**Blog Challenge**

**Rationale**

This challenge is an attempt to build a second ‘real-world’ site that demonstrates the first foray in to a non-linear layout. It allows students to practice the skills (and experience the challenges!) of CSS positioning for even a simple layout. It can also be revisited later in the course as an example for re-working with both Bootstrap Grid and flex box, time permitting.

**Placement**

After the display property and floats in positioning, but before the position property (absolute/fixed not needed here). Ideally this should replace the current ‘floating layout’ challenge.

**Execution**

Should be run as a challenge that is taken up together. Potential to split it up in to taking up the header first, then the main section.

**Key objectives**

* Learn how to structure a layout in to parts
* Practice what happens to blocks when they are floated and set inline
* See a clear example of what happens to content in a page when elements are floated, motivating the need for clearfix (which really can’t be understood too well from the slides alone)
* Experience the finer points of the box model as they pertain to layouts, i.e. elements that sum to 100% don’t fit when borders/margin/padding are applied. ***This is why there is a one-pixel border to the right of the side nav, to overflow.***
* Explain page-level containers as they pertain to fixed-width layouts (using width and margin:auto), with some mention of the fact that we’ll be able to spin them up easier soon (bootstrap)

**Additional items**

* Though not meant to be responsive, this exercise can be used to demonstrate why we’ll need to learn some additional skills to work with different screen sizes.
* Use of headings given SEO (i.e. why to use <h1> for the article title and not the site title here)

**Instructor Note**s

I suggest using floats in the header and only inline-blocks with set widths in the main section to show both methods. Let them run in to the problem of the divs spilling over the 100% mark when they add the border. You’d also have to address the one character of whitespace issue, as well as vertical-alignment.

There are several ways to overcome this:

* Put the border on the content (i.e. sidebar nav) instead of the sidebar container
* Adjust the box-sizing property
* Use outline instead of border
* Use floats
* Likely a few others…

Feel free to review one or more methods depending on time and progress, whatever best exhibits the box model’s ins and outs.