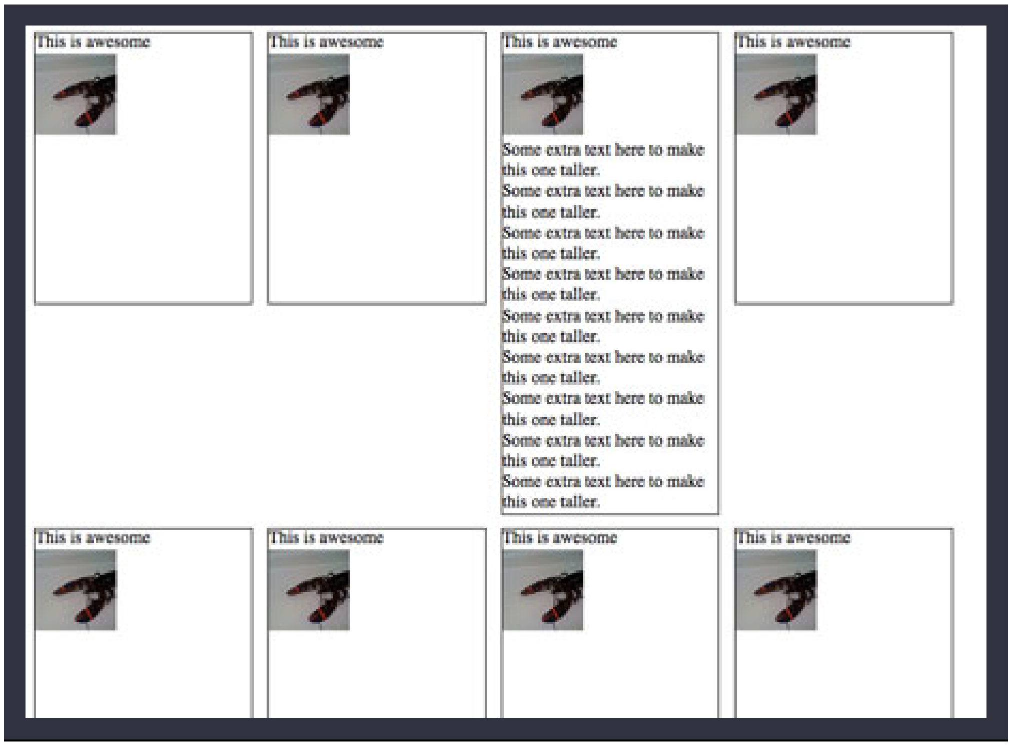


The baseline is the black line running through the text above. Putting it as simply as possible, the default vertical-align value on inline or inline-block element is baseline, which means the element's baseline will be aligned with its parent's baseline. Here's the first inline-block attempt with baselines shown:

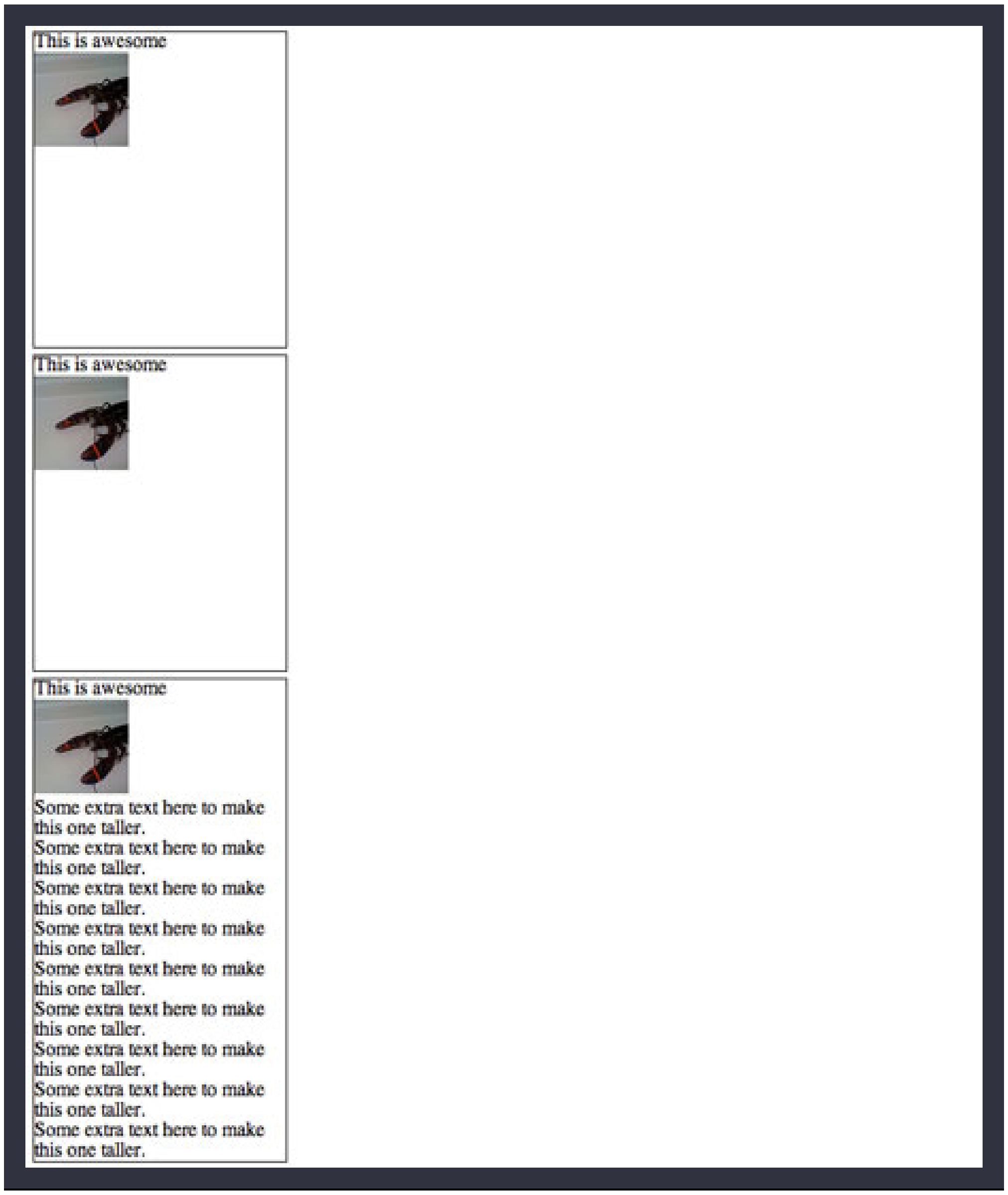


As you can see, each baseline is aligned with the baseline for the text 'This is the baseline'. That text is **not** in a <li>, but simply a text node of the parent <ul>, to illustrate where the parent's baseline is.

Anyway, the fix for this is simple: vertical-align:top, which results in a great looking grid:



Except it still doesn't work in Firefox 2, IE 6 and 7.



This is awesome

This is awesome

This is awesome

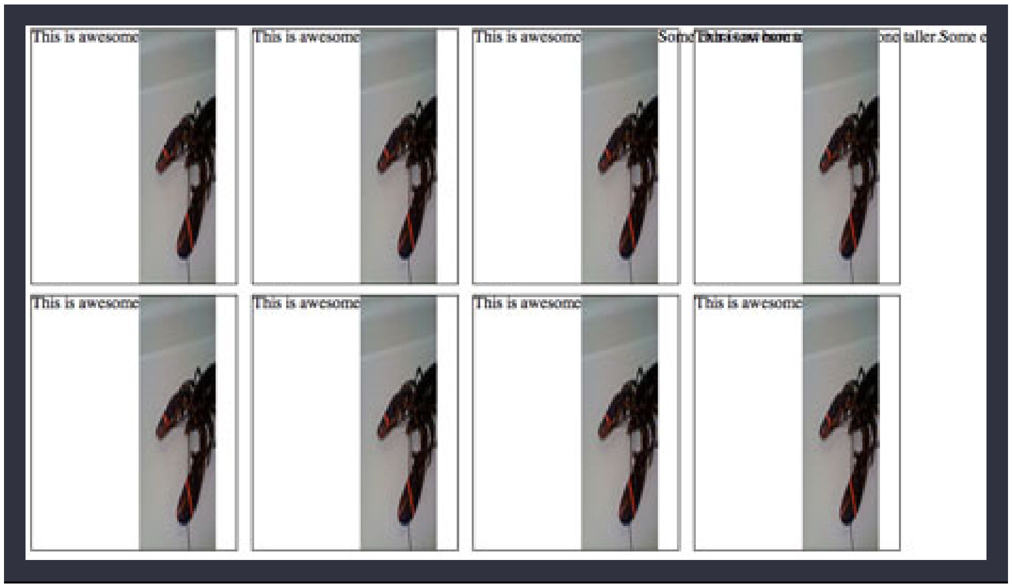
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.
Some extra text here to make
this one taller.

Let’s start with Firefox 2.

Firefox 2 doesn’t support inline-block, but it does support a Mozilla specific display property ‘-moz-inline-stack’, which displays just like inline-block. And when we add it before display:inline-block, FF2 ignores that declaration and keeps -moz-inline-stack because it doesn’t support inline-block. Browsers that support inline-block will use it and ignore previous display property.

```
<style>
    li {
        width: 200px;
        min-height: 250px;
        border: 1px solid #000;
        display: -moz-inline-stack;
        display: inline-block;
        vertical-align: top;
        margin: 5px;
    }
</style>
```

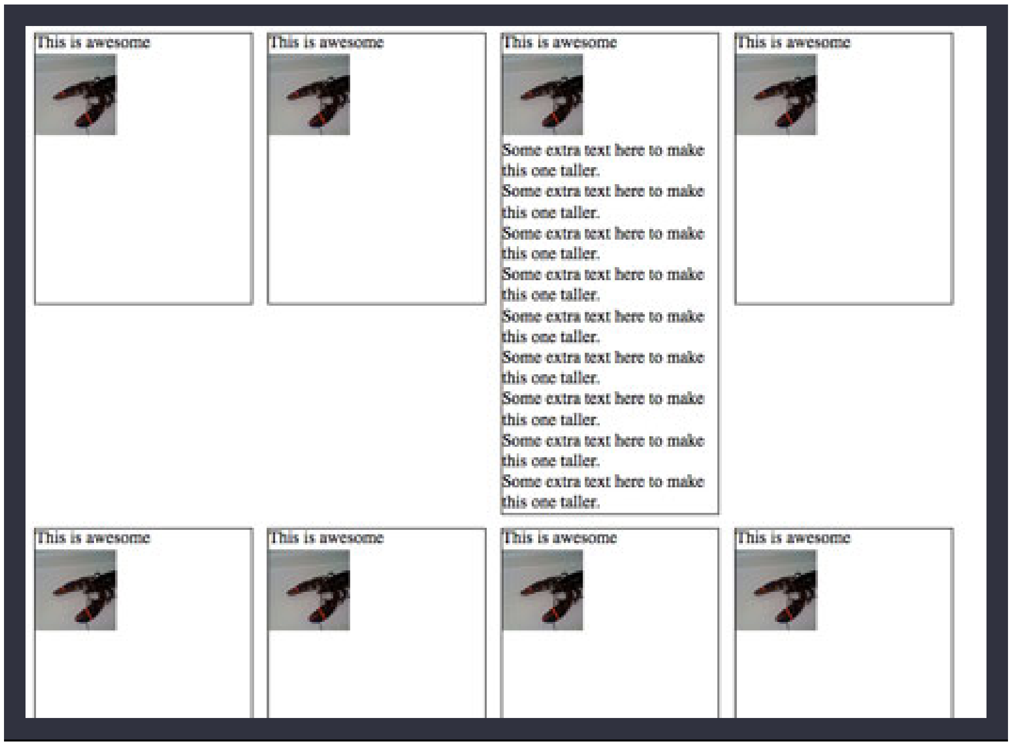
Unfortunately, it has a small bug:



Honestly, I don’t know what causes this bug. But there is quick fix. Wrap everything inside the <li> with a <div>.

```
<li>
        <div>
            <h4>This is awesome</h4>
            <img src="http://farm4.static.flickr.com/3623/3279671785_d1f2e665b6_s.jpg
            alt="lobster" width="75" height="75"/>
        </div>
</li>
```

This seems to ‘reset’ everything inside the <li>’s and makes them display appropriately.


This is awesome

This is awesome

This is awesome

Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller. Some extra text here to make this one taller.

This is awesome

This is awesome

This is awesome

This is awesome

This is awesome

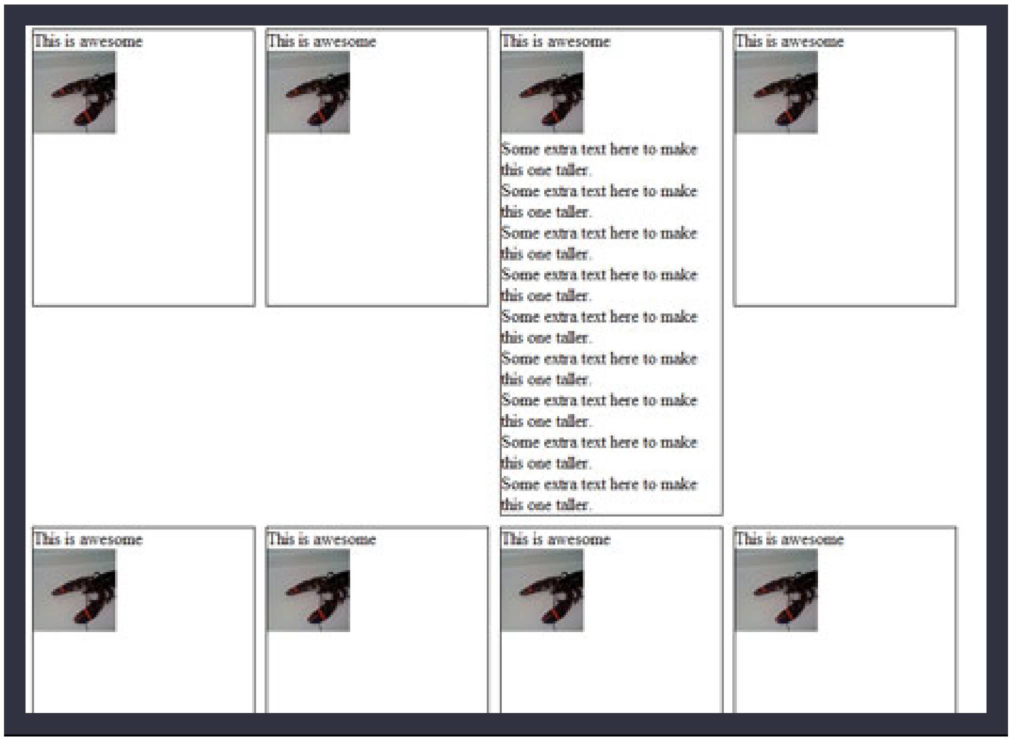
Now, on to IE 7. IE 7 does not support inline-block, but we can trick it into rendering the <li>s as if they were inline-block. How? *hasLayout*, a magical property of IE that allows for all sorts of fun! You can't set hasLayout explicity on an element with hasLayout:true; or anything easy like that, but you can trigger it with other declarations like zoom:1.

Technically, what hasLayout means is an element with hasLayout set to true is responsible for rendering itself and its children (combine that with a min-height and width, and you get something very similar to display:block). It's kinda like magical fairy dust you can sprinkle on rendering issues and make them disappear.

When we add zoom:1 and *display:inline (star hack to target IE6 & 7) to the <li>s, we make IE 7 display them as if they were inline-block:

```
<style>
    li {
        width: 200px;
        min-height: 250px;
        border: 1px solid #000;
        display: -moz-inline-stack;
        display: inline-block;
        vertical-align: top;
        margin: 5px;
        zoom: 1;
        *display: inline;
    }
```

```
</style>
```

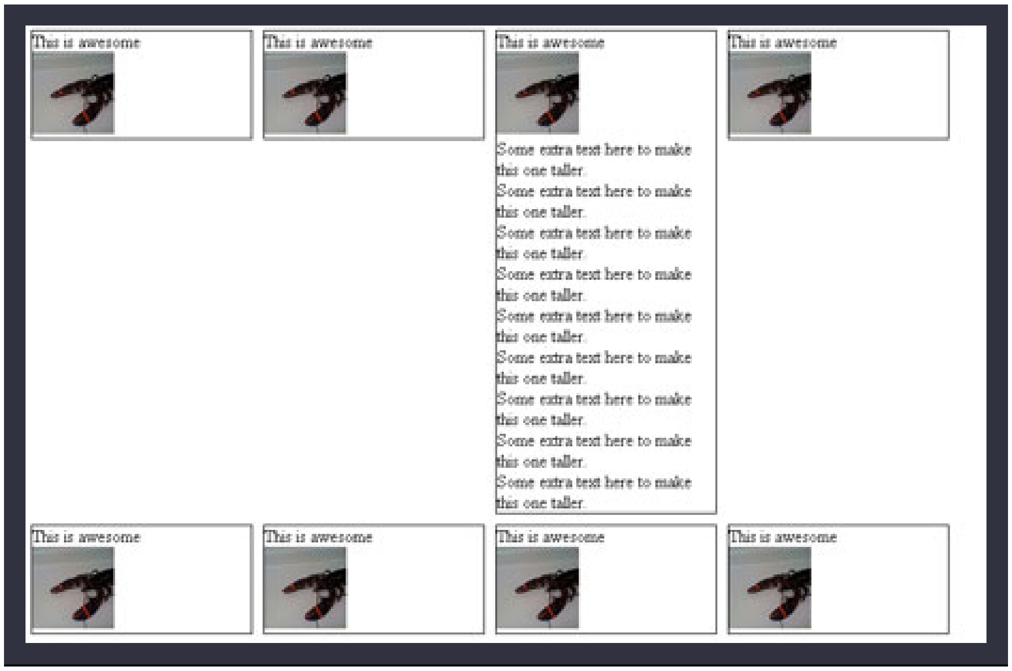


This is awesome
This is awesome
This is awesome
This is awesome

Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.

This is awesome
This is awesome
This is awesome
This is awesome

Phew! Almost done. Just IE 6 left:



This is awesome
This is awesome
This is awesome
This is awesome

Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.
Some extra text here to make this one taller.

This is awesome
This is awesome
This is awesome
This is awesome

IE 6 doesn't support min-height, but thanks to its improper handling of the height property, we can use that instead. Setting _height (IE6 underscore hack) to 250px will give all <li>s a height of 250px, and if their content is bigger than that, they will expand to fit. All other browsers will ignore _height.

So after all that work, here's the final CSS and HTML:

```
<style>
    li {
        width: 200px;
        min-height: 250px;
        border: 1px solid #000;
        display: -moz-inline-stack;
        display: inline-block;
        vertical-align: top;
        margin: 5px;
        zoom: 1;
        *display: inline;
        _height: 250px;
    }
</style>

<li>
        <div>
            <h4>This is awesome</h4>
            <img src="http://farm4.static.flickr.com/3623/3279671785_d1f2e665b6_s.jpg
            alt="lobster" width="75" height="75"/>
        </div>
</li>
```