written by Phil Giles on 24 April 2017 in design, development process, front-end

# css grid

Last month a new set of CSS properties were added to the spec. Together these make CSS Grid, the new way to layout our pages in CSS.

CSS Grid finally gives us the control we have always wanted for layouts in CSS. Gone are the days of horrible layout workarounds and hacks to get what you want. This move also makes layout and pages much more maintainable and a lot easier to use.

Now I know what you're thinking. What about flexbox? This time last year I wrote about how flexbox can fix your layout woes. Well don't worry, you don't need to forget all of that knowledge as flexbox and grid work together really well. But for now, I'm just going to go through the basics of grid and how you can get started straight away!

## getting started

So we start with a basic layout of three divs. You can see I have added `display: grid;` to the wrapper - but nothing happens! Of course, there's some more info the grid needs to know before it can do its magic. So let's tell the grid about how we want the columns to work.

I have added two new properties to the wrapper class, so let's see what they do! Firstly I have added `grid-template-columns`. This, as expected, tells the wrappers children how to behave as columns. You may or may not be familiar with the unit used here. `1fr` is one fractional unit, which means the column will take one share of the available space for its width. As you can see I have added three of these, which means I will get three columns of equal width.

The second property I have added is `grid-column-gap`, which is kind of self-explanatory. This adds space in between each column, which in this case is `1px`. This space will show through the background of the wrapper which I have made white to make it easier to see.

## step it up

Ok so we have the very basics down so let's make a few more changes. What will happen if I add in som more column divs?

The new columns have created a new row as expected, but where is the gap? Well, the property we added in the last step only covered the column gap and these new divs have created a new row. So we need to add, yes you guessed it, `grid-row-gap`

And now we have a nice looking grid! Well this has all been fairly simple so far, but

let's mess with the columns a bit more to get some more interesting layouts.

In the above example I have made it so the first column will always be three times wider than the second and third, and as you can see in the result this flows through all columns in that wrapper. You can really start to see the potential of CSS Grids now. So let's go one further.

So I have added another three divs to the wrapper and added a new property to the CSS - `grid-template-rows`. This works in the exact same way as `grid-template-columns` where you set the desired height for each row based on the available space. So, in this case, I have set the second row to be twice the height of the first, and the third row to be three times the height of the first.

## the support

As with all new technologies in CSS the support is usually a bit shaky when it comes to browsers adding it in, and then you will find yourself adding browser prefixes meaning repeated properties. But this time around many browsers have added it straight away with no feature flags. This is great news for all of us front-enders!

Of course, IE and edge are letting the side down again with only partial support - but I'm sure it won't be too long before they add full support. You can check the current browser support to keep up to date.

## thats all folks

It doesn't take too much imagination to see how massive this will be, and all the amazing things you can create with it. And I certainly haven't even scratched the surface of layout and positioning options, which I may go into in a future post.

But for now it's great to see that CSS is starting to catch up and add more and more useful features that make our code cleaner and more maintainable. If you have anymore basic CSS Grid tips, let me know in the comments!

*If you enjoyed the read, drop us a comment below or share the article, follow us on Twitter or subscribe to our #MetaBeers newsletter. Before you go, grab a PDF of the article, and let us know if it's time we worked together.*