

Grid Layout – Part 2

Now that we had a look at how we set the grid up. Let's talk about explicitly placing items in different areas of the grid.

It's important to know the parts of the grid as we move items around on the grid. We use the lines between the columns and rows to indicate where the children will be placed. We can also place them in more than one cell (or intersection of a row and column. We can place them in an area of the grid that spans multiple rows and/or columns.

Let's look at the html we created from a wireframe. We want to use the div with the class of home-grid to be the grid container (or parent). It has 5 direct children: the video, an h1, an aside, and two icon images with. Only these direct children can be placed in the grid. Not the h2 or p (they are grandchildren) They will go along with their parent the aside wherever it's placed in the grid.

If have set up the parent div with the class of home-grid with 3 equal columns and a grid-gap of 40px, the 5 children will automatically be placed in the order they were in the HTML into the grid, with the first 3 in the first row and the remaining two children wrapping around to make a second row. You can see the grid by using the developer tools. Just rest your mouse pointer over the parent element. Notice that the aside is making the first row very tall because it has a lot of content.

But we want to make it look like our wireframe. How do we do that? I would place the rows and columns of the grid over my wireframe to see how that looks. Notice how the columns and rows and how they are numbered between them.

To place an item of the grid (or one of the children) we can use the grid-column-start and grid-column-end properties. The number value you give this property refers to the lines between the columns and rows not the columns and rows themselves. For example, we want the 'path to the beach video' and title to go from column line 1 to column line 3 and from row line 1 to row line 2.

If I want the first child to go across two columns, I will say what line number to start from and which line number to go to. So, grid-column-start: 1 and grid-column-end: 3. It starts at line 1 and goes to line 3 spanning the first 2 columns. There is also a short cut for this: grid-column: 1/3; The number before the slash is the start line and the number after the slash is the end line. The same works with grid-row, 1/2 . Notice how the other children are still implicitly just going into the available cells. We need to place them as well. Let's place the 'Path to the Beach' h1 title into the same place since in our wireframe it is going over the top of the video. Wait, where did it go? The h1 comes after the video in our html. So, the stacking order has it going behind the video. We can change that with z-index. Giving it a positive number like 1 would bring it forward. 0 is the default and if you give it a negative number it will go back and a positive number will bring it forward. There it is. We will center it in a minute. Let's place the

aside child where we want it. According to our wireframe and grid outline it will span two rows but take only one column. The column will start on line 3 and end on line 4 so we put grid-column: 3/4 and grid-row: 1/3. The last two items, the two icons implicitly took the remaining spots and it happens to be where we wanted them so we don't have to explicitly place them. Cool huh?

Now that we have them all placed. Let's center that h1 title 'Path to the Beach'. To center child items of a grid vertically (it's already centered horizontally) we can use align-self: center. We'll cover more about centering grids, grid items and individual items later.

Looks great. And trust me, as difficult as that seemed. It was way easier than trying to use floats and positioning.