

Tables Be Gone! Issues Involved in Using CSS Instead of Tables For Layout

R. Scott Granneman

scott@granneman.com

www.granneman.com

© 2004 Scott Granneman

Last updated 20040623

You are free to use this work, with certain restrictions.
For full licensing information, please see the last slide/page.

CSS is not something a lot of people think about.

But let's think about it!

Yes,
you should use CSS instead of tables
whenever possible.

Of course, sometimes you need tables.

When?

1. Tabular data
2. Older browsers that do not support CSS
(pre-IE 5, pre-Netscape 6.2)

Which brings up a good point:
what browsers are "good"
when it comes to CSS & standards support?

Mozilla 1+, Firefox, & Netscape 6.2+
(Win, Mac, Lin)

IE 5+ (Mac)

IE 5.5 & 6 (Win)

Safari 1+ (Mac)

Konqueror 3+ (Lin)

I dearly hope you don't have to support
Netscape 4 or IE 4.

IE 5 is doable, but still pretty sucky
in the standards department.

For your code to work correctly,
& for your modern browsers
to behave consistently,
you must declare your DTD
(Document Type Definition)
in the first line of your code.

A DTD describes the allowed legal

structure,
elements,
attributes, &
values

of a markup language
(HTML, XHTML, XML)
that complies to the DTD.

HTML 4.01 DTDs:

Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

& Frameset :(

XHTML 1.0 DTDs:

Transitional

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

Strict

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
```

& Frameset :(

If you use XHTML 1.0 Strict,
note that you must also declare the namespace:

```
<html  
xmlns="http://www.w3.org/1999/xhtml">
```

If you do not use a DTD,
OR
if you use a DTD incorrectly
(leaving out the URL at the end, for instance),

THEN

your up-to-date IE or Mozilla browser
goes into "quirks" mode.

It now acts like version 4 of those browsers.

This is bad.

For more on DTDs, see:

[http://www.htmlhelp.com/tools/validator/
doctype.html](http://www.htmlhelp.com/tools/validator/doctype.html)

<http://www.alistapart.com/articles/doctype/>

To really understand CSS,
especially CSS used for positioning,
you've got to grasp the Box Model.

The Box Model allows you to specify & position block-level elements like `p`, `blockquote`, `table`, & `div`.

As part of positioning, you can specify `width` & `height`.

But what is `width`, exactly?

Given this:

```
div {  
width:200px;  
padding:10px;  
border:5px solid #000;  
margin:10px;  
}
```

width is defined by the W3C as

"box content"

(whatever is inside the padding;

if no padding, whatever is inside the borders)

Given this:

```
div {  
width:200px;  
padding:10px;  
border:5px solid #000;  
margin:10px;  
}
```

the *total* width of the box would be

left margin + left border + left padding
+ box content (the `w i d t h` property-value)
+ right padding + right border + right margin

= 10 + 5 + 10 + 200 + 10 + 5 + 10 = 250px.

Unfortunately, IE 5 & 5.5 broke the box model.

IE 6 works correctly
if
it's not in "quirks" mode
(remember the DTD?).

So how did IE break the box model?

Given this:

```
div {  
width:200px;  
padding:10px;  
border:5px solid #000;  
margin:10px;  
}
```

width is defined by IE as

left border + left padding + "box content"
+ right padding + right border

Given this:

```
div {  
width:200px;  
padding:10px;  
border:5px solid #000;  
margin:10px;  
}
```

the *total* width of the box in IE would be

left margin + left border + left padding + box content
(which is `width` - left & right padding - left & right border)
+ right padding + right border + right margin

$$\begin{aligned} &= 10 + 5 + 10 + (200 - 5 - 10 - 10 - 5) + 10 + 5 + 10 \\ &= 220\text{px (instead of 250px!)} \end{aligned}$$

By the way,
the same disparity
between the CSS spec & IE's "math"
holds true for the height property as well.

OK, so how do we fix this?

Either

(a) wait for IE 7 to be released
& for all prior IE 5, 5.5, & 6 users to upgrade
(don't hold your breath),

or

(b) take advantage of various "hacks" in browsers
to work around it.

Hack #1: The famous Tantek hack
takes advantage of a parsing bug in IE 5 & 5.5.

<http://www.tantek.com/CSS/Examples/boxmodelhack.html>

```
div.content {  
border:1px solid;  
padding:30px;  
}
```

```
div.content {  
width:400px;  
voice-family: "\}\"";  
voice-family:inherit;  
width:300px;  
}
```

Don't use the Tantek hack.

It breaks Opera, requiring another hack
(discussed on the same page as the Tantek hack).

It causes Netscape 4 to ignore all CSS.

Hack #2: The Tan hack.

```
div {  
width:200px;  
padding:10px;  
border:5px solid #000;  
margin:10px;  
}
```

```
* html div {  
width:230px;  
w\idth:200px;  
}
```

Let me explain.

* = universal selector, so * html div means,
any div inside html inside *.

html is the root element, except in IE,
which believes there is a wrapper around html.
(*sigh*)

All other browsers, therefore,
will ignore the selector * html div,
so the CSS following is ignored by all except IE.

```
* html div {  
width:230px;  
w\idth:200px;  
}
```

IE 5 & 5.5 see * html div,
so they apply 230px to the width
(230 - 5 - 10 - 10 - 5 = 200px, what we really want).

IE 6 & IE 5 for Mac do the box model correctly, remember.

IE 5x doesn't understand properties
with escaped characters (the "\\") in them,
but IE 6 & IE 5 for Mac do.

w\idth:200px is for IE 6 & IE 5 for Mac.

Some notes about the Tan hack:

The escape (\) must be in the property name.

The escape must *not* come before these letters:

a, b, c, d, e, or f

(or it will be interpreted as a hex character & ignored).

<http://www.positioniseverything.net/articles/box-model.html>

Ah, Internet Explorer.

The browser that gives Web developers fits.

Unfortunately,
we're going to have to deal with IE 6
(released October 2001)

– & its depraved ancestors,
IE 5.5 (released July 2000)
& IE 5 (released March 1999) –
for years.

On 7 May 2003,
MSFT announced that IE 6 will be the last
standalone version of the browser
that Windows users
can download & upgrade separately from the OS.

Want IE 7?

You've got to upgrade your entire OS
to "Longhorn"
(due late 2006 ... at the earliest).

http://en.wikipedia.org/wiki/Internet_Explorer

In June 2003,
MSFT announced that it was ceasing work
on further versions of IE for the Mac.

IE 5 (released 26 March 2000) is the last one, ever.

You all know the rate at which users upgrade their operating systems.

We're going to be dealing with IE 5, 5.5, & 6 for a loooooong time.

That means you'd better get used to IE's little "oddities" when it comes to CSS & how to fix (or "hack") them (not to mention simple HTML, but that's another time).

Another infamous IE "issue": the "double margin" bug

IE 5, 5.5, & 6

Margins get doubled
when applied to floats
that touch the left or right edge
of the browser viewport.

Results?

100px margin becomes a 200px margin.

5% margin becomes a 10% margin.

This code:

```
<div id="leftColumn"></div>  
<div id="centerColumn"></div>  
<div id="rightColumn"></div>
```

```
#leftColumn {  
float:left;  
width:48%;  
margin-left:3%;  
}
```

does this:

[Back](#)





[< Back](#)

Solution #1:

Remove `margin-left` from `leftColumn` id.
Float small `<div>` before the left column,
with same `width` as `margin-left`.

```
<div id="gutter"></div>  
<div id="leftColumn"></div>
```

```
#gutter {  
  float:left;  
  width:3%;  
  height:1px;  
}
```

```
#leftColumn {  
  float:left;  
  width:48%;  
}
```

Solution #2:

Add `display:inline` to the `leftColumn` id.


```
<div id="leftColumn"></div>  
<div id="centerColumn"></div>  
<div id="rightColumn"></div>
```

```
#leftColumn {  
  float:left;  
  width:48%;  
  margin-left:3%;  
  display:inline;  
}
```

Yes, this is totally bizarre.
According to the W3C's CSS spec
(<http://www.w3.org/TR/CSS2/visuren.html#floats>),
any `display` except `none`
is supposed to be ignored on a floated item.

All other browsers ignore the `display`
& `float` as before,
including IE!

But for some reason,
`display:inline` tells IE
not to double the margin.

Collections of IE CSS bugs & workarounds:

Explorer Exposed!

<http://www.positioniseverything.net/explorer.html>

CSS Bugs and Workarounds

<http://css.nu/pointers/bugs-ie.html>

CSS Bugs in IE5.x Mac

<http://www.macedition.com/cb/ie5macbugs/>

Just to be fair ...

a collection of Gecko-based bugs & workarounds:

Open Source Oopsies

<http://www.positioniseverything.net/gecko.html>

OK, so how do we create layouts using CSS instead of tables?

Easy – we steal ... uh, copy ... uh, borrow.

Lots of Web sites provide CSS-based layouts – just make your choice & copy the source.

2 columns?

Menu on the left

<http://bluerobot.com/web/layouts/layout1.html>

Menu on the right

<http://bluerobot.com/web/layouts/layout2.html>

2 columns, liquid width

<http://nemesis1.f2o.org/articles>

3 columns?

Center fluid

<http://glish.com/css/7.asp>

All fluid

<http://glish.com/css/2.asp>

All static width & all centered

<http://glish.com/css/3.asp>

4 columns?

(Not sure why you'd want to ...)

All fluid

<http://glish.com/css/8.asp>

Collections to view & grab:

Little boxes

http://www.thenoodleincident.com/tutorials/box_lesson/boxes.html

Sample CSS Page Layouts

http://www.maxdesign.com.au/presentation/page_layouts/index.cfm

Layout-o-matic

<http://www.inknoise.com/experimental/layoutomatic.php>

The best 3-column,
fixed-width,
CSS-only,
works-in-all-modern-browsers
layout I know can be found at
[http://www.positioniseverything.net/piefecta-
rigid.html](http://www.positioniseverything.net/piefecta-rigid.html)

Works in IE 5+ (Win), IE 5 (Mac),
Mozilla (Win, Mac, Lin),
Safari (Mac),
& even Opera 6+!

View the source & comprehend the code at
[http://www.positioniseverything.net/
piefecta-rigid.html](http://www.positioniseverything.net/piefecta-rigid.html)

Automatically generate the code at
[http://www.cornerstonechurches.org.uk/
ctindex.php](http://www.cornerstonechurches.org.uk/ctindex.php)

Your homework:

learn all you can about the `float` property.

<http://www.maxdesign.com.au/presentation/butterfly/>
"Float like a butterfly" – a tutorial

<http://css.maxdesign.com.au/floatutorial/>
"Floatutorial"

<http://www.positioniseverything.net/articles/float-theory.html>

"Float: The Theory" – much deeper than a tutorial

Where can you find more great CSS info?

[http://www.granneman.com/usefullinks/
development/coding/css/](http://www.granneman.com/usefullinks/development/coding/css/)

My collection of CSS resources.

<http://www.granneman.com/teaching/wuweb/>
Syllabus for my Wash U CSS course.

<http://www.positioniseverything.net>
Position is Everything (I bow before them)

<http://www.maxdesign.com.au>
MaxDesign

<http://www.alistapart.com/>
A List Apart

Yes,
you should use CSS instead of tables
whenever possible.

Thank you!

Email me: scott@granneman.com

Visit my Web site: www.granneman.com



Licensing of this work

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

In addition to the rights and restrictions common to all Creative Commons licenses, the Attribution-ShareAlike License features the following key conditions:

Attribution. The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original author credit.

Share Alike. The licensor permits others to distribute derivative works under a license identical to the one that governs the licensor's work.

Questions? Email scott@granneman.com