

## UI Design using CSS Grid Frameworks

[UPDATED 21/2/2012]

Grids have been used in print publishing to layout pages. They divide the page into horizontal and vertical consistent parts where text, headlines, images, and advertising can be placed.

There are advantages and disadvantages to using grids. If you use grids, you typically have to include extra HTML tag markup to your content, thus increasing page file sizes. This also means that markup and style are not truly separated and that there is no semantic benefit from the addition.

However, there are a number of design benefits to using grid frameworks - The Grid can be used to achieve many different layouts:

- You can divide up space for major regions of an application, or
- You can define the relationship in terms of size, position, and layers.

Like tables, grids enable an author to align elements into columns and rows, but unlike tables, grids don't have content structure, and thus enables a wide variety of layouts not possible with tables.

### Developing with grids:

1. The 1140 CSS Grid <http://cssgrid.net/> can be used to intelligently reflow elements into 12 columns. On smaller monitors, it becomes fluid and adapts to the width of the browser. Once it gets very small (e.g. for mobile) it will stack all of the columns on top of each other. Or into numerous uneven combinations like seven and five or five, four, three. Go to [cssgrid.net](http://cssgrid.net) and read the first page for a better understanding of what grids are and how this one works.

*Don't skip the bottom part of the article which describes the markup and the explanation of classes & markup – very important!*

2. Next click the "Download" button on the 1140 grid page to save the templates and demo to your desktop.



3. Copy the index.html and edit it so that you have a <div> with a container, row, and multiple columns like the example below. 1140.css controls the rows and cols.

```
<div class="container">
  <div class="row">
    <div class="threecol">
      <p>Column 1</p>
    </div>
    <div class="threecol">
      <p>Column 2</p>
    </div>
    <div class="threecol">
      <p>Column 3</p>
    </div>
    <div class="threecol last">
      <p>Column 4</p>
    </div>
  </div>
</div>
```

4. Create a row that spans across all 12 columns. This example is not in index.html – but there is a CSS rule in 1140.css already. Find the right rule and implement it.
5. Now edit it so that there are two rows, which each span across all 12 columns, followed by one row with three columns. Close it off with one more row spanning all 12 columns. The result is something like this:



**The tricky part is remembering that the class of the div you use is how many columns it spans, not how many show up on the page! (this is why I have said I want three columns across but my columns above say “Four”).**

- You are going to recreate a news website – these types of sites often have numerous entry points for large amounts of information. A grid can be a useful tool to get you started when designing this complex interface. Recreate a grid similar to below after taking a look at <http://www.rte.ie/news/> It doesn't have to be an exact replica, instead use this as inspiration!



- Next, style the page using your CSS. The boxes are currently just `<p>` paragraphs inside each div – this won't work for your page as we want to include other tags such as `<h1>` headings. You can take content from <http://www.rte.ie/news/> to fill in your page. Once you have some sample content change the style of the boxes.

If it helps you can look at the source & CSS of RTE, but remember they will have their own proprietary classes, grids and CSS so if you try to use the same classes they will likely not work for you.

## **FURTHER READING:**

If you've made it this far, here are some additional exercises & reading about grids.

“Oh Yeeahh” by Khoi Vinh, former designer of the New York Times online.  
<http://www.subtraction.com/2007/03/19/oh-yeeeahh>

Grids are Good by Khoi Vinh (again) and Mark Boulton – excellent presentation on why you need to use grids.  
[http://www.subtraction.com/pics/0703/grids\\_are\\_good.pdf](http://www.subtraction.com/pics/0703/grids_are_good.pdf)

Grids: Order out of Chaos – brief description of things like alleys, gutters & margins. [http://desktoppub.about.com/od/grids/l/aa\\_gridorder.htm](http://desktoppub.about.com/od/grids/l/aa_gridorder.htm)

Mark Boulton's “Five Simple Steps to Designing Grids” articles (very good!)  
<http://www.markboulton.co.uk/journal/comments/five-simple-steps-to-designing-grid-systems-part-1>

Layout Grid Bookmarklet by Andy Budd – overlays a grid on any webpage.  
[http://www.andybudd.com/archives/2006/07/layout\\_grid\\_bookmarklet/](http://www.andybudd.com/archives/2006/07/layout_grid_bookmarklet/)

Free online graph paper you can print out.  
<http://www.incompetech.com/graphpaper/>

***Further References:***

CSS3 Grid Layout Specification (This is still at the experimental phase)

<http://dev.w3.org/csswg/css3-grid-align/>

YUI CSS Grids <http://yuilibrary.com/yui/docs/cssgrids/>

1140 CSS Grid <http://cssgrid.net/>

960 CSS Grid <http://960.gs>

Adapt.js – Responsive adaptive CSS <http://adapt.960.gs/>

SASS <http://sass-lang.com/>

LESS <http://lesscss.org/>

CSS Easy <http://csseasy.com/>