Exercise Sheet 2

Exercise 1

Use grid areas to define a grid with the following structure:

Hints: *You'll have to make all grid items direct children of the grid container to assign them to grid areas (nested grids will be covered later).*

Solution: View solution on GitHub

Exercise 2

In this exercise, you'll scale things up a little and create the following grid of 6 rows and 12 evenly-spaced columns which should **evenly take up all width** in the grid container:

Hints: *Remember you can use Emmet to quickly generate 72 divs by expanding* `div*72` *– to do this, place your cursor behind the* `div*72` *and hit Ctrl+E in Atom (TAB may work as well). For the grid, you'll probably want to use the* `repeat` *function. Lastly, remember the effect of the new unit you learned in this section.*

Solution: View solution on GitHub

Exercise 3

In this exercise, you'll explore `auto-fill` and `auto-fit`. This is admittedly a trickier exercise than the ones so far but if you work through it, you'll have a great understanding of how `auto-fill` and `auto-fit` work. Make sure to use the grid visualization in Firefox for this exercise.

1. Create a grid container with a fixed width of `500px` and two rows
2. Add four grid items to it and add a `grid-gap` of `10px`.
3. Then, use `auto-fill` to generate as many columns of size between `100px` and `200px`.
4. How many columns does it create and why?

Now let's bring the `fr` unit and `auto-fit` into the game:

1. Change the maximum size from the fixed `200px` to the dynamic `1fr`.
2. How many columns are generated and why?
3. Now it becomes a little tricky: Replace `auto-fill` with `auto-fit`. Do the columns take up more space now? Why or why not?
4. Increase the grid container's width to `540px`. Why are the columns now larger with `auto-fit` than with `auto-fill`? (change back to `auto-fill` to check this)

Hints: *Remember which value of* ***`minmax(min, max)`*** *the grid prefers to use if it has a fixed value (not* ***`auto`*** *or some* ***`fr`****). Use Firefox to see how many columns are generated and what width the columns have.*

Solution: View solution on GitHub

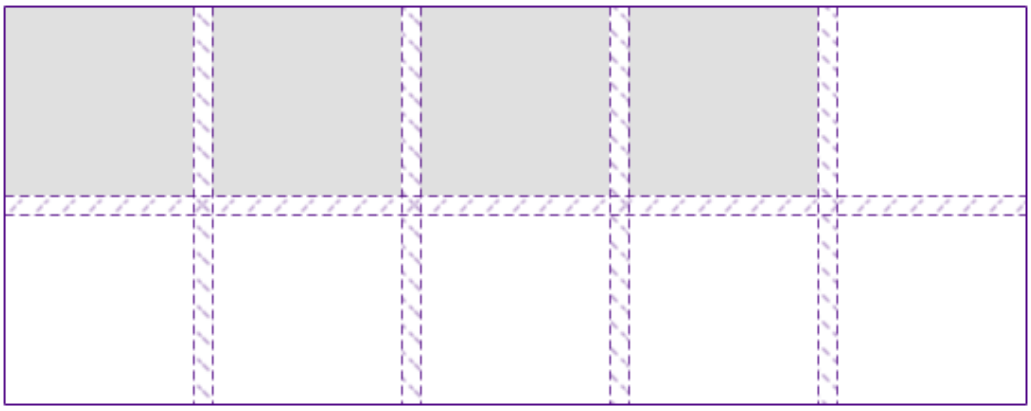

Figure 1: This is what happens in the last step when using `auto-fill` and a width if `540px`.