

Rachel Andrew @ MirrorConf

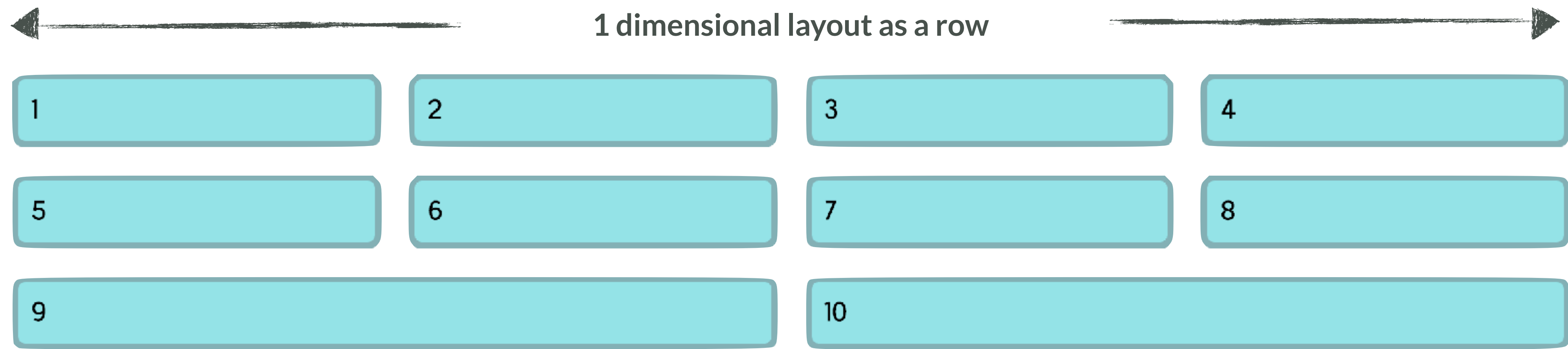
Solving Layout problems
with CSS Grid & friends

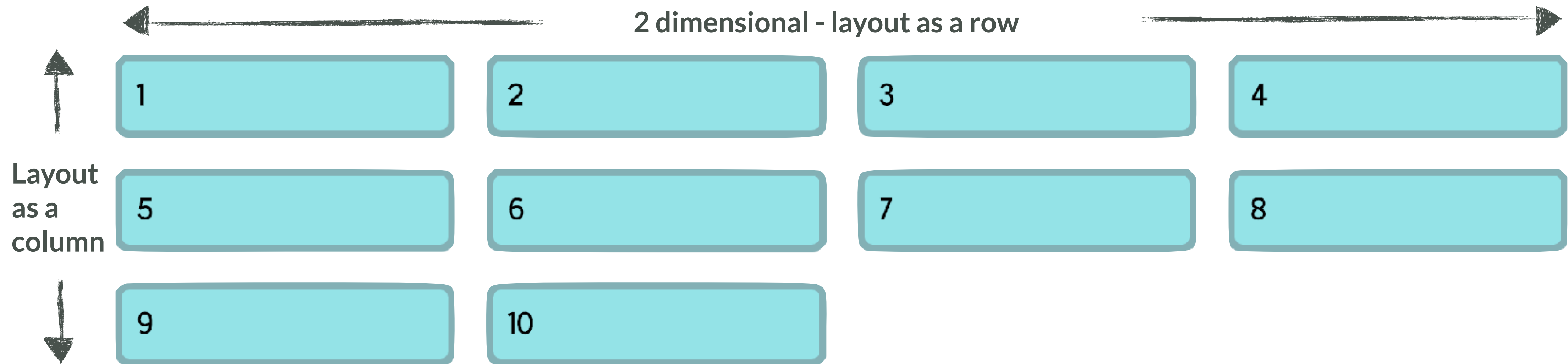
- ▶ @rachelandrew on Twitter
- ▶ <https://rachelandrew.co.uk>
- ▶ Invited Expert to the CSS Working Group
- ▶ Google Developer Expert
- ▶ co-founder Perch and Perch Runway CMS - <https://grabaperch.com>

So, about this Grid thing ...

... why not
use Flexbox?

Do you need layout in
one dimension or two?





Grid works from the container in

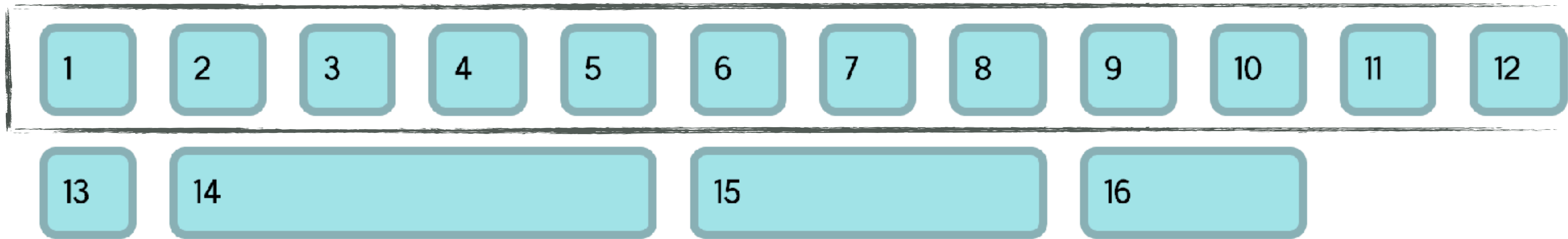
Every other method of creating a grid,
involves sizing the individual items.

A float based “grid”

We have to give the items a width. By stacking these carefully sized items up we get the appearance of a grid.

```
.col {  
    padding: 10px;  
    margin-bottom: 1em;  
    margin-left: 2.093333%;  
    width: 6.20%;  
    float: left;  
}  
  
.row::after {  
    content: "";  
    display: block;  
    clear: both;  
}  
  
.col.span2 {  
    width: calc((6.20%*2) + 2.093333%);  
}
```

row wrapper



$$(6.20\% * 4) + (2.093333\% * 3)$$

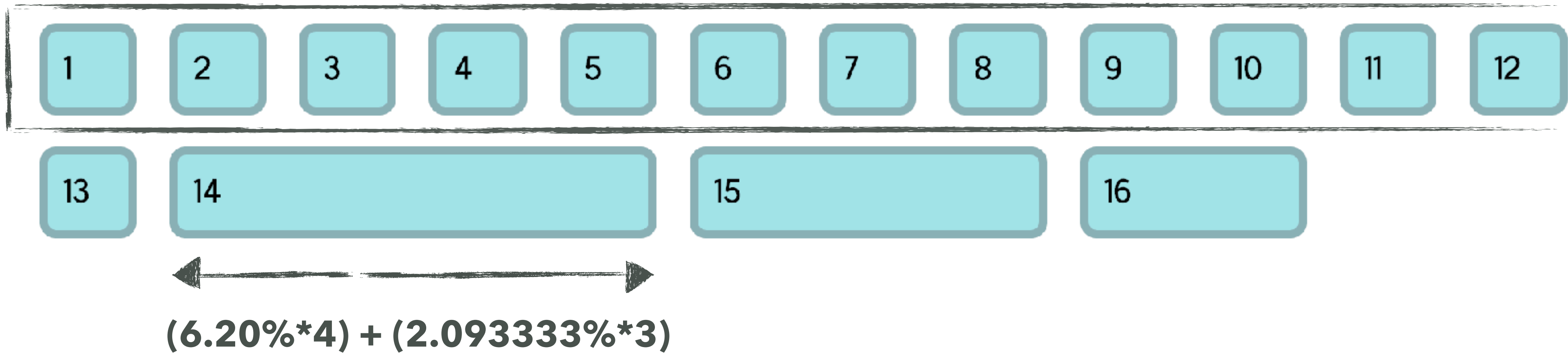
There is no grid. We made it look like
there is a grid by the fact things line up.

A flexbox “grid”

Using the width as the flex-basis.

```
.wrapper .row {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.col {  
  padding: 10px;  
  margin-bottom: 1em;  
  margin-left: 2.093333%;  
  width: 6.20%;  
  flex: 0 0 auto;  
}  
  
.col.span2 {  
  width: calc((6.20%*2) + 2.093333%);  
}
```


row wrapper as flex container

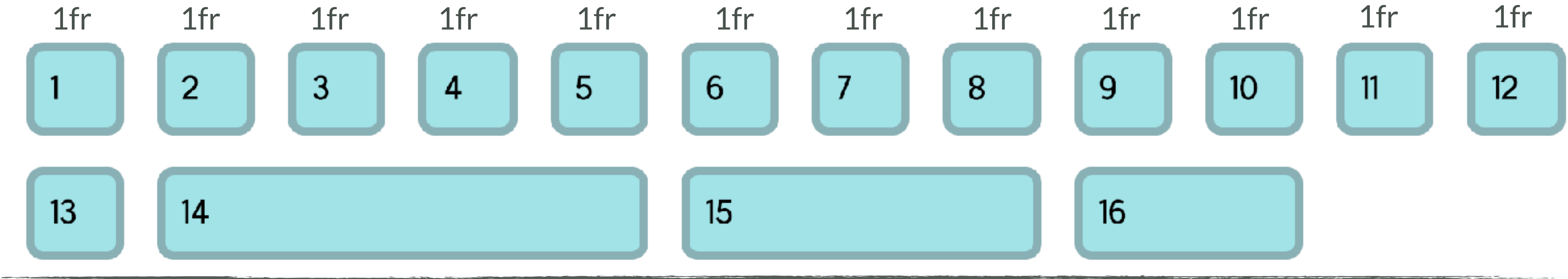


A Grid ... grid

No row wrappers. No sizing information on the items, just an instruction on how many columns to span.

```
.wrapper {  
  display: grid;  
  grid-template-columns:  
    repeat(12, minmax(0,1fr));  
  grid-gap: 20px;  
}  
  
.col.span2 {  
  grid-column: auto / span 2;  
}
```

Grid container



grid-column: 2 / span 4;

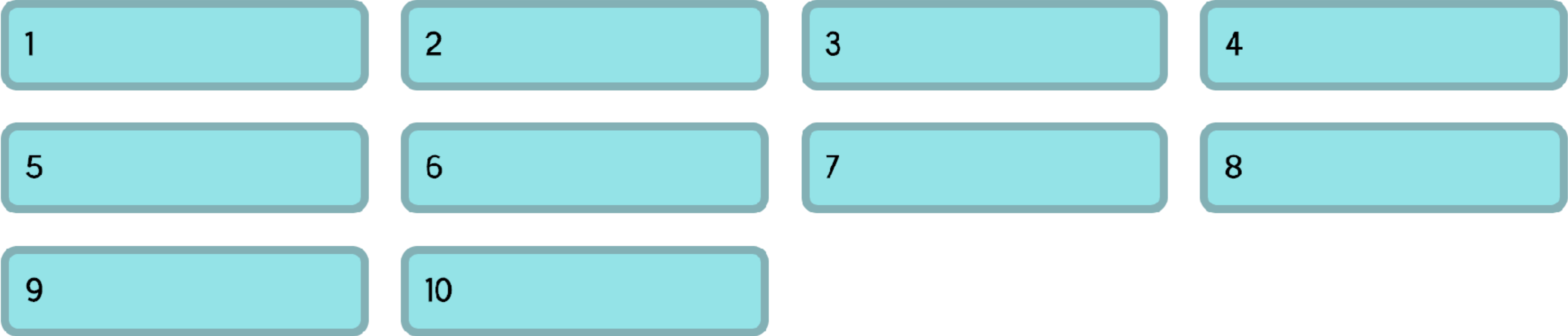
Grid frameworks create something that looks like a grid by controlling item size.

CSS Grid Layout creates an actual grid
and you place items into it.

CSS Grid Layout is a native CSS
framework. Built into the browser.

Precision & Flexibility

Sizing Grid Tracks



← minmax(200px, 1fr) →

repeat()

```
.grid-wrapper {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    repeat(auto-fill, minmax(200px, 1fr));  
}
```

auto-fill

```
.grid-wrapper {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    repeat(auto-fill,minmax(200px, 1fr));  
}
```

minmax()

```
.grid-wrapper {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    repeat(auto-fill, minmax(200px, 1fr));  
}
```

The fr unit - distributing available space

The fr unit

With this track listing the available spaces divided into 4.

```
.wrapper {  
  display: grid;  
  grid-template-columns: 2fr 1fr 1fr;  
  grid-gap: 20px;  
}
```

2fr

1fr

1fr

1

2

3

4

5

6

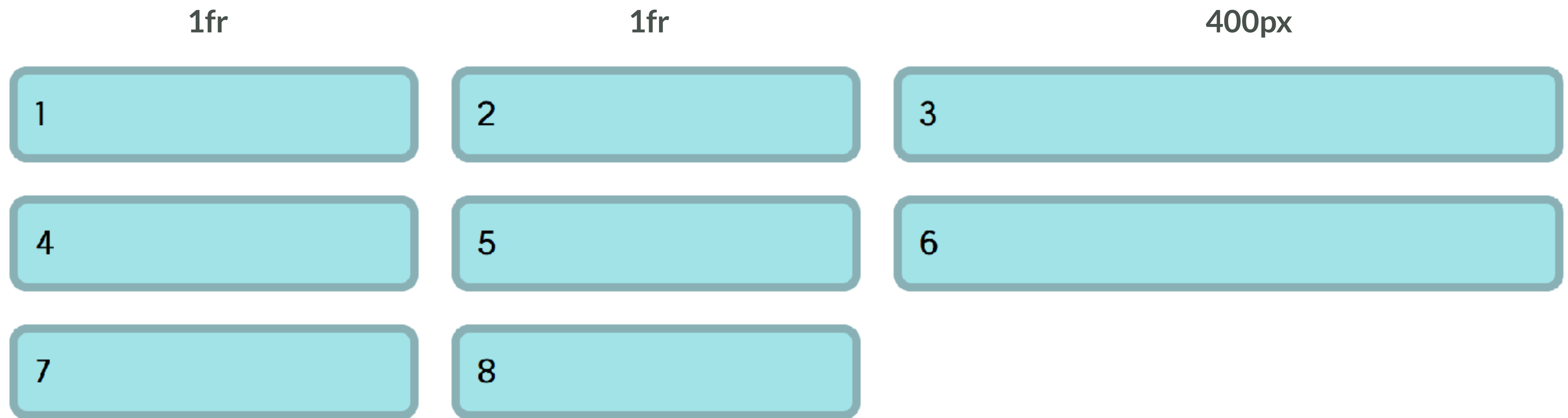
7

8

The fr unit

Mix absolute length units and fr units.

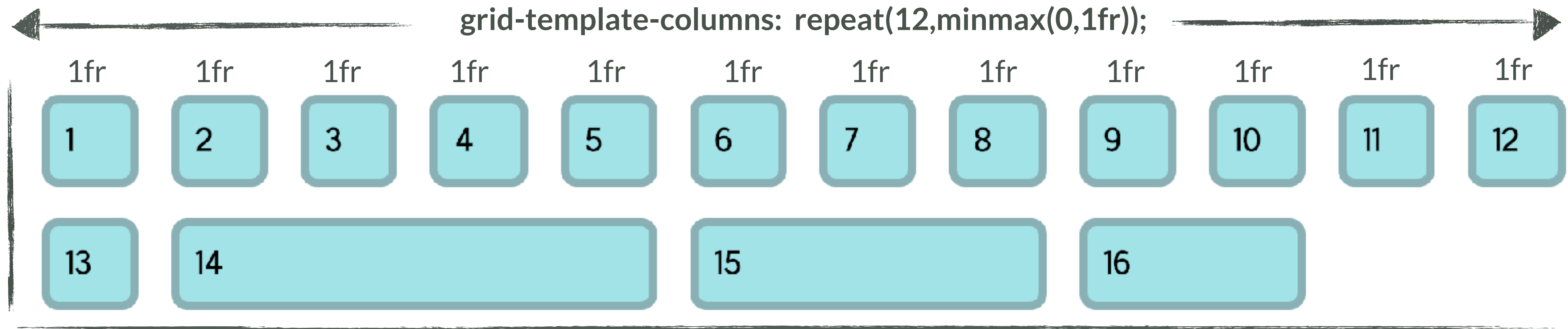
```
.wrapper {  
  display: grid;  
  grid-template-columns: 1fr 1fr 400px;  
  grid-gap: 20px;  
}
```



The fr unit

Creating a 12 column flexible grid with no percentage calculations.

```
.wrapper {  
  display: grid;  
  grid-template-columns:  
    repeat(12, minmax(0,1fr));  
  grid-gap: 20px;  
}
```



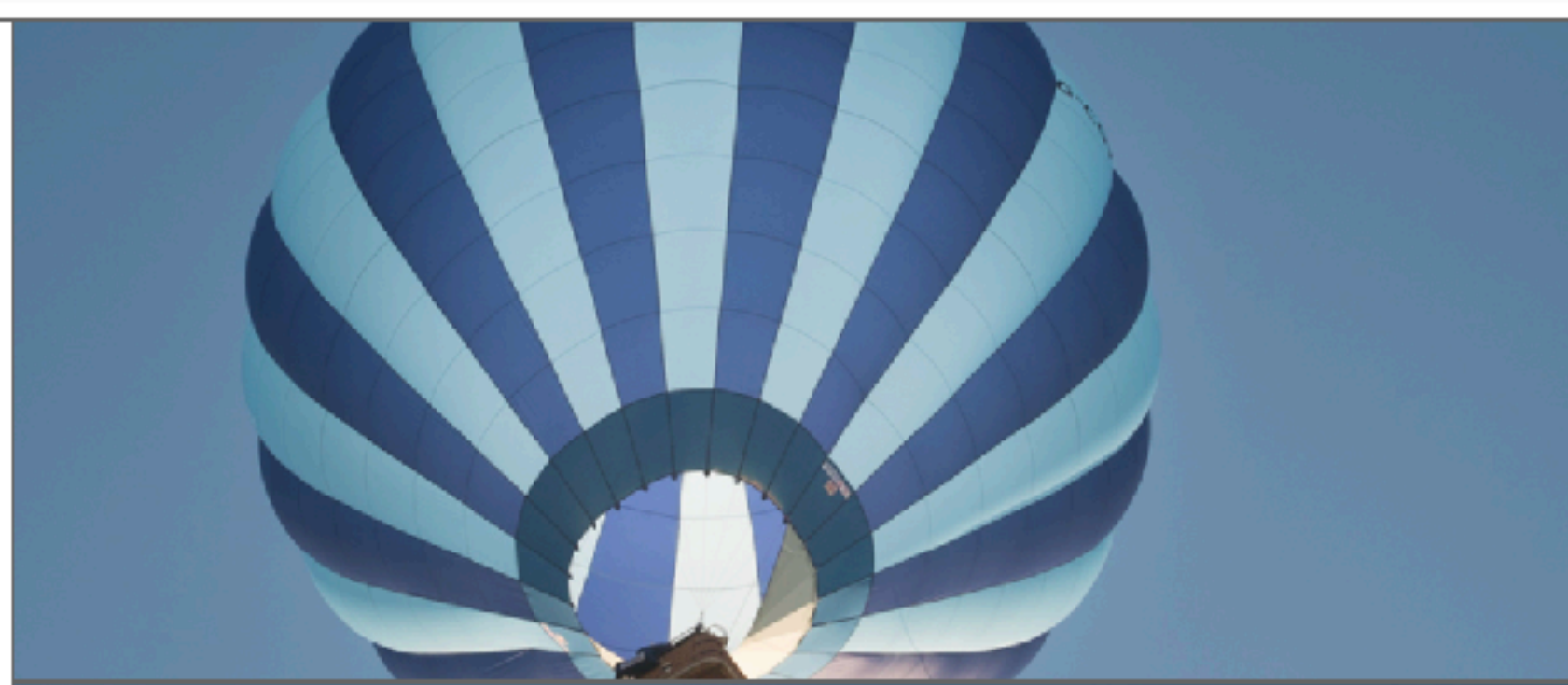
The fr unit replaces percentages in most cases when using grid layout.

minmax()

```
.grid-wrapper {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    repeat(auto-fill, minmax(200px, 1fr));  
}
```


Balloon Fiesta 2017

Despite the usual British weather, a not-insignificant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.



minmax()

Row tracks will be 200 pixels tall *unless* there is more content, in which case they will grow as the max is auto.

```
.panel {  
  max-width: 800px;  
  display: grid;  
  grid-template-columns: 2fr 3fr;  
  grid-auto-rows: minmax(200px, auto);  
  grid-gap: 1px;  
}
```


Balloon Fiesta 2017

Despite the usual British weather,
a not-insignificant number of hot air
balloons took to the Bristol skies.
Those of us who think 6am a
suitable time to be roaming around
with a camera got some nice
photos.

200px



minmax(200px, auto)



minmax(200px, auto)



minmax(200px, auto)

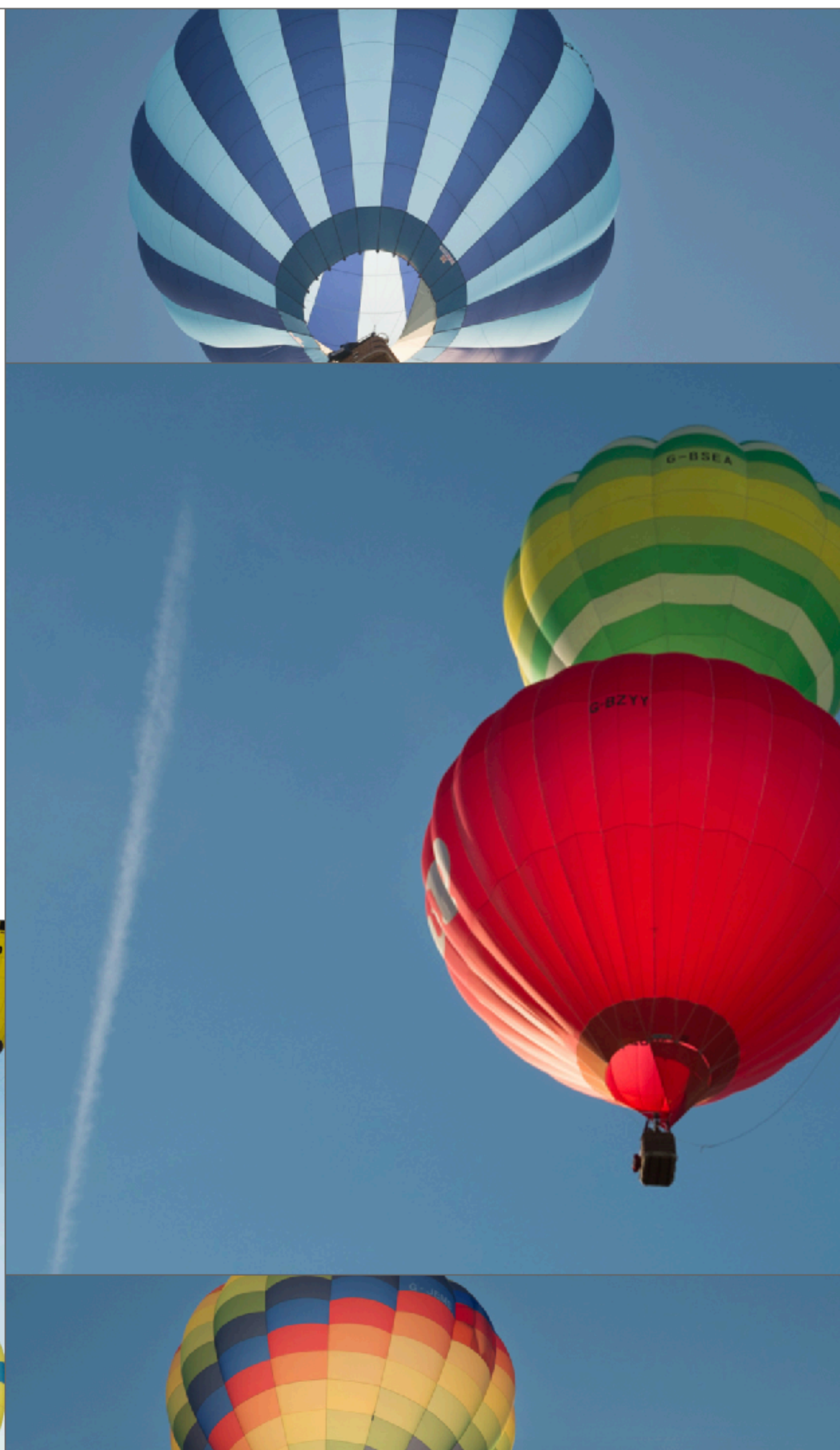


minmax(200px, auto)

Balloon Fiesta 2017

Despite the usual British weather, a not-insignificant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.

The morning launches are worth the early start though. By the evening most of Bristol is up on the hill toasting the ascent with the local cider.



Balloon Fiesta 2017

Despite the usual British weather, a not-insignificant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.

The morning launches are worth the early start though. By the evening most of Bristol is up on the hill toasting the ascent with the local cider.

auto



minmax(200px, auto)



minmax(200px, auto)



minmax(200px, auto)



Nested grids

The figure is a grid item that also becomes a grid container, so we can make use of the ability to layer items on the grid.

```
figure {  
  display: grid;  
  grid-template-rows: 1fr minmax(100px,  
auto);  
}
```

```
figure img {  
  grid-row: 1 / -1;  
  grid-column: 1;  
  object-fit: cover;  
  height: 100%;  
  width: 100%;  
}
```

```
figure figcaption {  
  grid-row: 2;  
  grid-column: 1;  
  padding: 20px;  
}
```


Balloon Fiesta 2017

Despite the usual British weather, a not-insignificant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.



Several balloons take to the sky in the early morning



A nice blue striped balloon



G-BZYY and G-BSEA



Colourful

New sizing keywords from the CSS Sizing specification

- ▶ min-content
- ▶ max-content
- ▶ fit-content

min-content

Roughly, the inline size that would fit around its contents if all soft wrap opportunities within the box were taken.

```
.wrapper {  
  display: grid;  
  grid-template-columns: min-content 1fr 1fr;  
  grid-gap: 20px;  
}
```

min-content

1fr

1fr

I am
sized
with
min-
content

2

3

4

5

6

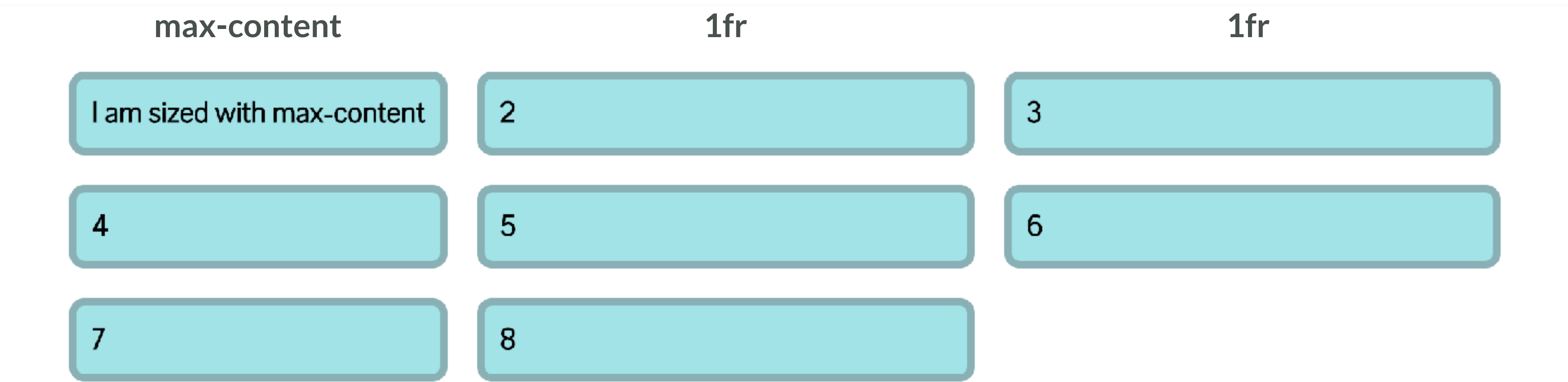
7

8

max-content

Usually the narrowest inline size it could take while fitting around its contents if none of the soft wrap opportunities within the box were taken.

```
.wrapper {  
  display: grid;  
  grid-template-columns: max-content 1fr 1fr;  
  grid-gap: 20px;  
}
```

fit-content

If the available space in a given axis is finite, equal to $\min(\text{max-content size}, \text{max}(\text{min-content size}, \text{stretch-fit size}))$. Otherwise, equal to the max-content size in that axis.

```
.wrapper {  
  display: grid;  
  grid-template-columns: fit-  
content(200px) fit-content(200px) 1fr;  
  grid-gap: 20px;  
}
```

fit-content(200px)

fit-content(200px)

1fr

I am sized with fit-content, so is the column to my right

2

3

4

5

6

7

8

Dealing with old browsers

CSS is here
to help

Float & clear properties

Have no effect on a grid item. You can float an item for old browsers then try it into a grid item for new ones.

```
.grid > div {  
  float: left;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```

```
.grid > div {  
  // I'm now a grid item, and act as if I  
  am not floated!  
}
```

Display: inline-block

An inline-block item that becomes a grid item loses any attributes that would apply when it was inline-block.

```
.grid > div {  
  display: inline-block;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```

```
.grid > div {  
  // I'm now a grid item, inline-block  
behaviour such as preserving white space is  
gone.  
}
```

Display: table

Anonymous element creation does not happen if an item with `display: table-cell` or another `table-*` property becomes a grid item.

```
.grid > div {  
  display: table-cell;  
  vertical-align: top;  
}
```

```
.grid {  
  border-spacing: 10px;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```

vertical-align

Can be used as a fallback for the Box Alignment properties in inline-block or table layout and stops applying when the item becomes a grid item.

```
.grid > div {  
  display: inline-block;  
  vertical-align: top;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```


Multiple-column layout

Can be used as a fallback for some grid layouts, and the column-* properties cease to apply once the container becomes a grid container.

```
.grid {  
  column-count: 3;  
  width: 500px;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
}
```

Flex Layout

Flexbox can also be used as a fallback, making things easier as they both use the Box Alignment properties.

```
.grid {  
  display: flex;  
  align-items: center;  
  width: 500px;  
  height: 200px;  
  border: 1px dotted #694486;  
}
```

```
.grid > div {  
  flex: 1;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
}
```

Use the cascade. Write code for old
browsers then code for new browsers.

A problem!

The width set to make the floated item the right size to fit into our layout will be interpreted on the grid item as a percentage of the grid track.

```
.grid > div {  
  float: left;  
  width: 33.333%;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, 1fr);  
  width: 500px;  
}
```

CSS Feature Queries - CR

CSS Feature Queries allow authors to condition rules based on whether particular property declarations are supported in CSS using the @supports at rule.

Global92.38%

www.rachelandrew.co.uk82.42%

U.K.89.46%

- Current aligned
- Usage relative
- Date relative
- Show all

IE	Edge*	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
			49						
		52	58			9.3		4.4	
	14	54	59		46	10.2		4.4.4	
11	15	55	60	10.1	47	10.3	all	56	59
	16	56	61	11	48	11			
		57	62	TP	49				
		58	63						

- Notes
- Known issues (2)
- Resources (7)
- Feedback

See also the [CSS.supports\(\) DOM API](#)

Feature Queries

Test for support of a property and value.
Inside the Feature Query add code only
for browsers that claim support.

```
.grid > div {  
  float: left;  
  width: 33.333%;  
}
```

```
@supports (display: grid) {  
  .grid > div {  
    width: auto;  
  }  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, 1fr);  
  width: 500px;  
}
```

You need to understand CSS

The fundamentals of CSS haven't
changed

Next for Grid

Subgrids?

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title



The contents.

Footer contents

CSS Grid

Creating a three column layout on the parent element with fr units.

```
.grid {  
  display: grid;  
  max-width: 960px;  
  margin: 0 auto;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 20px;  
}
```


My title



The contents.

Footer contents

My title



The contents. I have a lot of content, more content than the other ones.

Footer contents

My title



The contents.

Footer contents, I have a mahoosive footer. Just look at this stuff.

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title this one is very long indeed



The contents.

Footer contents

Make the card a flex item

Allow the inner to grow, it pushes the footer down to the bottom of the cards

```
.card {  
  display: flex;  
  flex-direction: column;  
}  
  
.card .inner {  
  flex: 1;  
}
```


My title



The contents.

Footer contents

My title



The contents. I have a lot of content, more content than the other ones.

Footer contents

My title



The contents.

Footer contents, I have a mahoosive footer. Just look at this stuff.

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title this one is very long indeed



The contents.

Footer contents

1

My title

2



3

The contents. I have a lot of content, more content than the other ones.

4

Footer contents

5

display: subgrid

The card is a direct child of the grid so needs to span four rows of the grid to make room for the four rows in the subgridded internals.

display: subgrid means the card now uses the tracks of the parent grid.

```
.card {  
  border: 4px solid rgb(24,154,153);  
  background-color: #fff;  
  grid-row: auto / span 4;  
  display: subgrid;  
}
```


My title



The contents.

Footer contents

My title



The contents. I have a lot of content, more content than the other ones.

Footer contents

My title



The contents.

Footer contents, I have a mahoosive footer. Just look at this stuff.

My title



The contents.

Footer contents

My title



The contents.

Footer contents

My title this one is very long indeed



The contents.

Footer contents

Subgrid Links & Thoughts

- ▶ <https://rachelandrew.co.uk/archives/2017/03/16/subgrid-moved-to-level-2-of-the-css-grid-specification/>
- ▶ <https://github.com/w3c/csswg-drafts/issues/958>
- ▶ <https://github.com/rachelandrew/cssgrid-ama/issues/13>
- ▶ <http://meyerweb.com/eric/thoughts/2016/01/15/subgrids-considered-essential/>

Next for Grid

Masonry Layout

Balloon Fiesta 2017

Despite the usual British weather, a not-insignificant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.



Using auto-placement

Allowing items to auto-place with certain items spanning rows and columns.

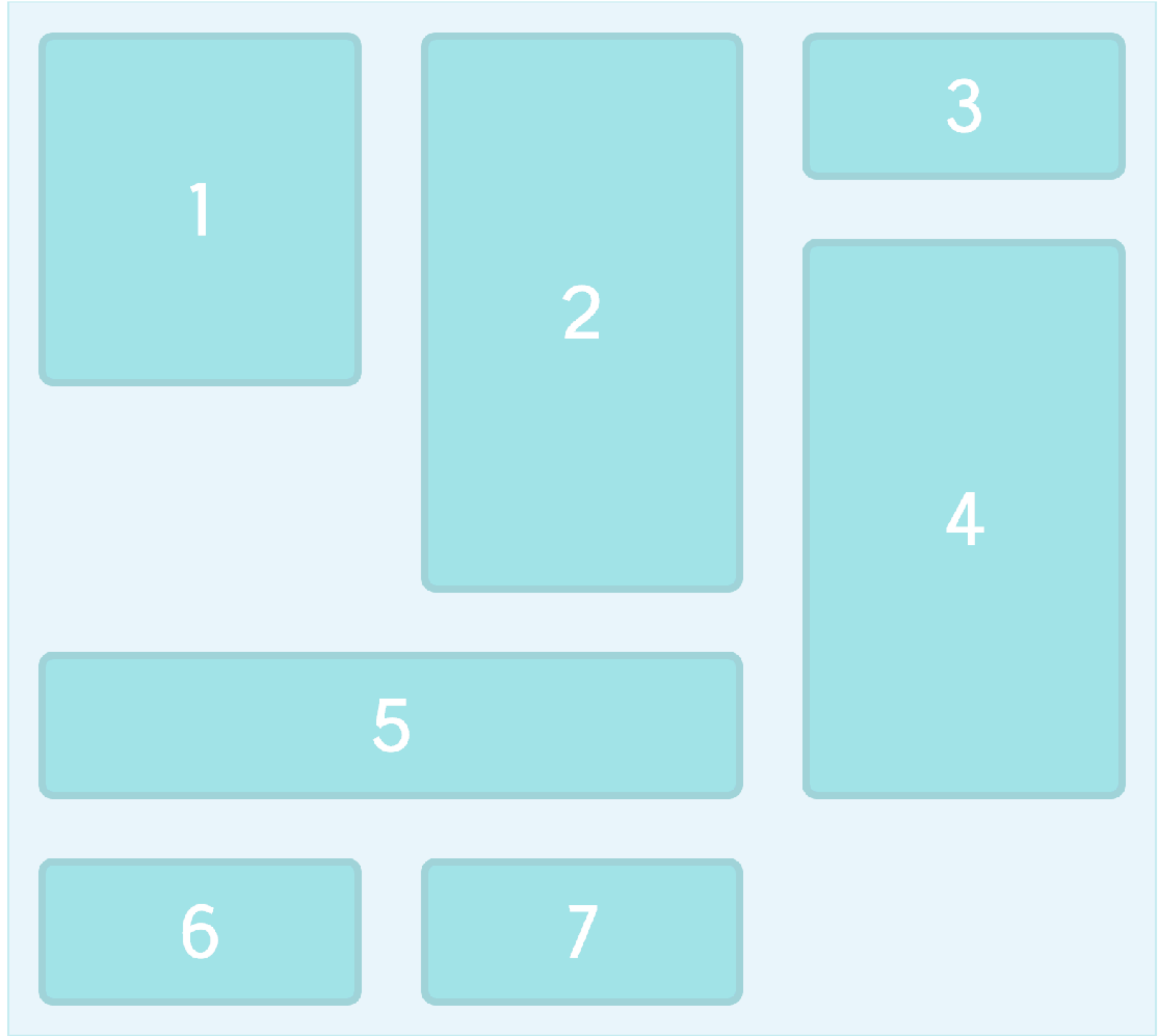
```
.grid {  
  display: grid;  
  grid-gap: 40px;  
  grid-template-columns: repeat(auto-fill,  
minmax(160px, 1fr));  
  grid-auto-rows: minmax(100px, auto);  
}
```

```
.grid > div:nth-child(1) {  
  grid-row-end: span 2;  
}
```

```
.grid > div:nth-child(2) {  
  grid-row-end: span 3;  
}
```

```
.grid > div:nth-child(4) {  
  grid-row-end: span 3;  
}
```

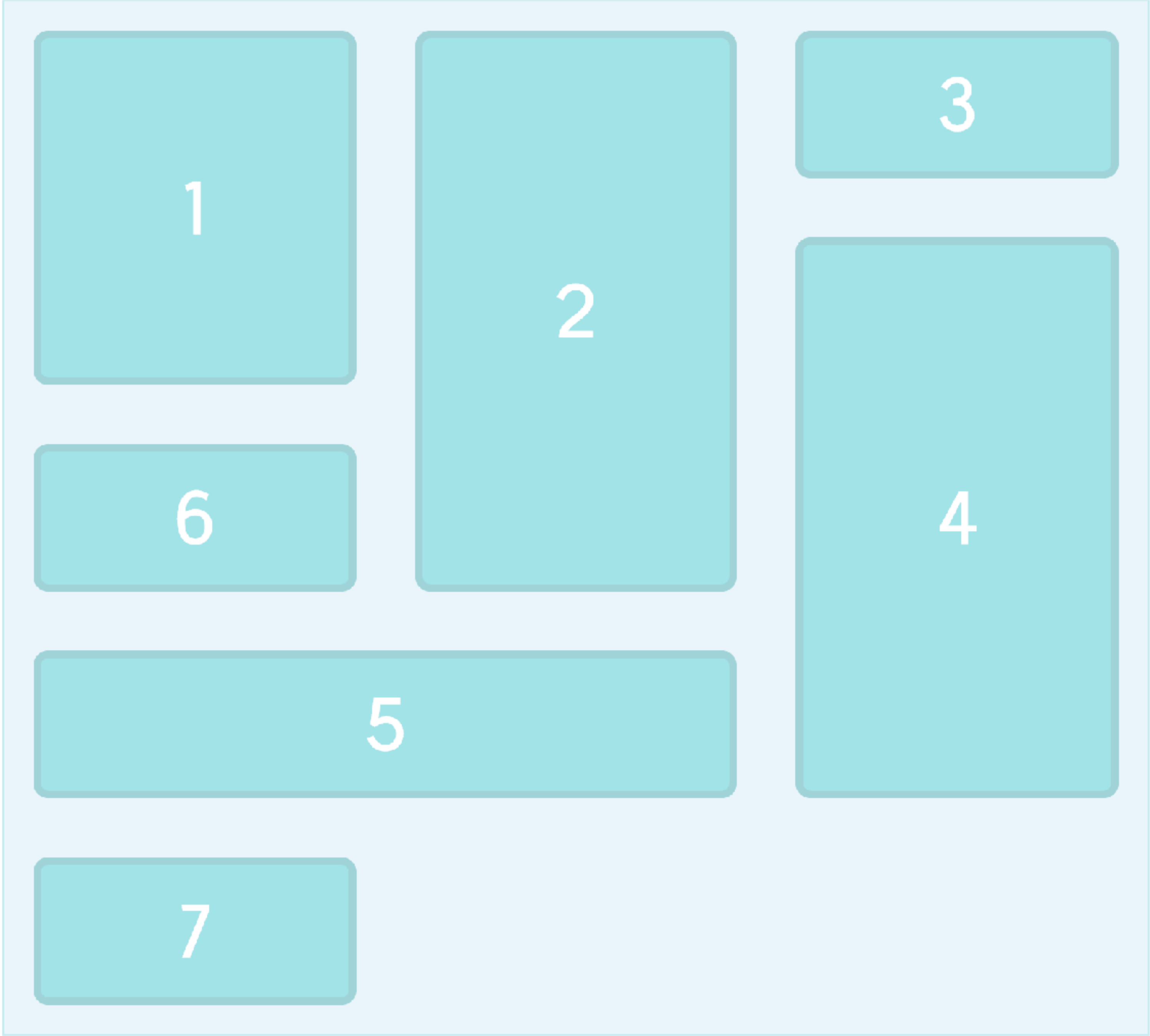
```
.grid > div:nth-child(5) {  
  grid-column-end: span 2;  
}
```

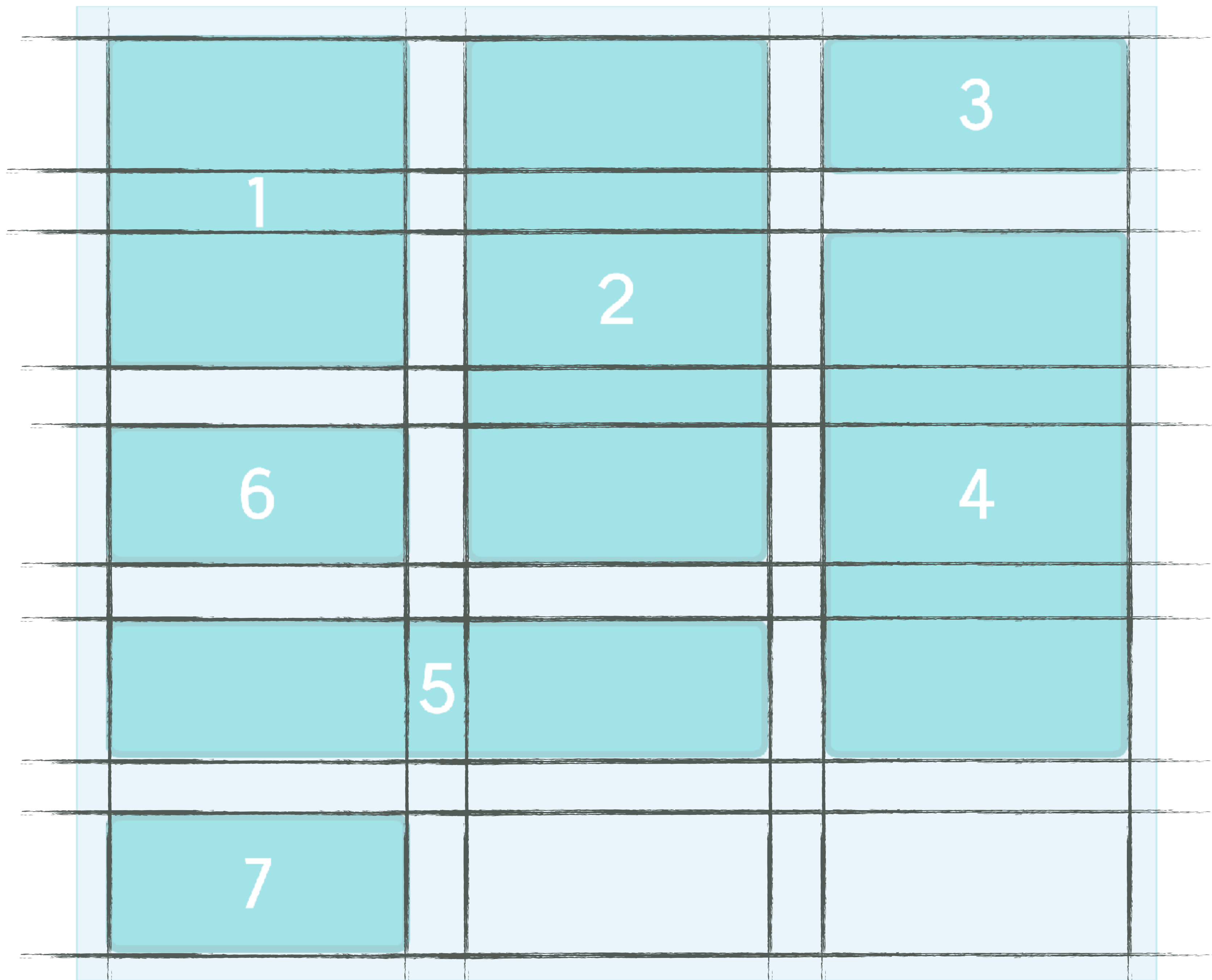



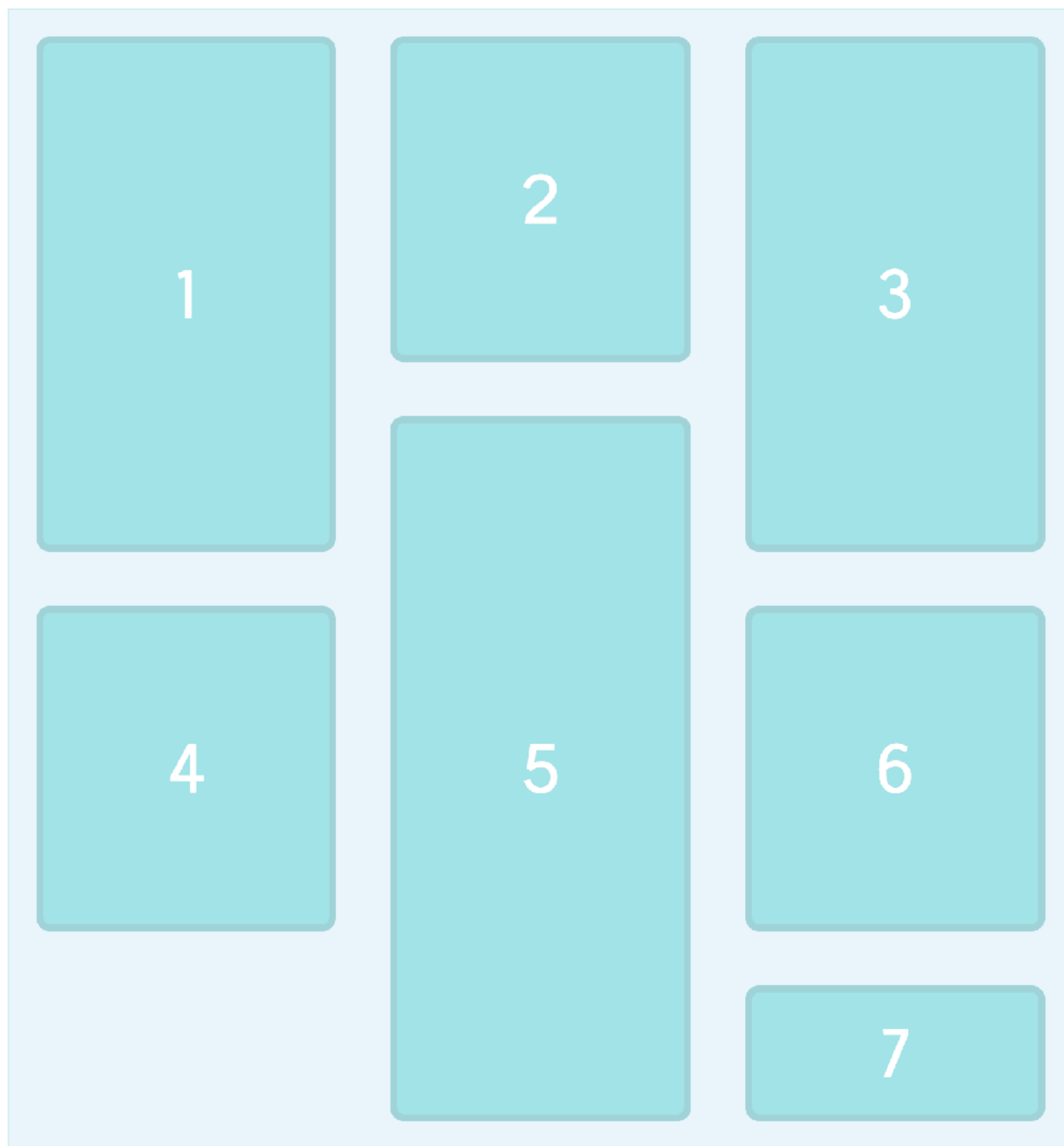
Set auto-flow to dense

Grid will backfill gaps taking items out of document order visually.

```
.grid {  
  display: grid;  
  grid-gap: 40px;  
  grid-template-columns: repeat(auto-fill, minmax(160px, 1fr));  
  grid-auto-rows: minmax(100px, auto);  
  grid-auto-flow: dense;  
}
```





Masonry Layouts

- ▶ <https://rachelandrew.co.uk/archives/2017/01/18/css-grid-one-layout-method-not-the-only-layout-method/>
- ▶ <https://github.com/w3c/csswg-drafts/issues/945>

This is all new.

We are all learning.

**Solve problems and
share what you find out.**

Grid Resources

- ▶ Visit Grid by Example for worked examples, patterns with fallbacks, and a free video tutorial: <https://gridbyexample.com>
- ▶ I created a huge set of guides for MDN: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout
- ▶ Over 5 years of grid thoughts on my site at: <https://rachelandrew.co.uk/archives/tag/cssgrid>
- ▶ GridBugs! I'm collecting and trying to get fixed interop issues: <https://github.com/rachelandrew/gridbugs>

Rachel Andrew

THE NEW CSS LAYOUT

FOREWORD BY Jen Simmons

Published this week!

The New CSS Layout

@rachelandrew

Slides & Resources: <https://rachelandrew.co.uk/speaking/event/mirrorconf>

Thank you!