

The Personal Disquiet of
MARK BOULTON

Five simple steps to designing grid systems - Part 3

– August 9th, 2005 –

The third installment to this series is going to be a little different. The previous installments have been talking through some of the basics of grid construction using ratios as the primary device. They've also dealt with grid construction for print media. Unfortunately, as designers for web media, we don't have some of the luxuries as our print designer colleagues.

Rather than go through tutorials (I'll be covering these in the last two installments), I'll be using this installment as a platform to discuss some of the challenges and rewards of designing grid systems for the web.

A whole load of considerations

Designing grid systems for print is considerably more straight forward than designing grid systems for the web. First off, in print, the designer has a *fixed* media size - the paper size (or packaging, poster, whatever). Let's say a print designer has designed a magazine. The reader of this magazine can't suddenly increase the font size if they find it difficult to read - well they just move it closer to their eyes I guess. This is just one consideration, there are more but I'm sure you get the point.

So, that's media size. On the web you have other considerations such as the browser, the OS platform, the screen size and the actual devices that web sites can be viewed on, from PDA's and

Mobile's to assistive technologies such as screen readers. How do you design a grid for all of this? It's a really good question and I'm not claiming I have the answer.

In an ideal world

We all know about the problems with websites rendering differently across different browsers, platforms, devices etc. But, just for a moment, let's forget about that.

Designing a grid for the web should not be difficult, in fact, it shouldn't be any different from designing a grid for an media. As discussed in the previous parts of the series, you can construct a grid in the same manner for screen as you do for print. Base it on ratios, experiment with form and white space etc. Use pixels as your base measurement and go from there in the knowledge that your design will look exactly the same in every browser. After all, you, the designer, knows best for your reader right? You *know* they want light grey, 10px verdana with a measure of 600 pixels.

In the real world

Good designers for the web *know* that the users who use their sites may want different, and know, with the web, they have the power to change things. The designer has lost, to a degree, the ability to *control*. For a lot of designers (including me), this has been a tough transition. We're taught for years to create the delicate balances of white space, the manicured typography and delicate colour palettes, all designed to create harmonious designs which do their job very well.

Then some short sighted user comes along and increases the text size... and... and... totally breaks *your* design.

I think you get the idea. We can't be upset when the user wants to change something like the text size. What we can do is design grid systems to *adapt* to those changes.

Not just columns

Over the past couple of years, coupled with the increase in CSS based sites, we've seen a rise in certain grid configurations which are all based on the amount of columns. 2, 3, 4 column layouts - float this, position that etc. Why, even this site falls into the '750 px, 3 column' category. These grids have quickly become a convention, and for good reason too. They are quick to create, fairly stable across many platforms and don't degrade to the same degree as table based layouts. This is all good. What isn't so good though is the general lack of understanding of grid systems when perhaps the question on most designers lips when they sit down to begin a design is, 'how many columns should I have'. This is not grid system design.

Grid systems for the web

The next two installments of this series will go through details of creating considered grid systems for the web and the implementation using CSS. I thought it would be useful to just go through some of the considerations before hand.

The series

This is the third installment of this "Simple Steps..." series.

1. Subdividing ratios
2. Ratios and complex grid systems
3. Grid systems for web design: Part 1
4. Grid systems for web design: Part 2 Fixed
5. Grid systems for web design: Part 3 Fluid

Filed in: [articles](#), [design](#), [simple-steps](#), [typography](#).

Further reading

My Handbook – Environment – June 5th, 2014

Collaborative Moodboards – April 1st, 2014

Responsive Web Design – Defining The Damn Thing – March 10th, 2014

Recently: [Green n Blue - An EE control panel theme](#) Previously: [Professional CSS: A first look](#)

SEARCH THE SITE

ELSEWHERE

I work for [Monotype](#). Previously I founded [Mark Boulton Design](#) and co-founded [Five Simple Steps](#). You can follow me on [Twitter](#).

Browse the archives by date:

[2014](#), [2013](#), [2012](#), [2011](#), [2010](#), [2009](#), [2008](#),
[2007](#), [2006](#), [2005](#), [2004](#), [2003](#)

THANKS TO

BUY MY BOOK!



[13 Of Them Are Free! Prices Are Starting From \\$55, Free T-shirt Included In The](#)



[A Practical Guide to Designing for the Web](#)

Design your website using the principles of graphic design.

[Pirate Club Plan](#)
[Powered by Fusion](#)

Copyright © 1999 – 2014 [Mark Boulton](#). [About](#), [Contact](#), [Journal](#), [Speaking](#). Typeset in [Amasis eText](#), [Clarendon Text](#) and [Mundo Sans](#) from [Fonts.com](#). Designed using an [Apple Mac](#) in [Wales](#). Built using [Gridset](#), [Statamic](#) and [Espresso](#). Hosted with [Media Temple](#).