

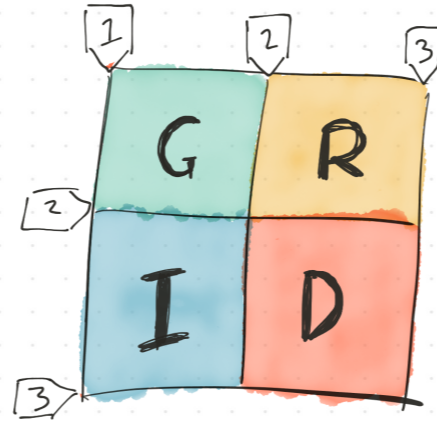
Anton Ball

Fenders, January 2018



Anton Ball

Senior UI Integrator at Velpic

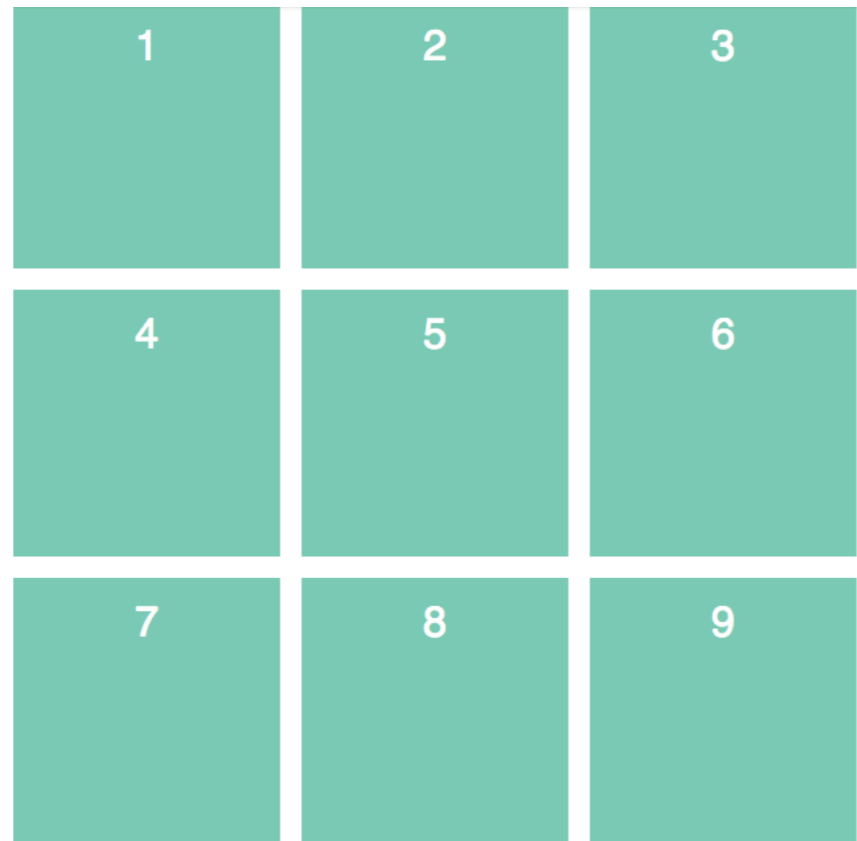


## What's Grid?

CSS Grid Layout is a CSS layout method designed for the two-dimensional layout of items on a webpage or application.

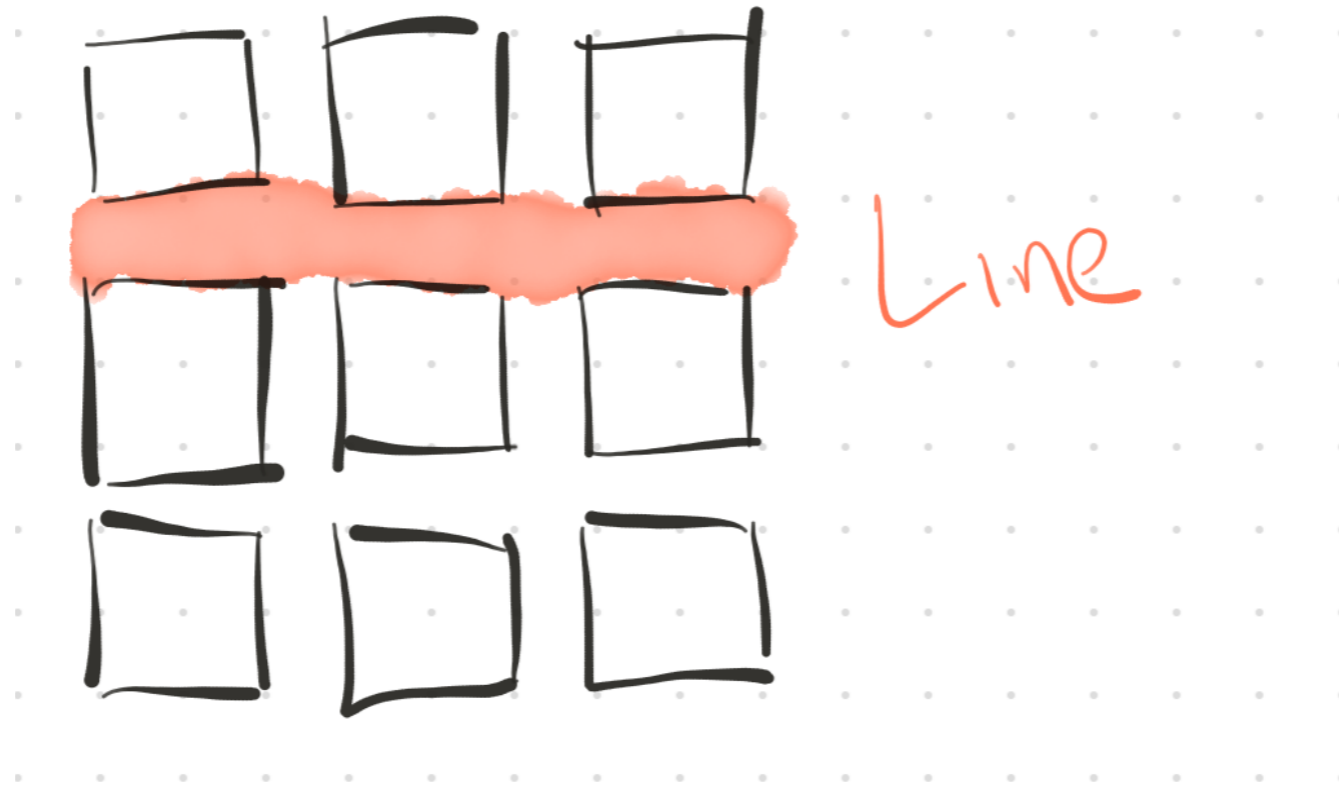
Rachel Andrew

# Basic grid



```
.grid {  
  display: grid;  
  grid-template-columns: 100px 100px 100px;  
  grid-template-rows: 100px 100px 100px;  
  grid-gap: 0.5rem;  
}
```

What's this?

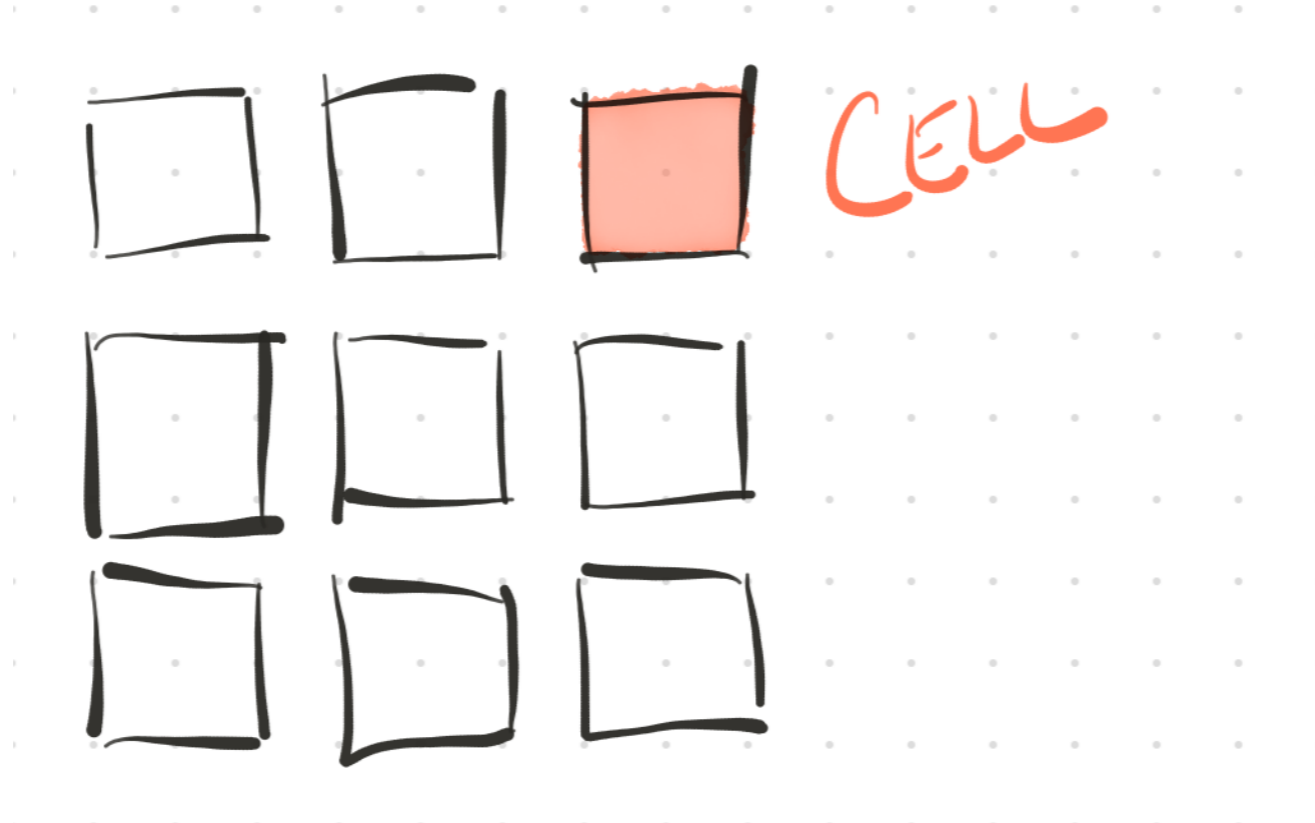


What's this?

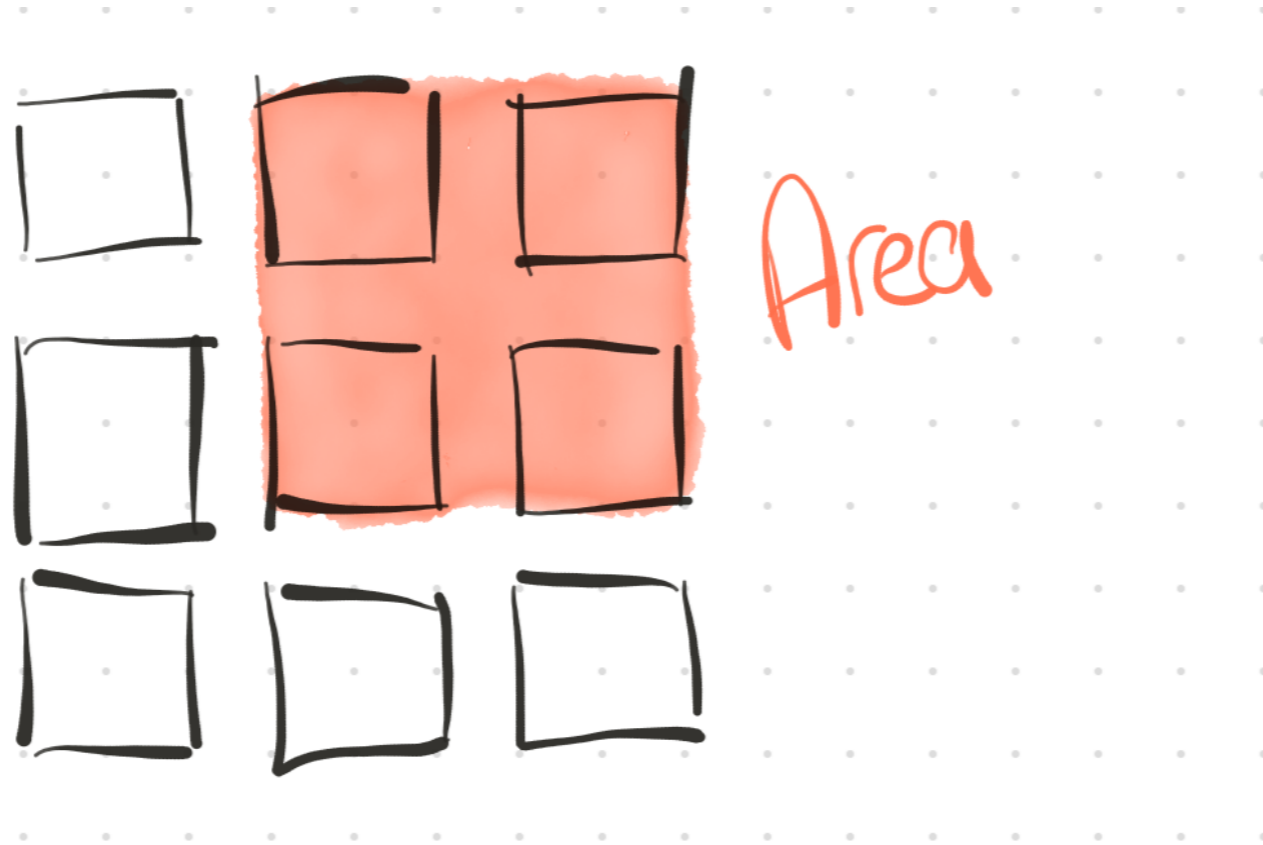


TRACK

What's this?



What's this?





# Example HTML

```
<div class="grid">  
  <div class="cell">1</div>  
  <div class="cell">2</div>  
  <div class="cell">3</div>  
  <div class="cell">4</div>  
  <div class="cell">5</div>  
  <div class="cell">6</div>  
  <div class="cell">7</div>  
  <div class="cell">8</div>  
  <div class="cell">9</div>  
</div>
```

# So what's new?

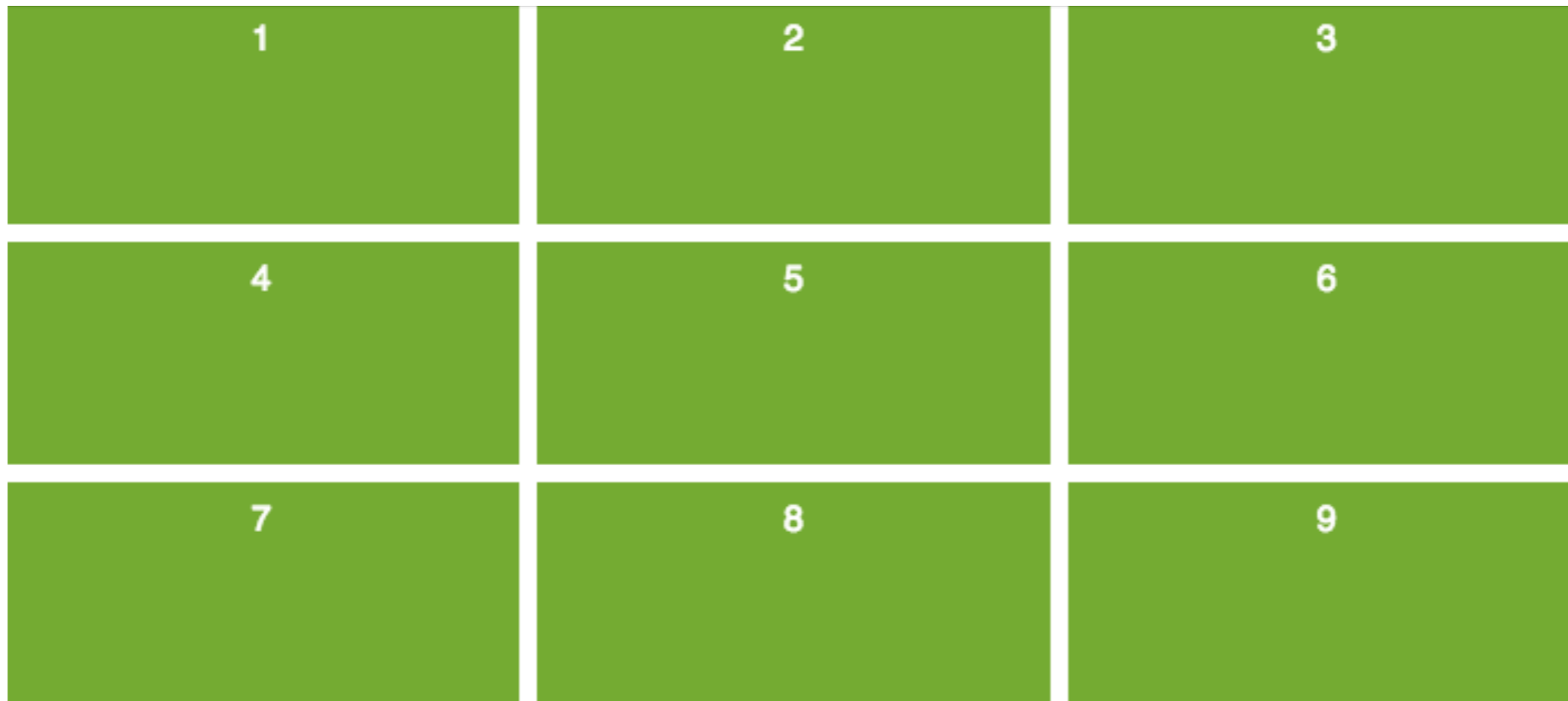
```
.grid {  
  display: grid;  
  grid-template-columns: 100px 100px 100px;  
  grid-template-rows: 100px 100px 100px;  
  grid-gap: 0.5rem;  
}
```

- **display: grid**  
Where the magic happens. All children to grid items
- **grid-template-columns/grid-template-rows**  
The number and size of the columns/rows.  
There's shorthand syntax...but I won't use it
- **grid-gap**  
Essentially the margin between cells. You'll know gutter if you have used a grid framework before.

That's interesting...  
if you're making a static gallery

Lets look at new units and CSS functions and make our grid fluid

## New unit on the block, fraction (fr)

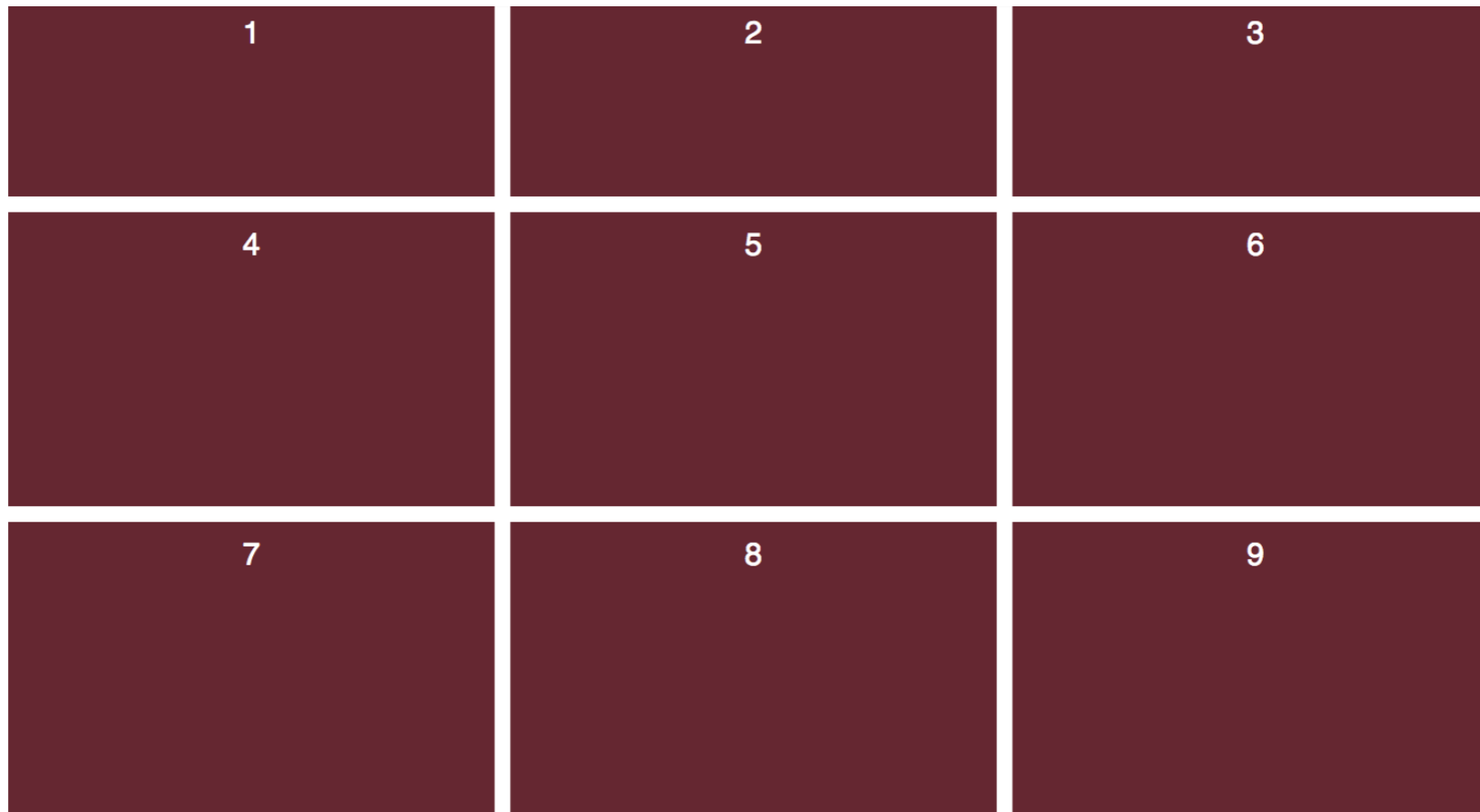


```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-template-rows: 100px 100px 100px;  
  grid-gap: 0.5rem;  
}
```

## Setting up rows and columns is repetitive

If we had a lot more columns than we are currently making, this will get painful. Enter `repeat`.

# repeat()

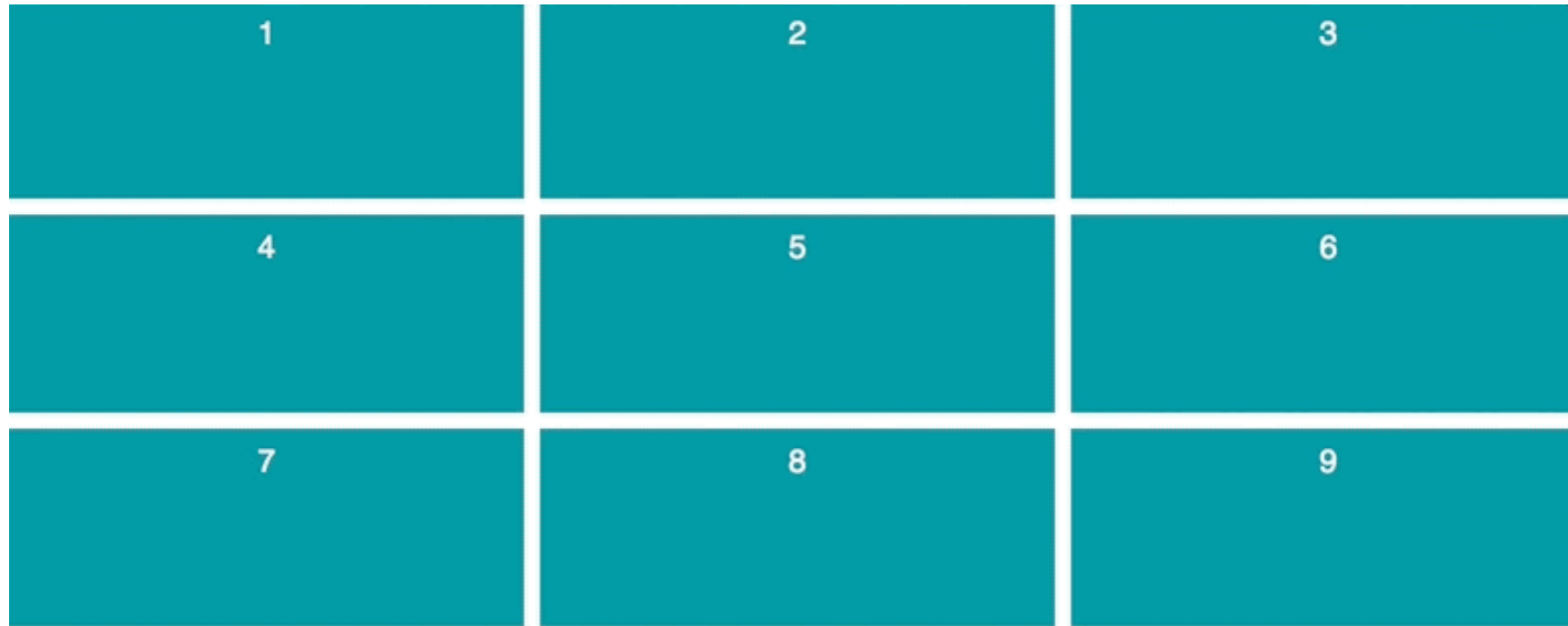


```
.grid {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: 100px repeat(2, 150px);  
  grid-gap: 0.5rem;  
}
```

## Cells could get tiny, so minmax()

Sets the sizes no smaller than the min and no greater than the max. You can get a long way with this without needing media queries!

# minmax()



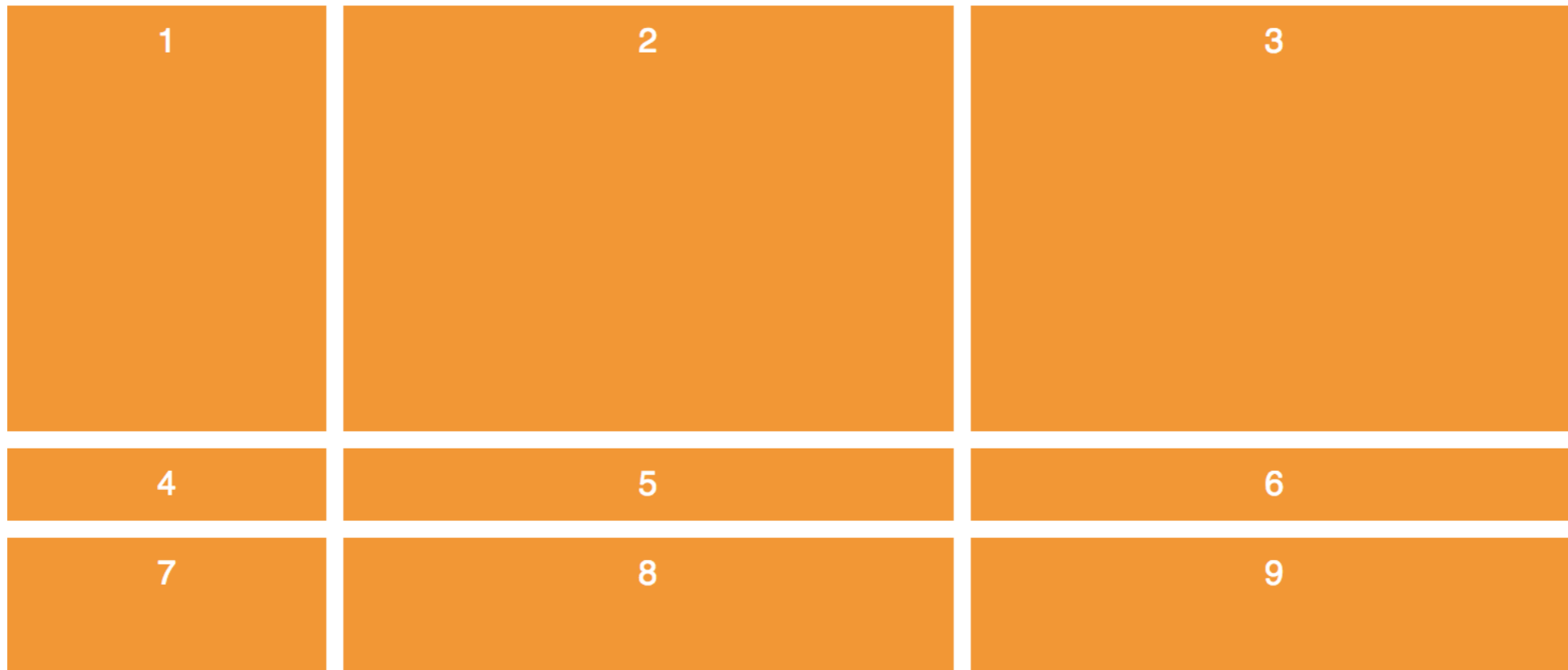
```
.grid {  
  display: grid;  
  grid-template-columns: repeat(3, minmax(200px, 1fr));  
  grid-template-rows: repeat(3, 100px);  
  grid-gap: 0.5rem;  
}
```



## All together now

You aren't limited to just one of the units we have discussed. You can mix them all together.

# All together now

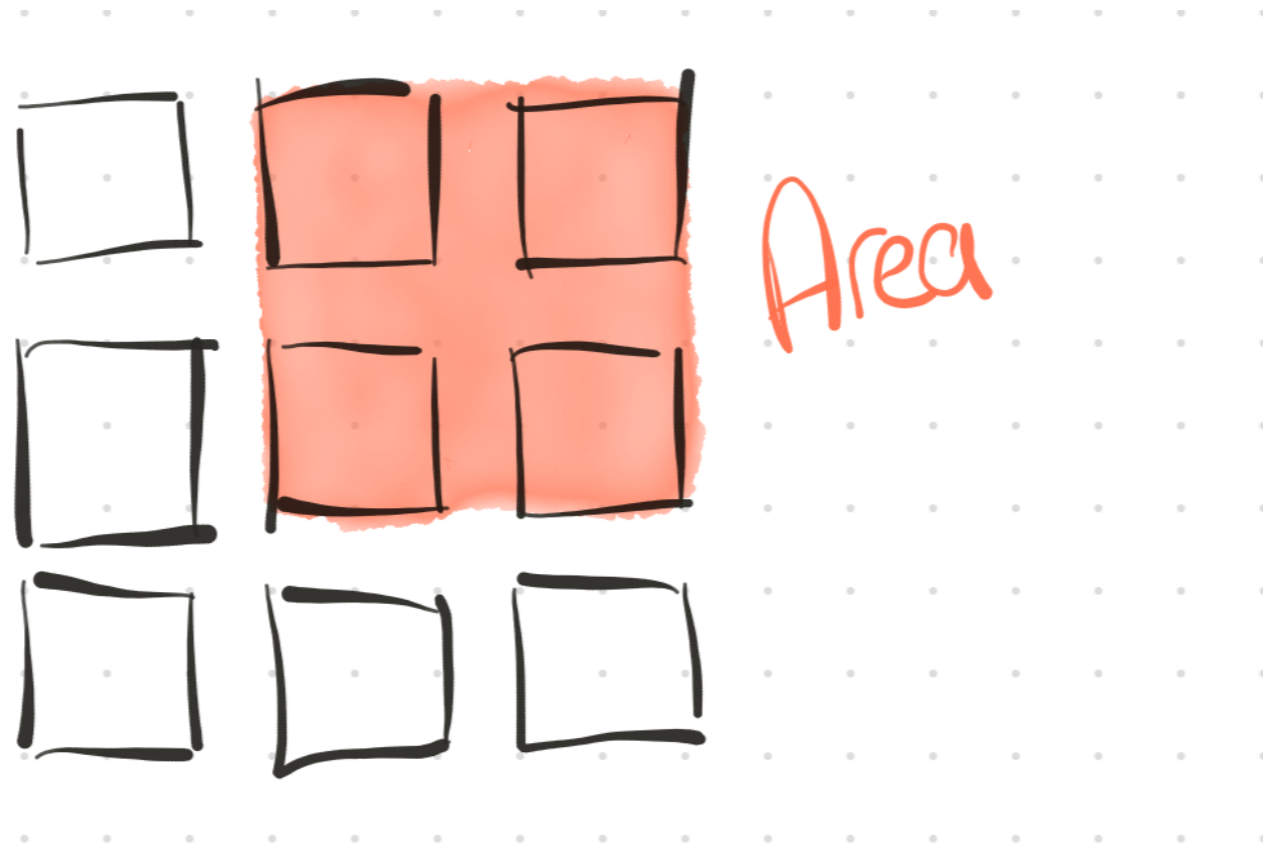


```
.grid {  
  display: grid;  
  grid-template-columns: 150px repeat(2, 1fr);  
  grid-template-rows: minmax(200px, 1fr) 10% 4rem;  
  grid-gap: 0.5rem;  
}
```

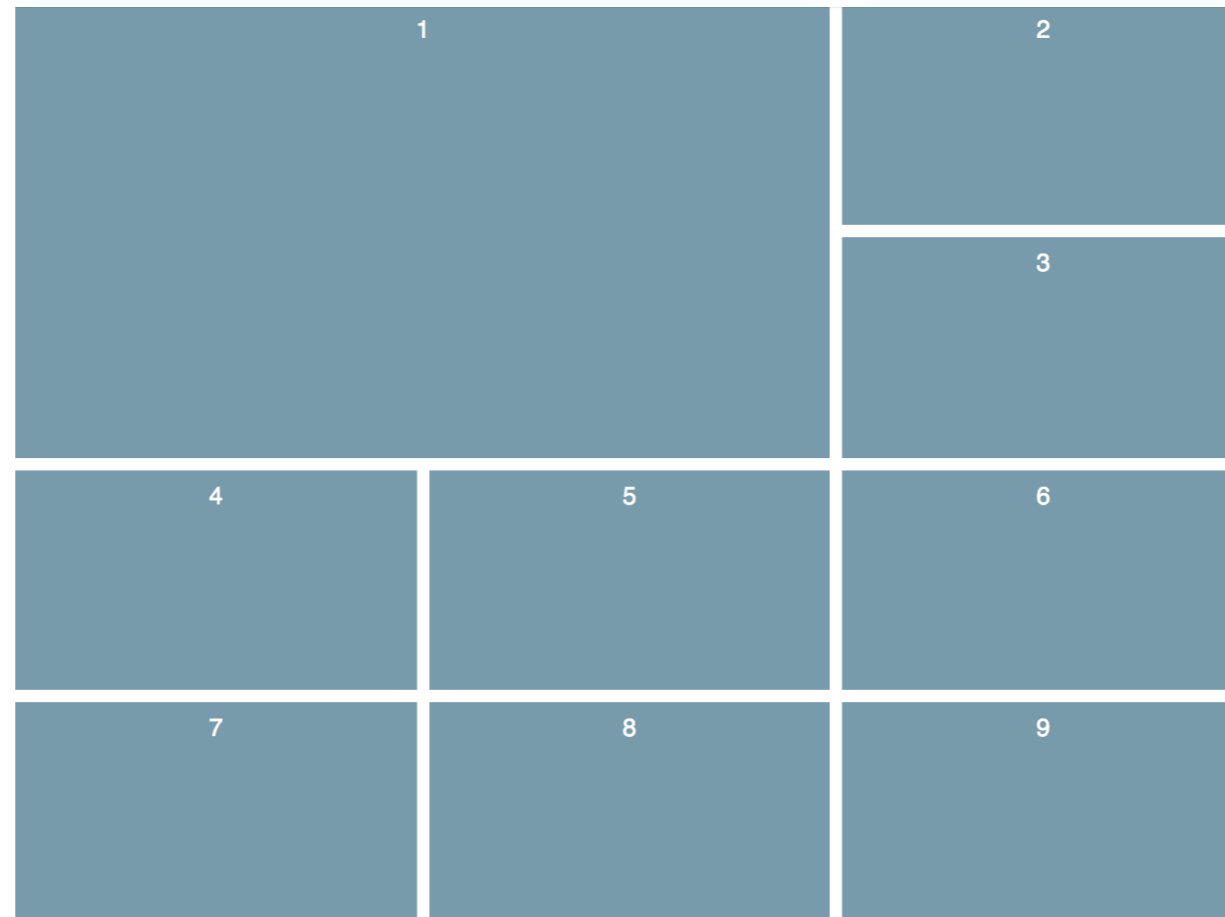
# Row/Column further reading

- **grid-column: span**  
You can define how many columns to span
- **grid-template-column: auto/auto-fit**  
Enables a lot of responsiveness without media queries
- **grid-auto-rows/columns**  
Create implicit grid tracks

# Positioning areas

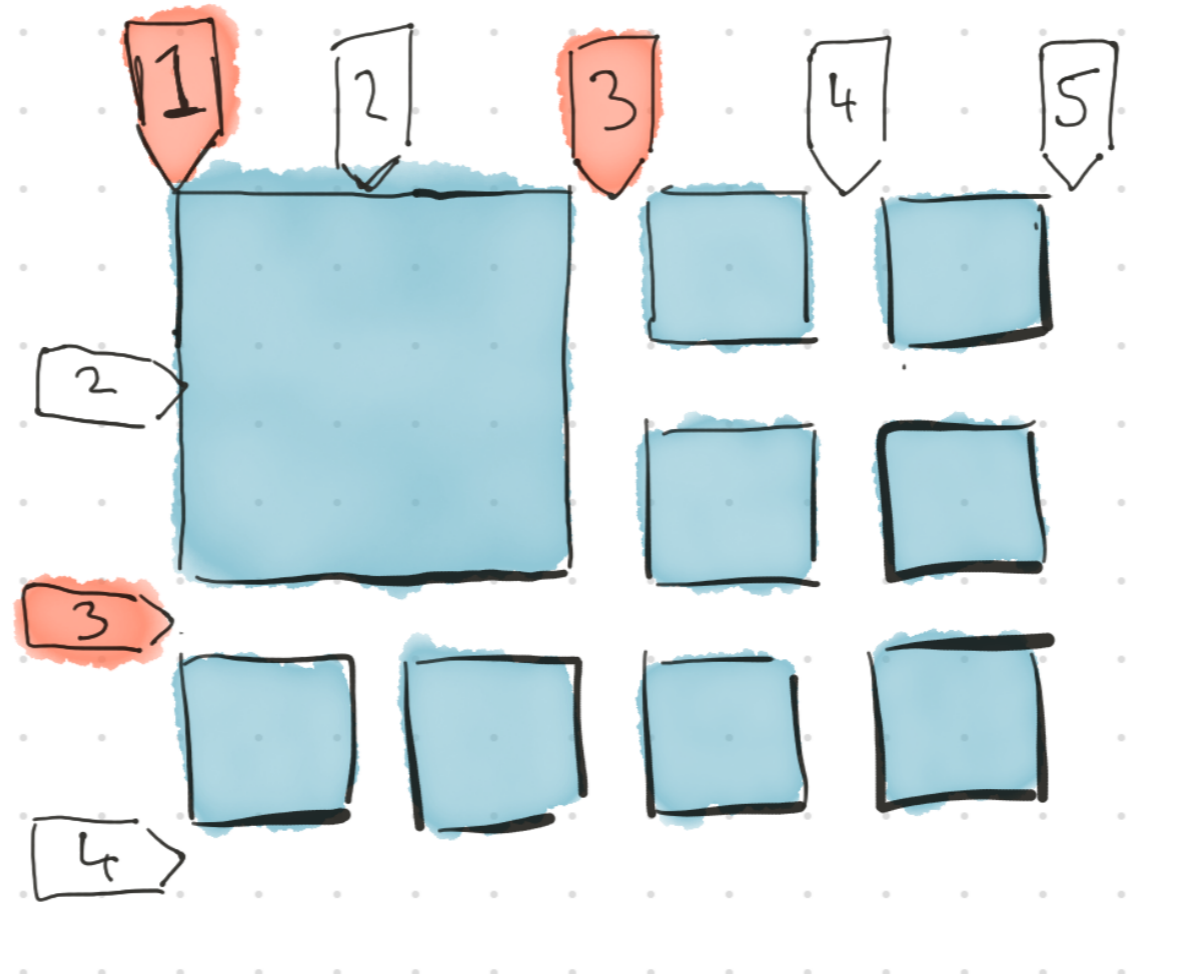


# Positioning areas



```
.cell-1 {  
  grid-column: 1 / 3;  
  grid-row: 1 / 3;  
}
```

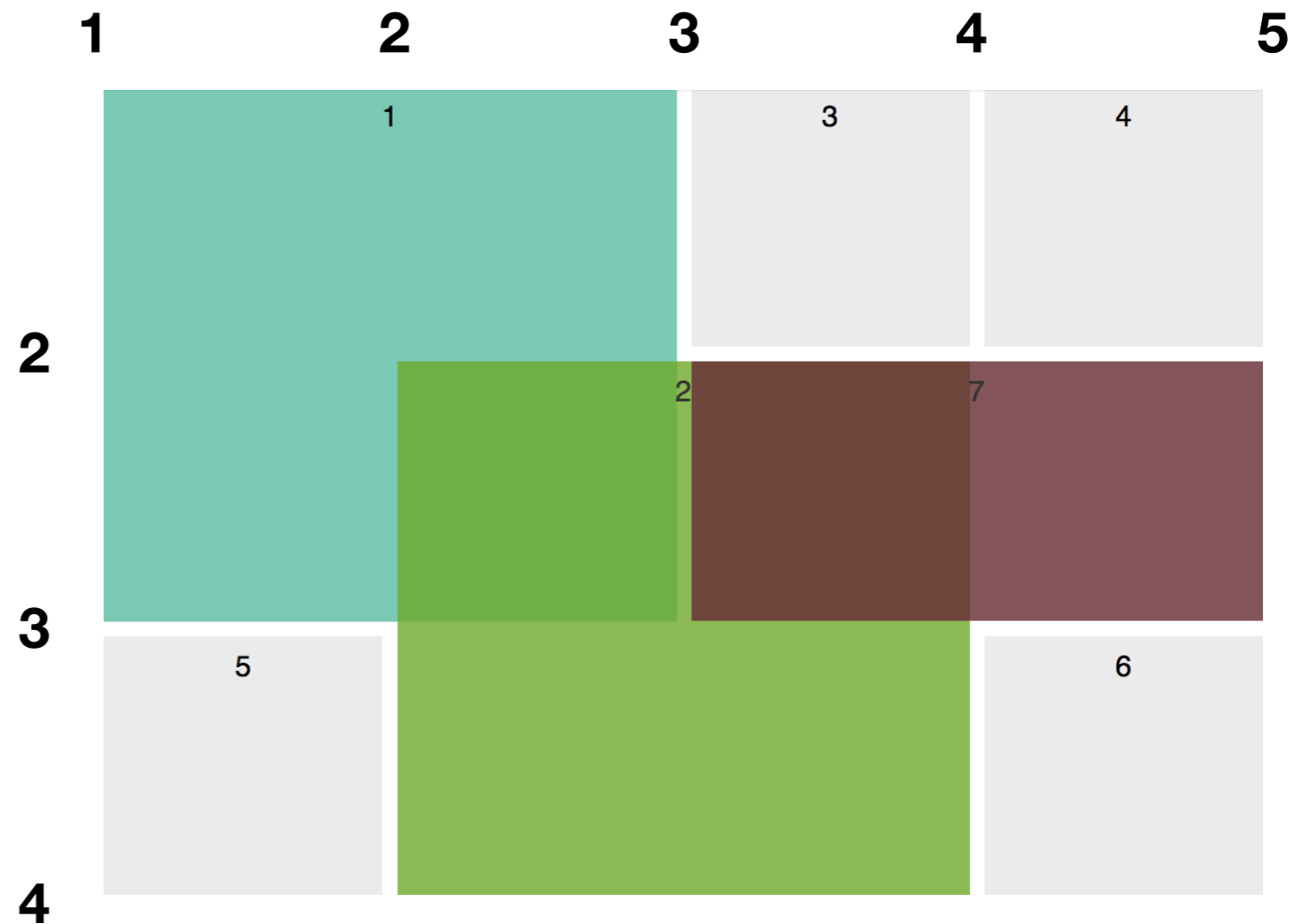
# Grid column/row



```
.cell-1 {  
  grid-column: 1 / 3;  
  grid-row: 1 / 3;  
}
```

# Overlap

```
.cell-1 {  
  grid-row: 1 / 3;  
  grid-column: 1 / 3;  
}  
  
.cell-2 {  
  grid-row: 2 / 4;  
  grid-column: 2 / 4;  
}  
  
.cell-7 {  
  grid-column: 3 / 5;  
  grid-row: 2 / 3;  
}
```



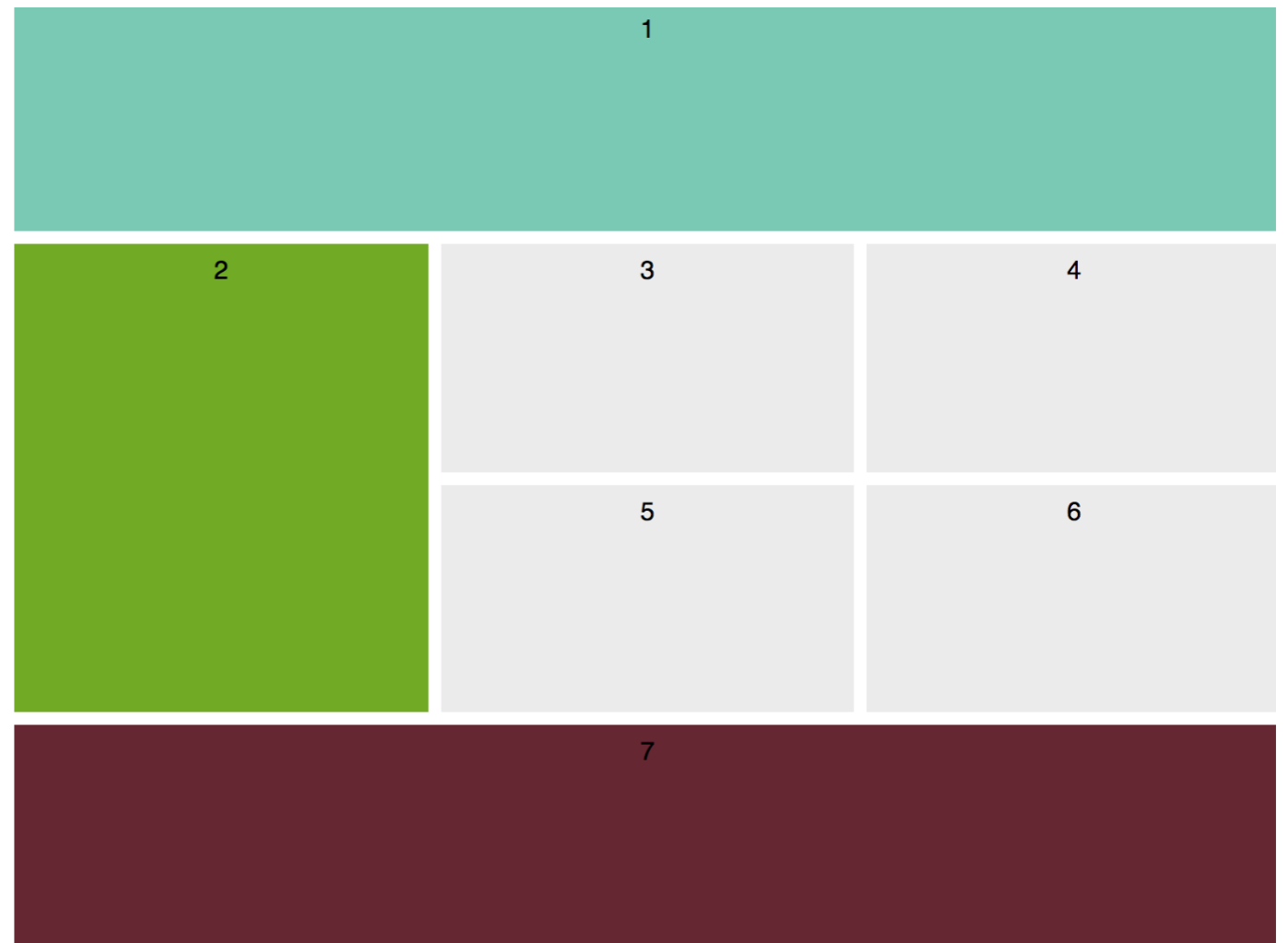
The holy grail

at least it was.



# The holy grail

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: auto;  
  grid-gap: 0.5rem;  
}  
  
.cell-1 {  
  grid-column: 1 / 4;  
}  
  
.cell-2 {  
  grid-row: 2 / 4;  
}  
  
.cell-7 {  
  grid-column: 1 / 4;  
}
```

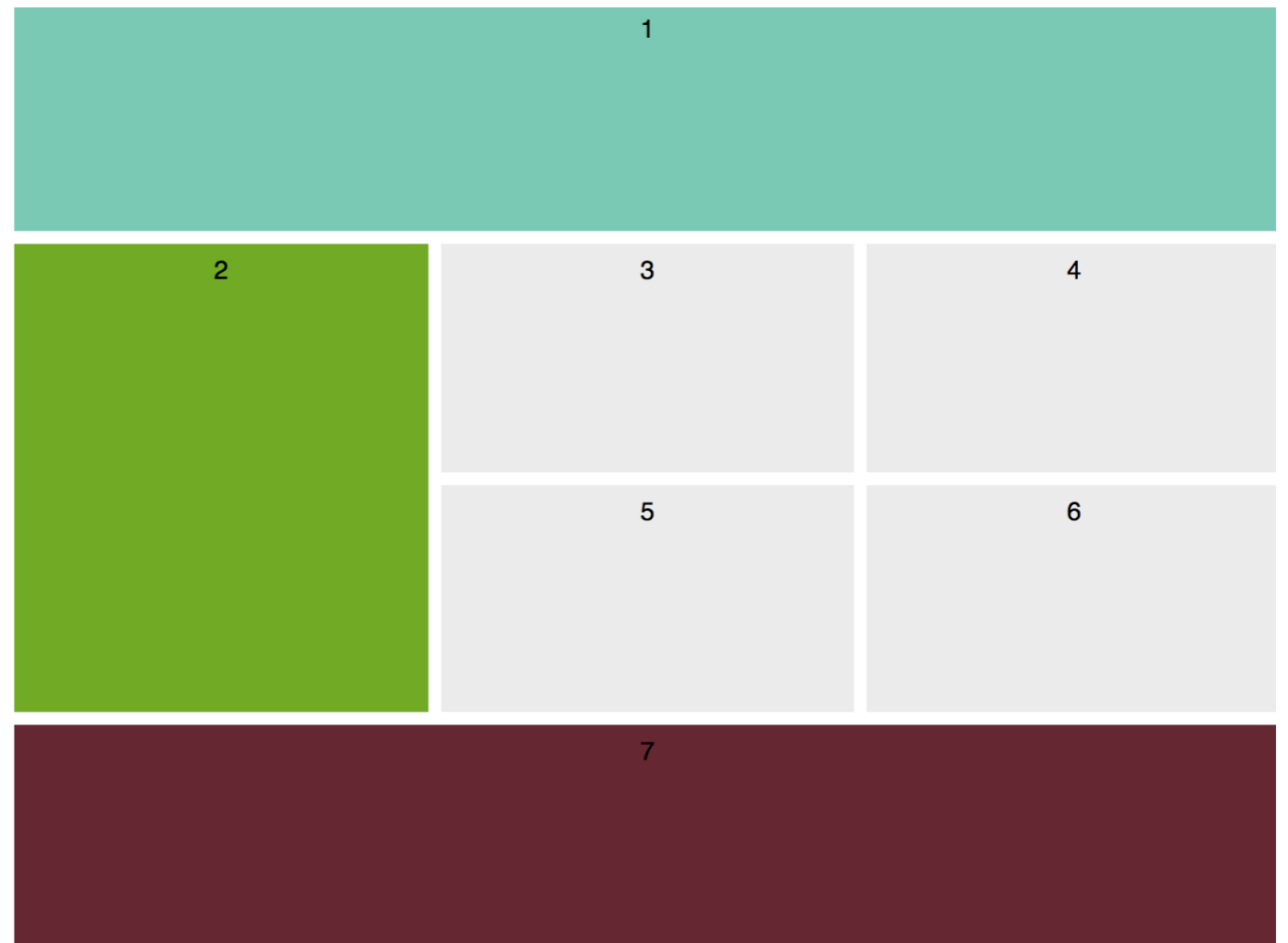


We can do better

the numbering can be tricky, naming is a nicer option

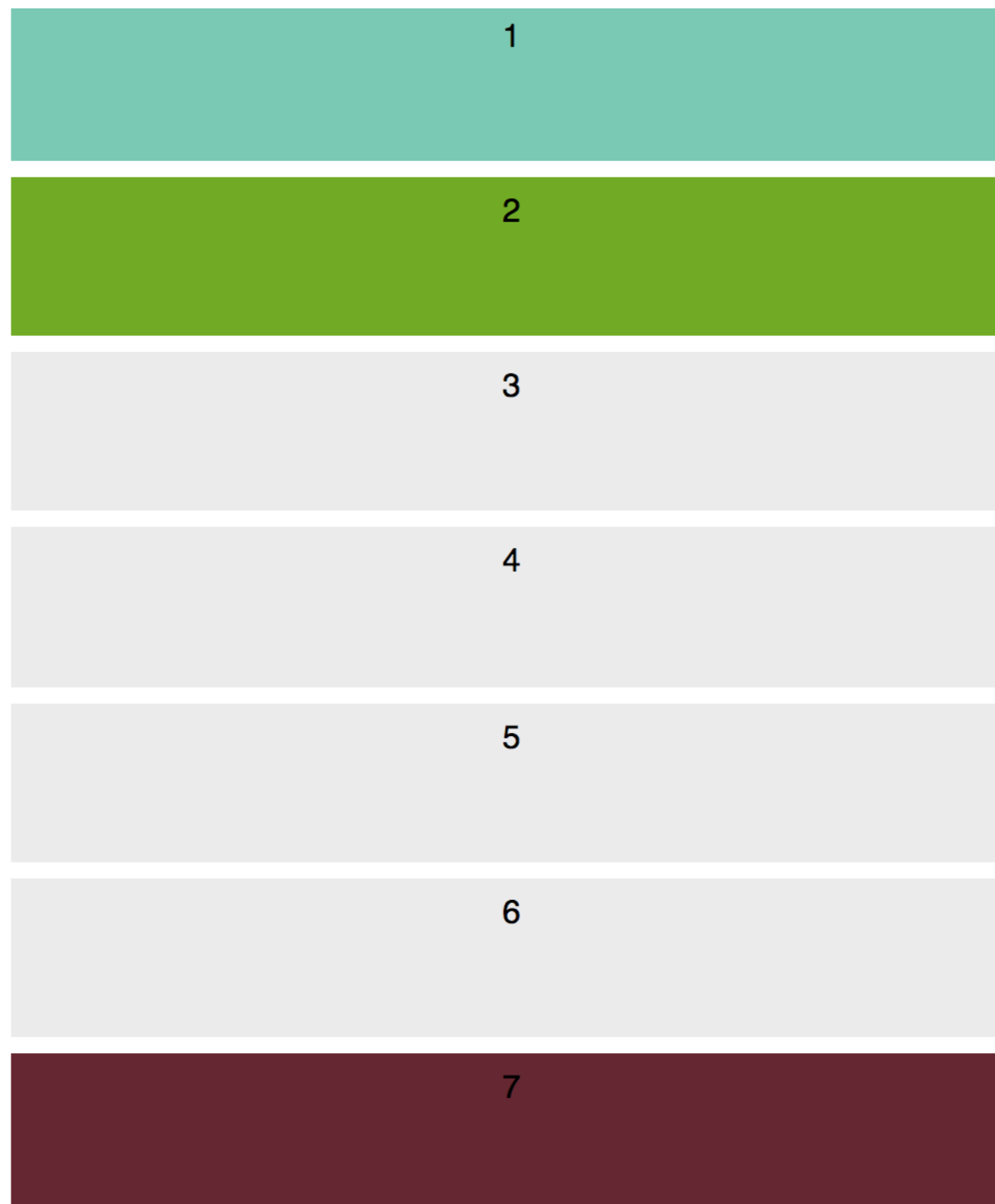
# The named holy grail

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-template-rows: auto;  
  grid-gap: 0.5rem;  
  grid-template-areas:  
    "nav nav nav"  
    "aside contenta contentb"  
    "aside contentc contentd"  
    "footer footer footer";  
}  
  
.cell-1 {  
  grid-area: nav;  
}  
  
.cell-2 {  
  grid-row: aside;  
}
```



# Responsive

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns: 1fr;  
  grid-template-rows: auto;  
  grid-gap: 0.5rem;  
  grid-template-areas:  
    "nav"  
    "aside"  
    "contenta"  
    "contentb"  
    "contentc"  
    "contentd"  
    "footer";  
}
```

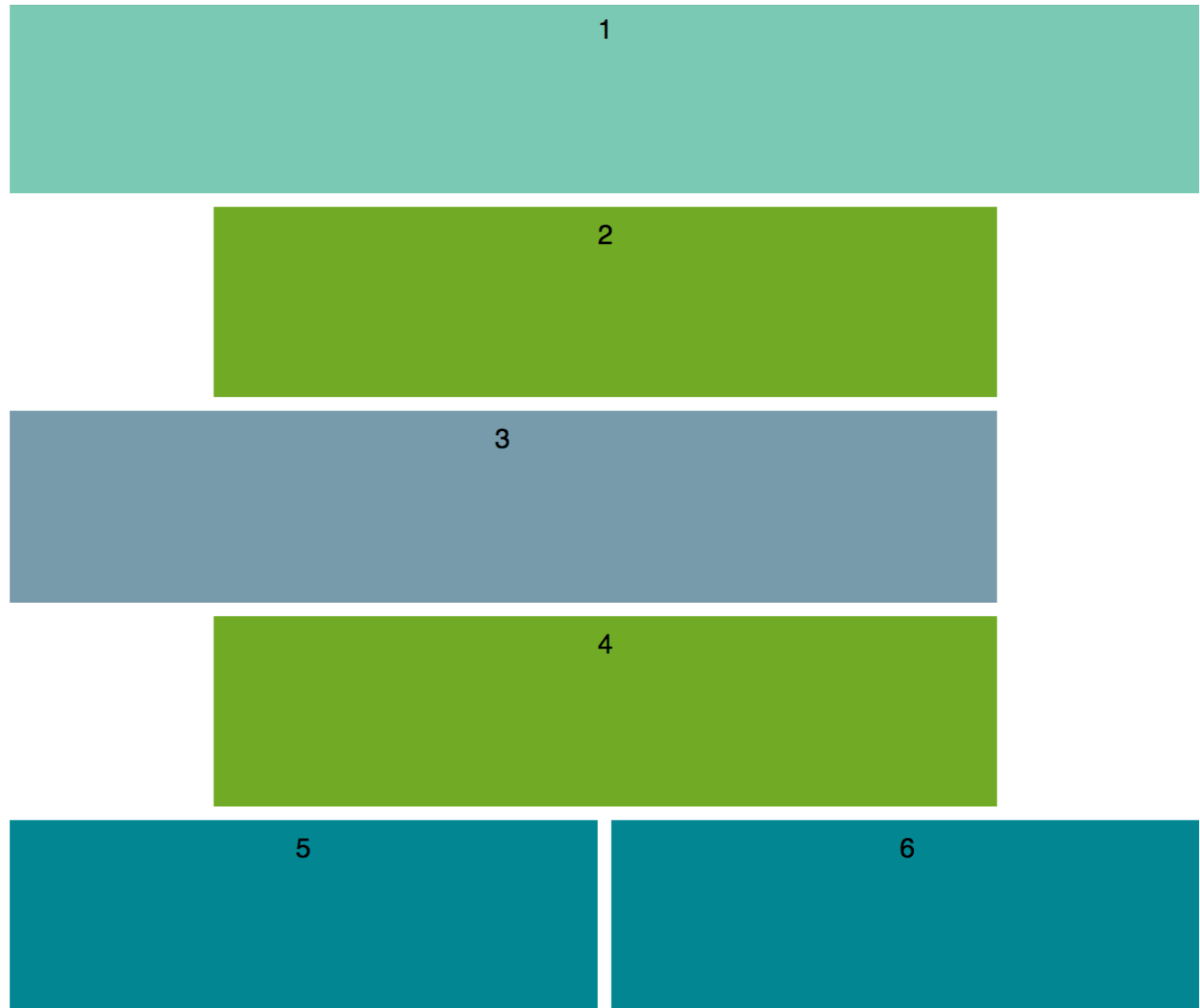


## Named lines

You can name each line in a row or column

# Named lines

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns:  
    [full-start] 1fr  
    [container-start] 2fr  
    [center-start] 2fr  
    [container-end] 1fr  
    [full-end];  
  grid-template-rows: auto;  
  grid-gap: 0.5rem;  
}  
  
.cell-1 {  
  grid-column: full-start / full-end;  
}  
  
.cell-2 {  
  grid-column: container-start / container-end;  
}  
  
.cell-5 {  
  grid-column: full-start / center-start;  
}  
  
.cell-6 {  
  grid-column: center-start / full-end;  
}
```



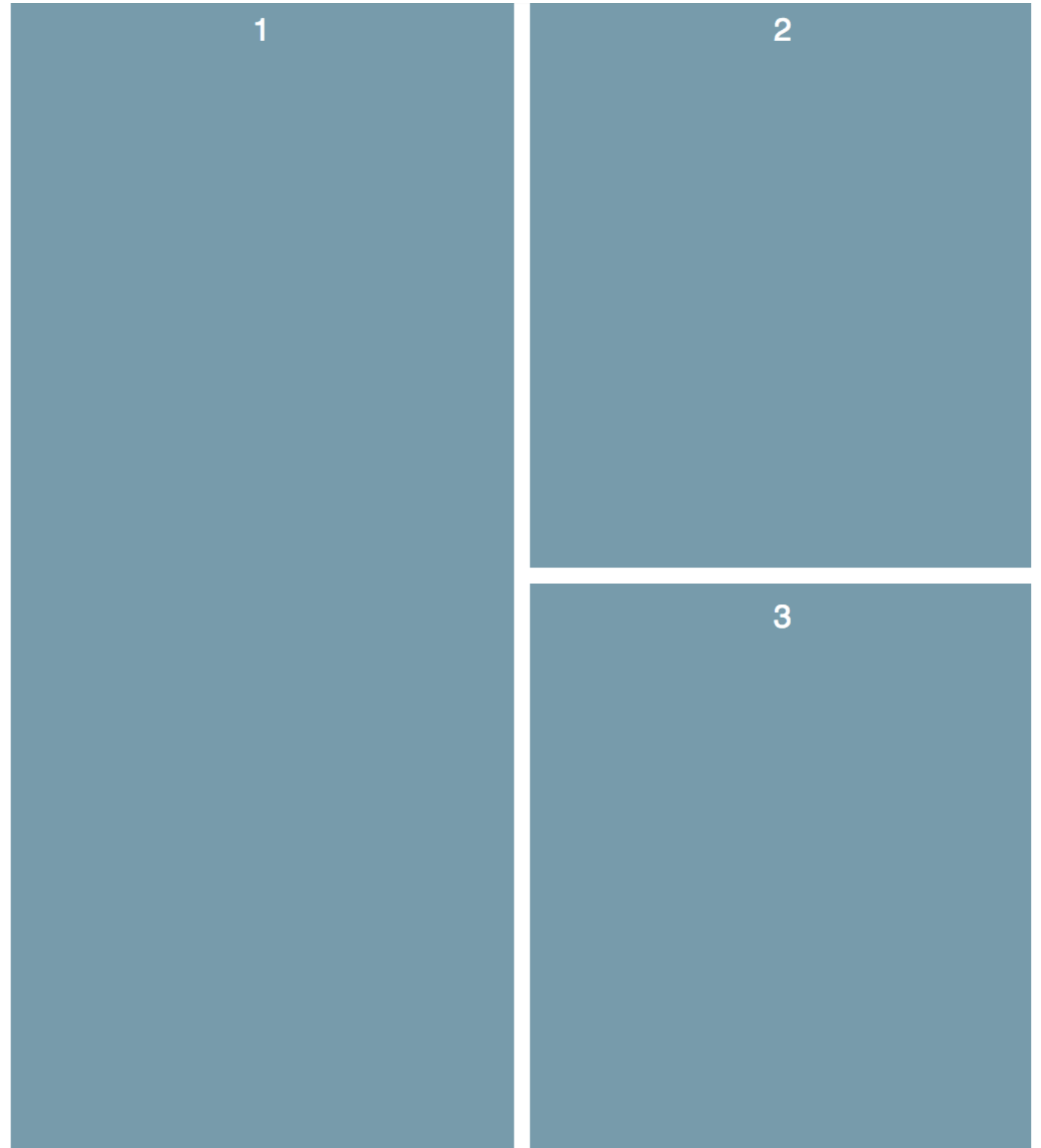
# Named lines

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns:  
    [full-start] 1fr  
    [container-start] 2fr  
    [center-start] 2fr  
    [container-end] 1fr  
    [full-end];  
  grid-template-rows: auto;  
  grid-gap: 0.5rem;  
}  
  
.cell-1 {  
  grid-column: full-start / full-end;  
}  
  
.cell-2 {  
  grid-column: container-start / container-end;  
}  
  
.cell-5 {  
  grid-column: full-start / center-start;  
}  
  
.cell-6 {  
  grid-column: center-start / full-end;  
}
```



# Named lines

```
.grid {  
  height: 100%;  
  display: grid;  
  grid-template-columns:  
    [main-start] 1fr  
    [main-end aside-start] 1fr  
    [aside-end];  
  grid-template-rows:  
    [main-start] 1fr 1fr  
    [main-end];  
  grid-gap: 0.5rem;  
}  
  
.cell-1 {  
  grid-area: main;  
}
```





# Support

It's not bad

| IE              | Edge *          | Firefox | Chrome          | Safari | iOS Safari * | Opera Mini * | Chrome for Android | UC Browser for Android | Samsung Internet |
|-----------------|-----------------|---------|-----------------|--------|--------------|--------------|--------------------|------------------------|------------------|
|                 |                 |         | <sup>1</sup> 49 |        | 9.3          |              |                    |                        |                  |
|                 |                 |         | 61              |        | 10.2         |              |                    |                        |                  |
|                 | <sup>2</sup> 15 |         | 62              | 10.1   | 10.3         |              |                    |                        | 4                |
| <sup>2</sup> 11 | 16              | 57      | 63              | 11     | 11.2         | all          | 62                 | 11.4                   | 6.2              |
|                 | 17              | 58      | 64              | TP     |              |              |                    |                        |                  |
|                 |                 | 59      | 65              |        |              |              |                    |                        |                  |
|                 |                 | 60      | 66              |        |              |              |                    |                        |                  |
|                 |                 |         |                 |        |              |              | Global             | 72.4% + 4.31% = 76.71% |                  |
|                 |                 |         |                 |        |              |              | unprefixed:        | 72.4%                  |                  |

# Use in production

- **There's no Polyfill**  
Grid is too complex
- **@supports(display: grid)**  
Fallback will likely be Flex - depends on your project
- **JS load a fallback**  
There's efficiency to be gained  
Stripe does this, but...

# Further Reading

- **Rachel Andrew**  
anything - book, talks, blogs, but especially:
- **gridbyexample.com/**
- **Wes Bos is releasing a grid course**
- **mozilladevelopers.github.io/playground/**



Wes Bos 🔥 ✓  
@wesbos

TOMORROW [ift.tt/2ESwnK2](https://ift.tt/2ESwnK2)



18/1/18, 2:35 am

Thank you

Examples on CodePen:

**<https://codepen.io/collection/nLxmgY/>**