

Grid Layout - Part 2

Female: Grid layout part two. Now that we had a look at how we set up the grid, Let's talk about explicitly placing items in different areas of the grid. *[A grid pops up. It has five columns and three rows.]* It's important to know the parts of the grid, as we move items around on the grid. We can 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 common row. 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.

[A new screen is shown. The screen shows a page with pictures of the beach, a wireframe with spots for pictures, and a page of HTML.]

We want to use the “div” with the class of “home-grid” to be the grid container or parent. It has five direct children, the video an H1, an aside, and 2 icon images. Only these direct children can be placed in the grid, not the H2 or the p. They are grandchildren. They will go along with their parent, the aside, wherever it's placed in the grid. If we have set up the parent “div” with the class of “home-grid” with three equal columns, and a grid gap of 40 pixels, the five children will automatically be placed in the order they were in HTML into the grid, with the first three in the first row, and the remaining two children wrapping around to make a second row. *[The five beach pictures move so there are now three on top and two on bottom.]* You can see the grid by using the developer tools. *[The screen shows the developer tools tab open.]* Just rest your mouse pointer over the parent element.

[Female hovers her mouse cursor over the <div class="home-grid"> element. This causes the grid layout from the start of the video to appear as an overlay on the webpage.]

Notice that the aside is making the first row very tall because it has a lot of content. But if we want to make it look like our wireframe, how are we going to do that? I would place the rows and columns of the grid over my wireframe to see how that looks. *[Vertical and horizontal dividing lines are now shown drawn on the wireframe.]*

Notice how the columns and rows are and how they have numbers between them. To place an item of the grid or one of the children, we can use the grid-column-start and the grid-column-end properties. *[Clicks on row 54 of HTML, then types “grid-column-start: 1;”]* The number value you give this property refers to the lines between the columns and rows, not the columns and rows themselves. *[Clicks on row 55, then types “grid-column-end: 3”]* For example, we want the path to the beach video and this 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 two columns.

There's also a shortcut for this: grid-column One slash three. The number before the slash is the start line, and the number after the slash is the nth line. *[Replaces row 54 with “grid-column: 1 / 3;”, and row 55 with “grid-row: 1 / 2;”]* The same works with grid-row: one slash two.

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's going over the top of the video.

[Types “h1 {“ into row 57, then copies and pastes rows 54 and 55 into rows 58 and 59. She reloads the webpage preview, and the title “Path to the Beach” disappears from the webpage.]

Wait, where did it go? The H1 comes after the video inner HTML. So the stacking order has to go behind the video. We can change that with the z-index, giving it a positive number, like 1, would bring it forward. *[Types*

“z-index: 1;” into row 60.] 0 is the default. And if you give it a negative number, it's going to go back. So with that positive number, it will bring it forward. So there it is. We will center it in a minute.

Let's place the “aside” child where we want it. [Types “aside {“ into row 62, and types “grid-column: 3 / 4;” into row 63.] According to our wireframe. The grid outline will spend two rows, but only take one column. The column will start on line 3 and on line 4. So he put grid-column three slash four and grid-row one slash three, spanning two rows. [Types “grid-row: 1 / 3;” into row 64.] The last two items, the two icons implicitly took the remaining spots. It happens to be where we wanted them. So we don't have to explicitly place them.

Now that we have them all placed, let's center the H1 title, “path to the beach.” To center a child item up a grid vertically—because ours is already centered horizontally—[Types “align-self: center;” in row 61.] So vertically, we're going to use “align-self: center.” We'll cover more about centering grid items later. All right, looks great. And trust me, as difficult as that seemed, it is way easier than trying to use floats and positioning.

[End of video]