

Jay Fallon

## CSS Layout Reflection

---

### Floats

I'm going to be hard-pressed to say anything positive about floats, however I do recall with distinction when I started to design layout with them, January '07 and I vividly remember when I started using Flexbox, May '15, so while they did leave a lasting impression on me, I have not looked back since and everything I've done up until this week was using Flexbox. I had always been a table person and it wasn't until I was forced to give them up that I started using floats. To begin with, Clearfix was always a hack and remains so today. There is no way to measure the development hours lost nor the levels of disappointment that many of my peers have suffered trying to figure out the mysteries of the floats. The side-effect of this was that shortcut artist became adept at using positioning, which meant that they invariably introduced the infamous z-indexes into their code to make up for their previous mistakes.

Mastery of floats did not come easy but over a period of eight years I was able to keep myself in business without looking over my shoulder too much. In short, layouts using floats should be consigned to the dustbin of history, along with z-indexes and positioning. They should not be resurrected nor should they be combined with any other approach even if the richest client in the world is begging for it. It's malpractice, plain and simple.

---

### Flexbox & Grid

How do I love thee, even if you may just be an abstraction of floats? I really enjoy using Flexbox and have done so without the benefit of frameworks like the never-to-be-released Bootstrap 4. It really took the pain out of designing dynamic layouts with a certain ease of use that came with being able to extricate yourself from making mistakes simply by commenting out `display:flex` on the parent and then re-arranging the children accordingly. However, this did lead to a lot of nested flex components, I'm guilty as charged, and if I found myself working with someone who didn't really understand the concepts of Flexbox, I'd been on my own to triage, diagnose and fix any code where it was employed. Not bad for business but not great for collaboration, either. Like floats, it was also imprecise for data display and is best employed as a grid in a row-by-row basis where you can control the flow of the design through `.wrapper>section+section+section...+section`. I've done most of my work like that and it's been very good to me.

Grid on the other hand seems like an abstraction of tables into CSS with minimal markup on the HTML side, and some powerful additions like reordering, repeats and regions as well. I really like it for that and I hope people will come to realize that it should replace tables for data and Flexbox for the immutable layout components like header, nav, main and footer. But as the name implies, it's not as flexible as Flexbox and this where I'd pair the two approaches out of simplicity, replacing the grandparent flex grid and styling the components with flex only. That being said, there is a certain preciseness with Grid that can't be achieved with Flexbox unless you're using a row-based paradigm.

Overall, I think we're going to go far with this pairing and best of all we won't need to rely on hacks or frameworks to do it. Everyday developers will be able to get pretty granular with once they've mastered the modules and it'll be something I really look forward to working with.