Rachel Andrew @ #DevFest17

Solving Layout problems with CSS Grid & friends

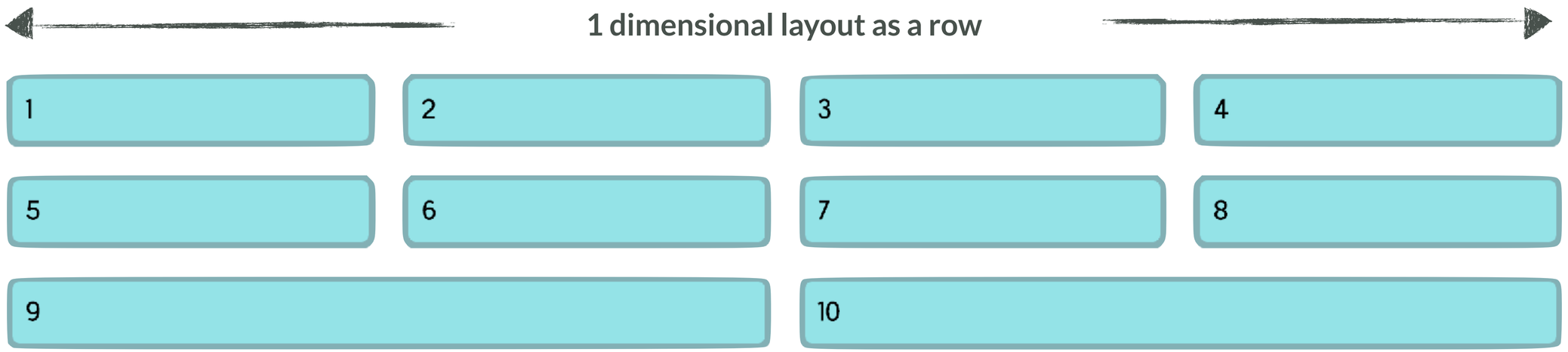
Rachel Andrew - I do stuff on the web.

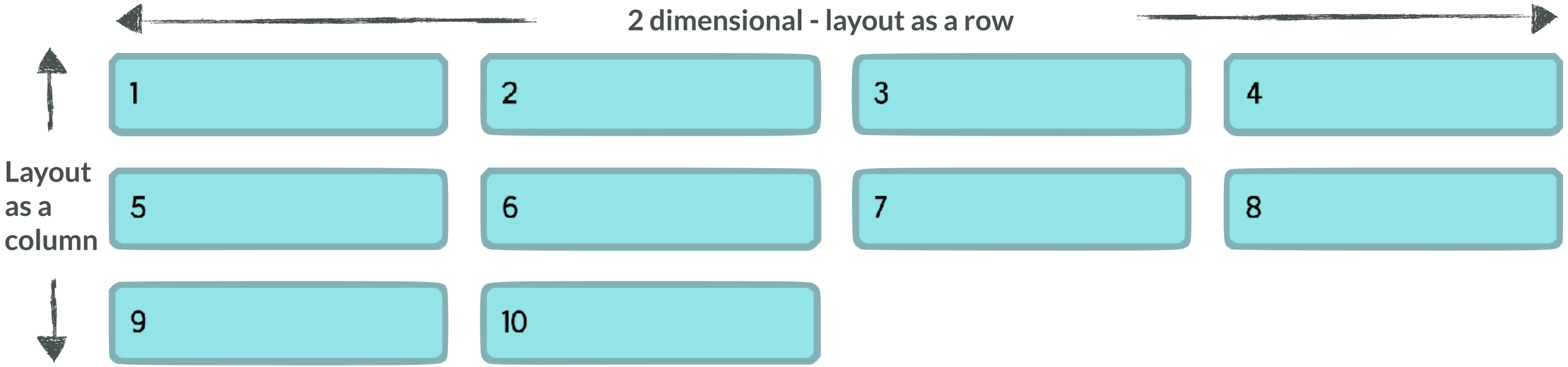
- @rachelandrew on Twitter
- https://rachelandrew.co.uk
- Invited Expert to the CSS Working Group
- Google Developer Expert
- co-founder Perch and Perch Runway CMS - https://grabaperch.com
- Smashing Magazine Editor in Chief

So, about this Grid thing …

… why not use Flexbox?

Do you need layout in one dimension or two?


1 dimensional layout as a row
1
2
3
4
5
6
7
8
9
10


2 dimensional - layout as a row
Layout as a column
1
2
3
4
5
6
7
8
9
10

Grid works from the container in

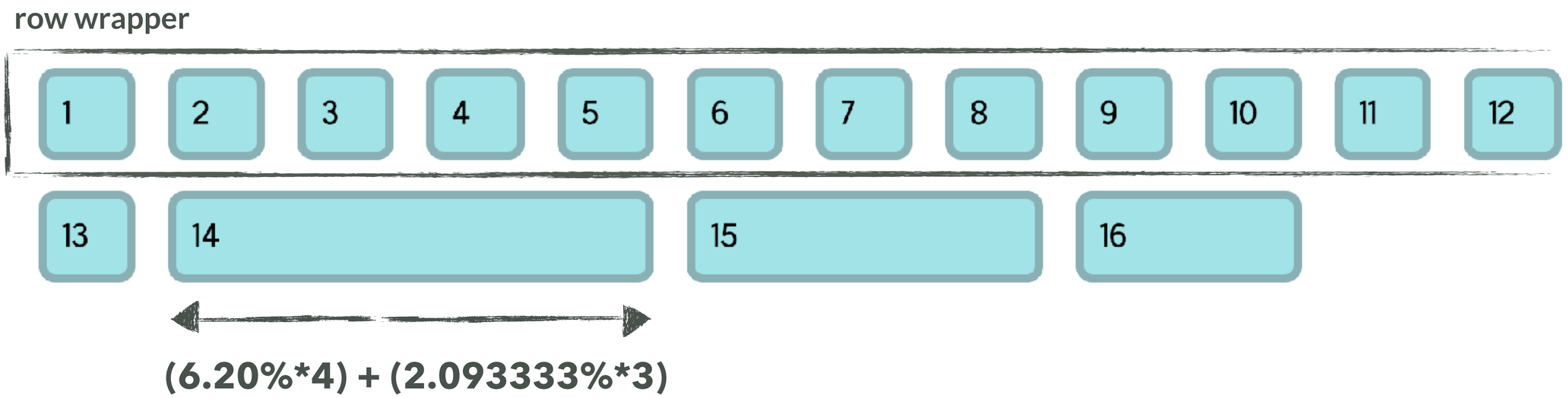
Every other method of creating a grid, involves sizing the individual items.

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.

https://codepen.io/rachelandrew/pen/brjymK

```
.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
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
(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.

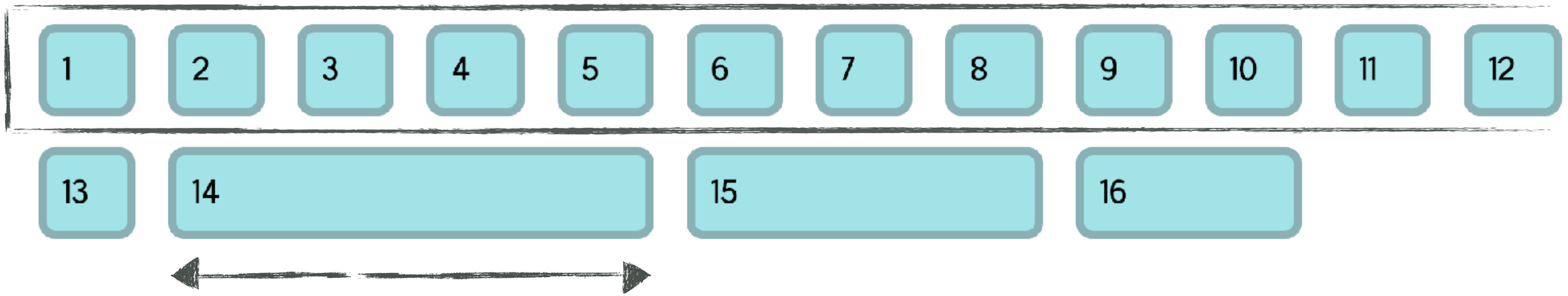
A flexbox “grid”

Using the width as the flex-basis.

https://codepen.io/rachelandrew/pen/KvBLbJ

```
.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



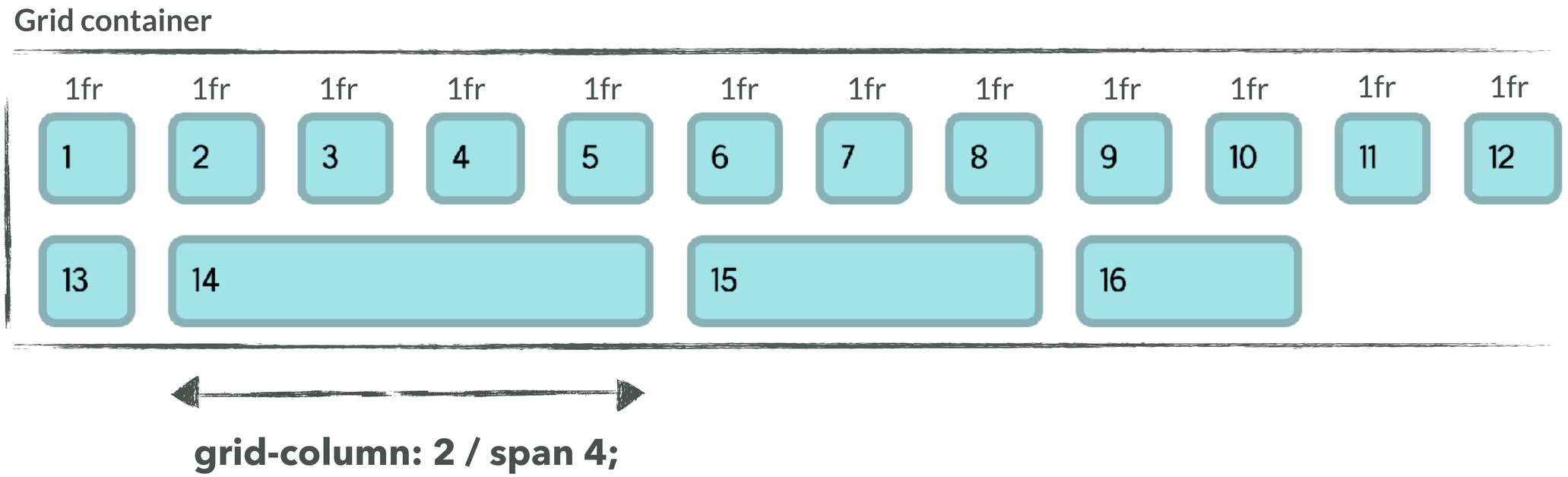
(6.20%*4) + (2.093333%*3)

A Grid … grid

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

https://codepen.io/rachelandrew/pen/VzBOJW

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

.col.span2 {
  grid-column: auto / span 2;
}
```


Grid container
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
grid-column: 2 / span 4;

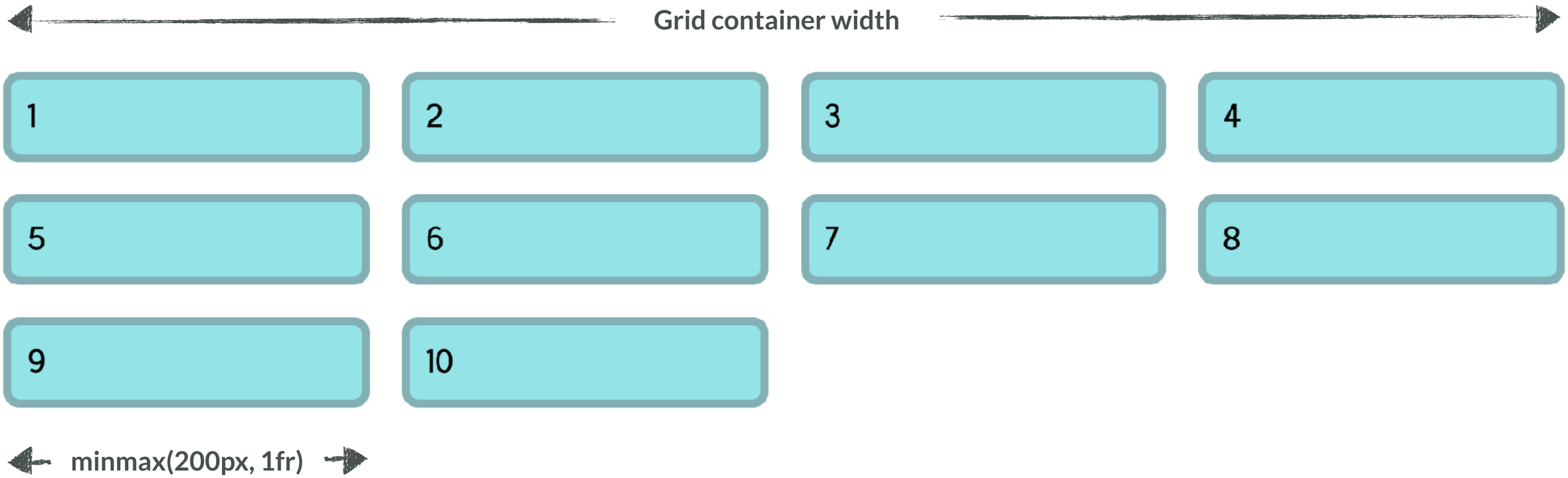
Grid frameworks create something that looks like a grid by controlling item size.

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


Grid container width
1
2
3
4
5
6
7
8
9
10
minmax(200px, 1fr)

repeat()

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

```
.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));
}
```

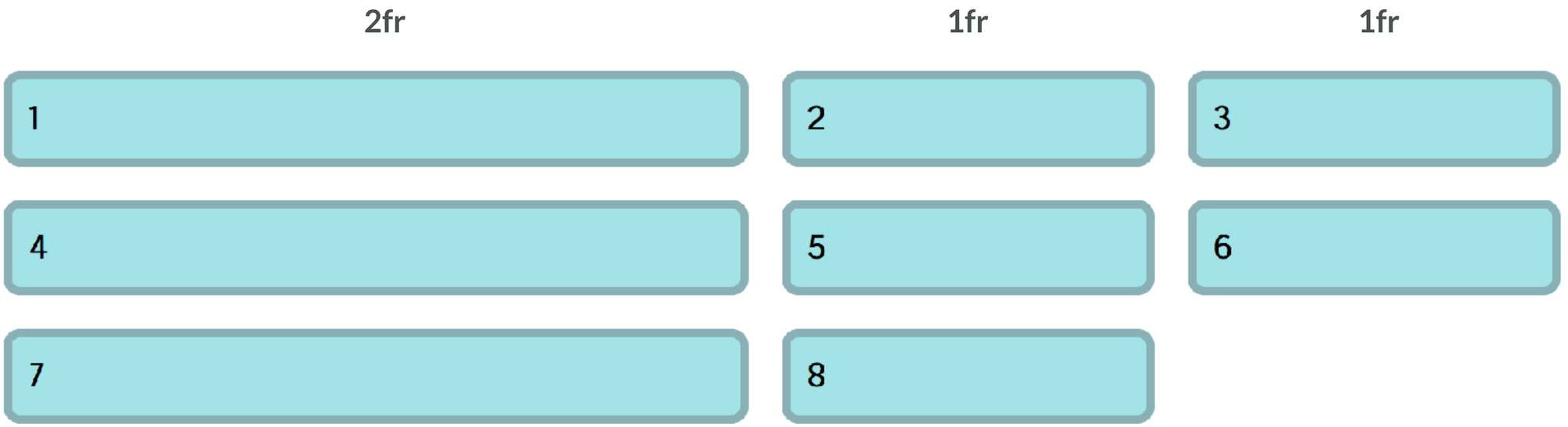
The fr unit - distributing available space

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;
}
```

https://codepen.io/rachelandrew/pen/BdeqoJ

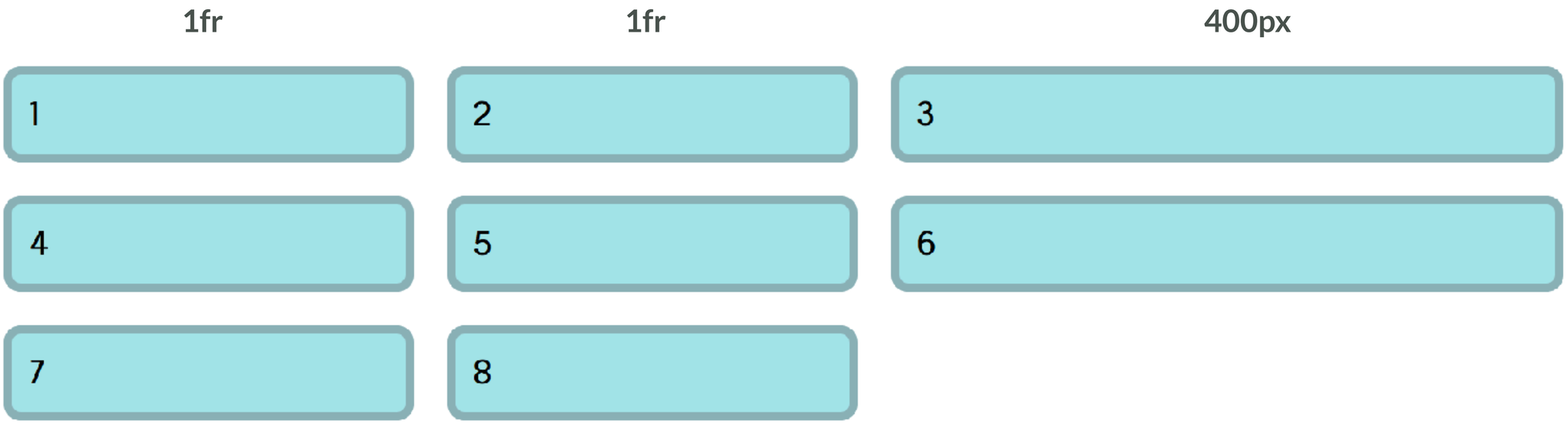

2fr
1fr
1fr
1
2
3
4
5
6
7
8

The fr unit

Mix absolute length units and fr units.

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

https://codepen.io/rachelandrew/pen/RZYZad

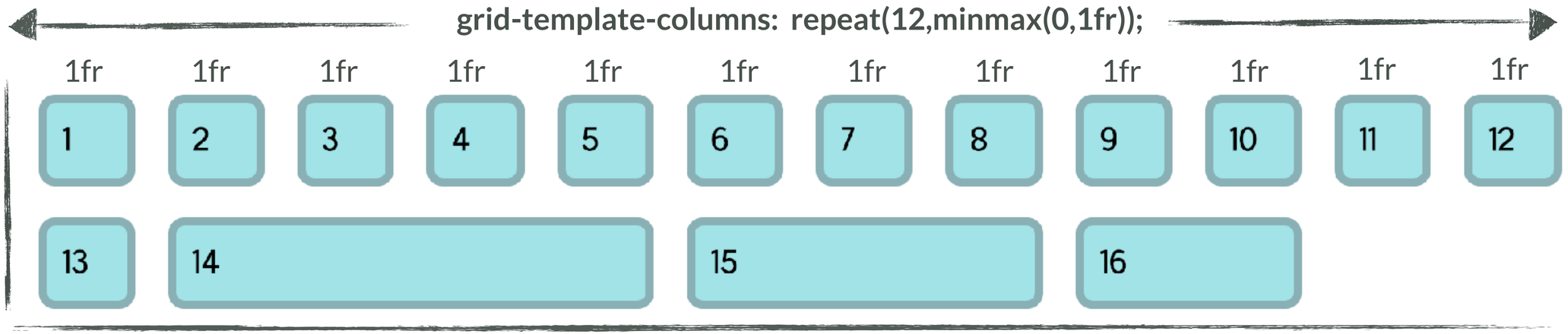

1fr
1fr
400px
1
2
3
4
5
6
7
8

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;
}
```

https://codepen.io/rachelandrew/pen/VzBOJW


grid-template-columns: repeat(12,minmax(0,1fr));
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1fr
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

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-insignifiant 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()

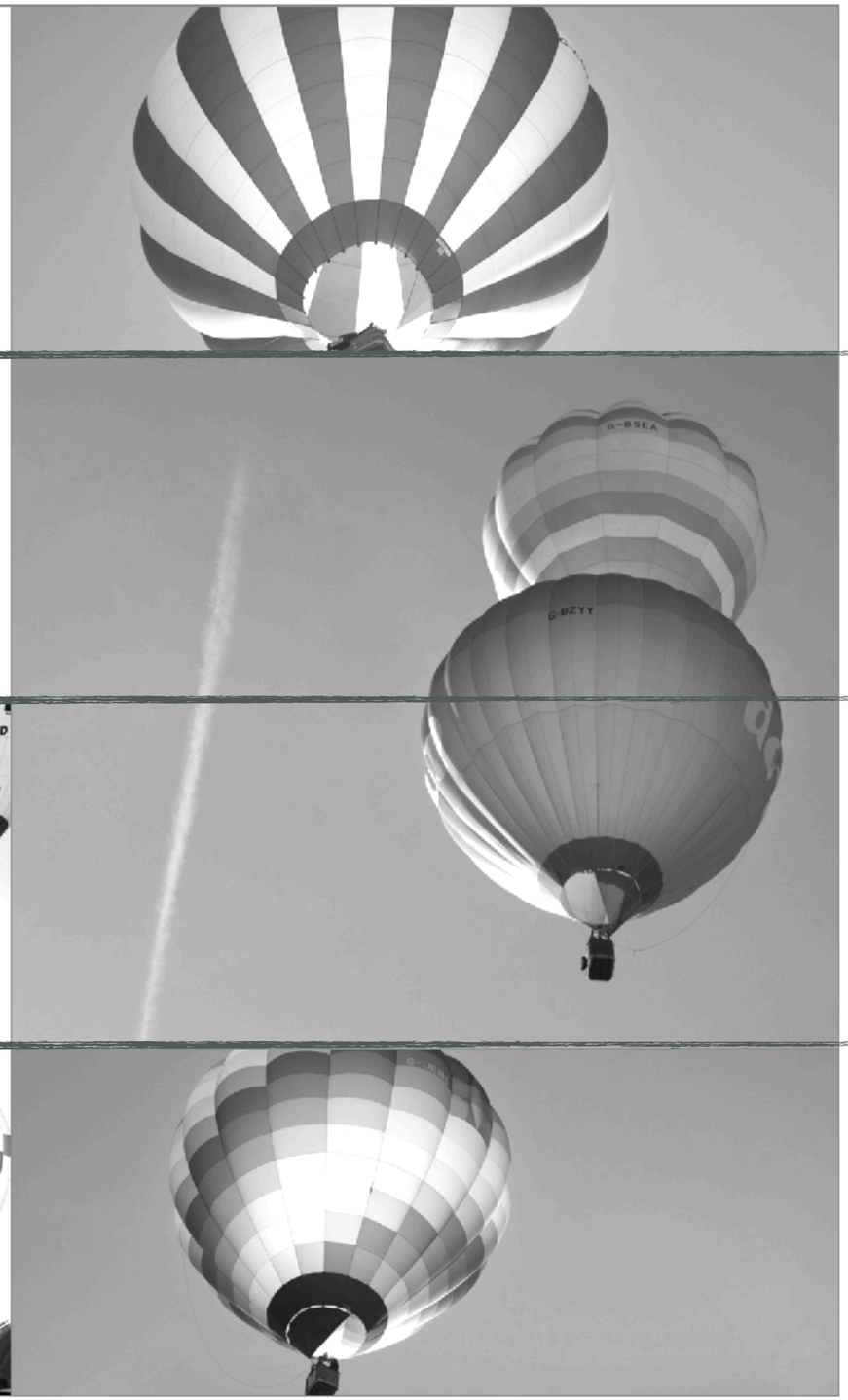
Row tracks will be 200 pixels tall *unless* there is more content, in which case they will grow as the max is auto.

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

https://codepen.io/rachelandrew/pen/Mvqvbm

Balloon Fiesta 2017

Despite the usual British weather, a not-insignifiant 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-insignifiant 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-insignifiant 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.

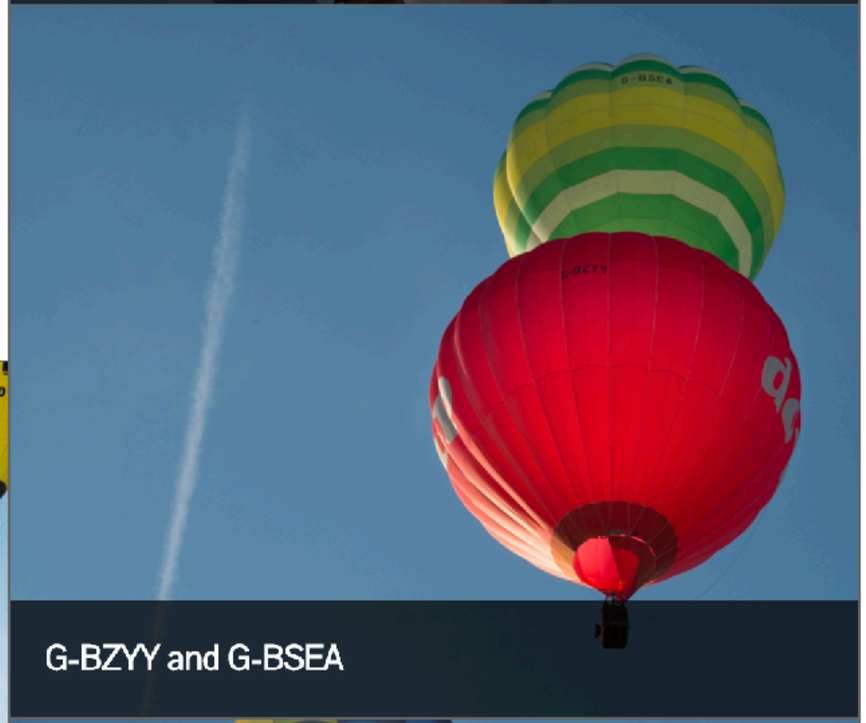
https://codepen.io/rachelandrew/pen/xLePZY

```
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-insignifiant 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.

A nice blue striped balloon



G-BZYY and G-BSEA

Several balloons take to the sky in the early morning

Colourful

New sizing keywords from the CSS Sizing specification

CSS Intrinsic & Extrinsic Sizing Module Level 3: https://www.w3.org/TR/css-sizing-3/

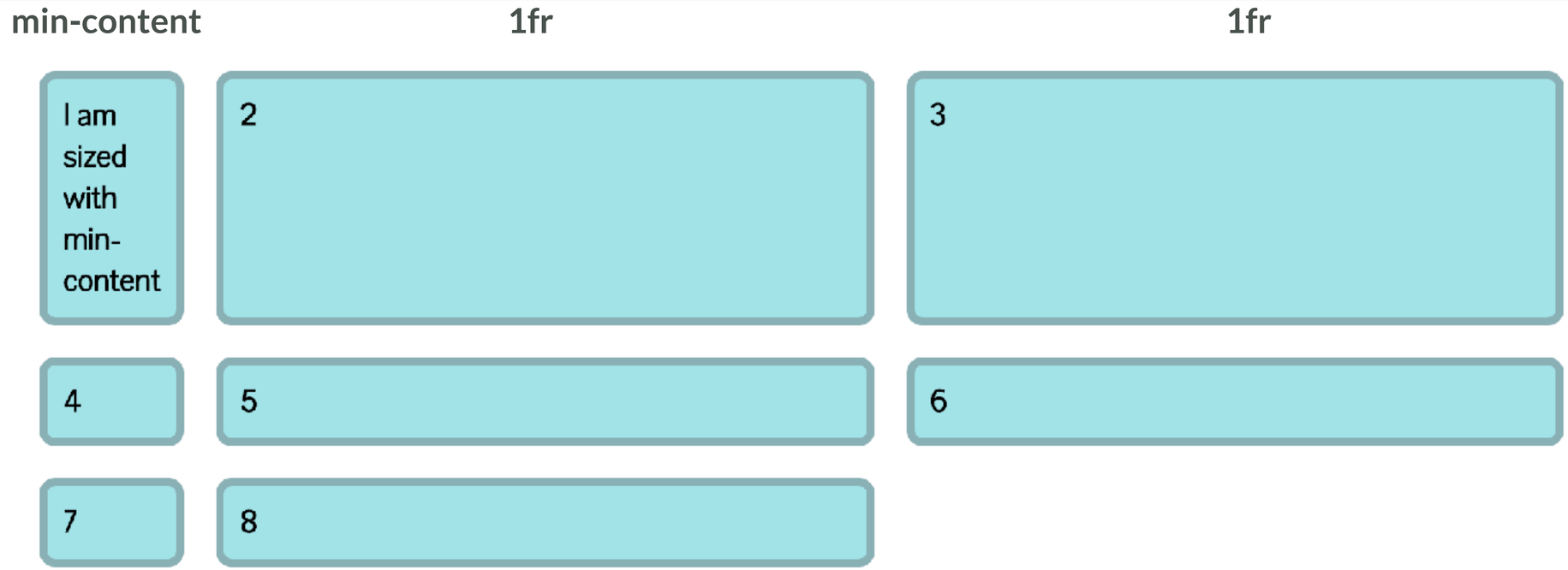
- min-content
- max-content
- fit-content

min-content

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

https://codepen.io/rachelandrew/pen/xLejpK

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

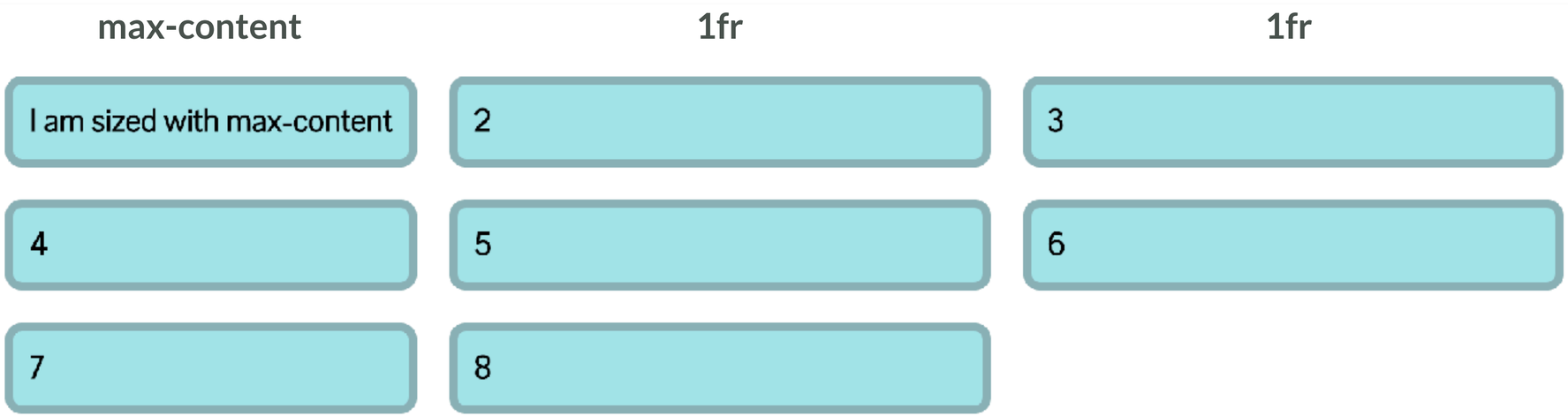

min-content
1fr
1fr
I am sized with min-content
2
3
4
5
6
7
8

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;
}
```

https://codepen.io/rachelandrew/pen/KvYRZB

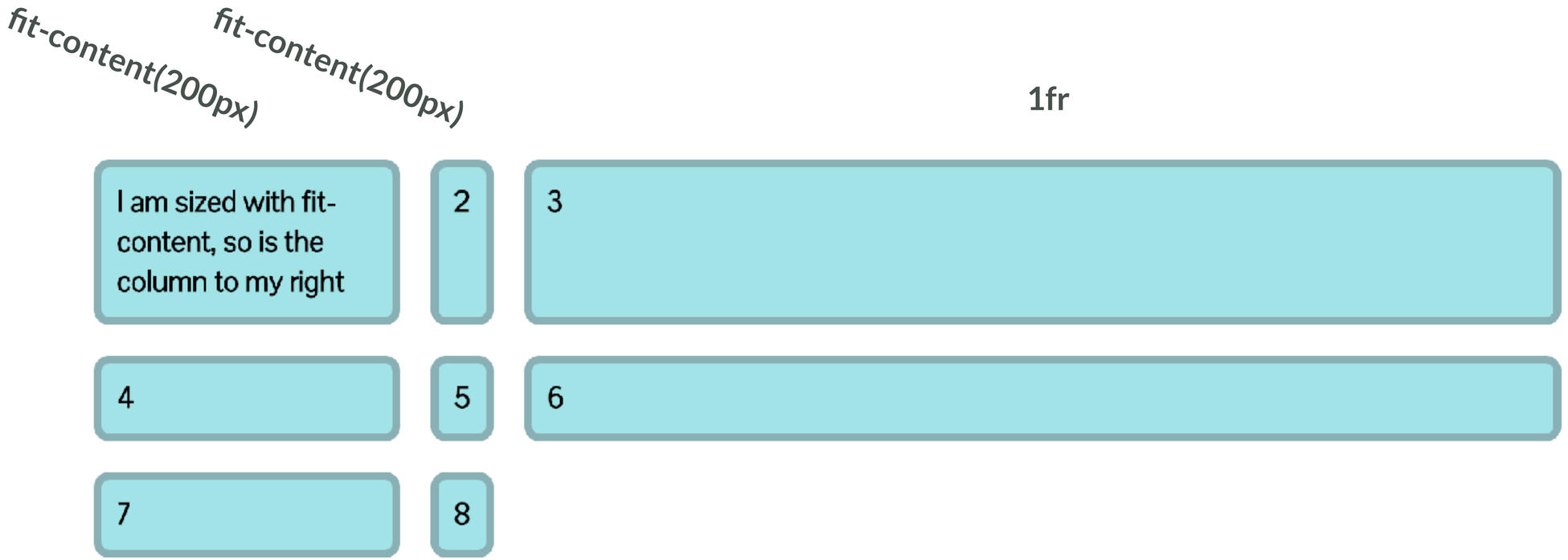

max-content
1fr
1fr
I am sized with max-content
2
3
4
5
6
7
8

fit-content

If the available space in a given axis is finite, equal to min(max-content size, max(min-content size,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;
}
```

https://codepen.io/rachelandrew/pen/NvLvRG


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.

https://codepen.io/rachelandrew/pen/NvmMOM

```
.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.

https://codepen.io/rachelandrew/pen/LjvmXG

```
.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.

https://codepen.io/rachelandrew/pen/OjGZaO

```
.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.

https://codepen.io/rachelandrew/pen/NvmMEM

```
.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.

https://codepen.io/rachelandrew/pen/MvRGzL

```
.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.

https://codepen.io/rachelandrew/pen/MvRGzx

```
.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

Global 92.59%

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

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	60			10.2			
	15	55	61	10.1		10.3		4.4	
11	16	56	62	11	48	11	all	56	61
		57	63	TP	49				
		58	64		50				
		59	65						

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.

https://codepen.io/rachelandrew/pen/ayxGPz

```
.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;
}
```

https://codepen.io/rachelandrew/pen/XgdydE

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.

https://codepen.io/rachelandrew/pen/ZyWmVM

```
.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.

My title

The contents.

My title this one is very long indeed

The 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

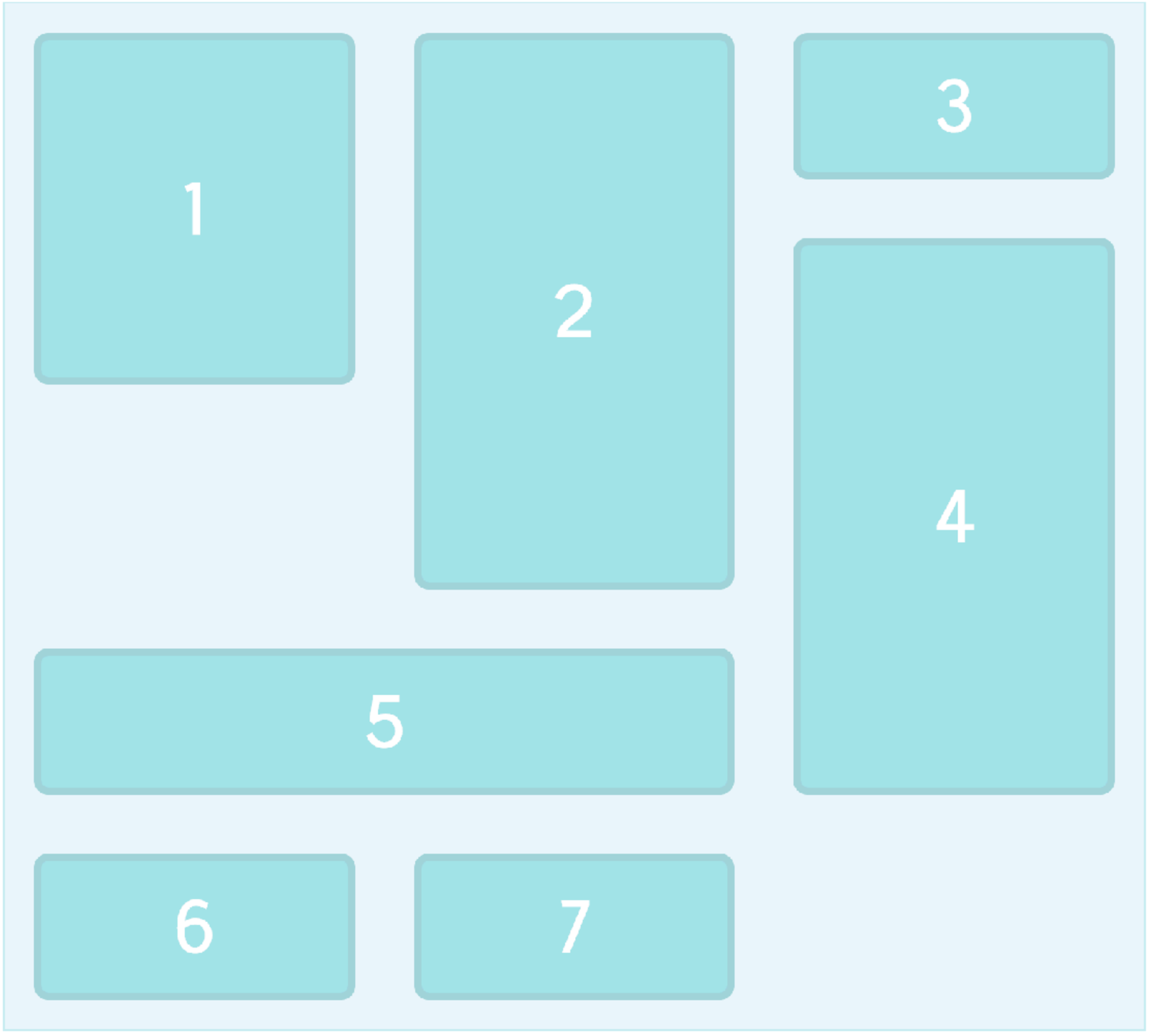
Despite the usual British weather, a not-insignifiant number of hot air balloons took to the Bristol skies. Those of us who think 6am a suitable time to be roaming around with a camera got some nice photos.

Using auto-placement

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

https://codepen.io/rachelandrew/pen/NvmZeR

```
.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;
}
```

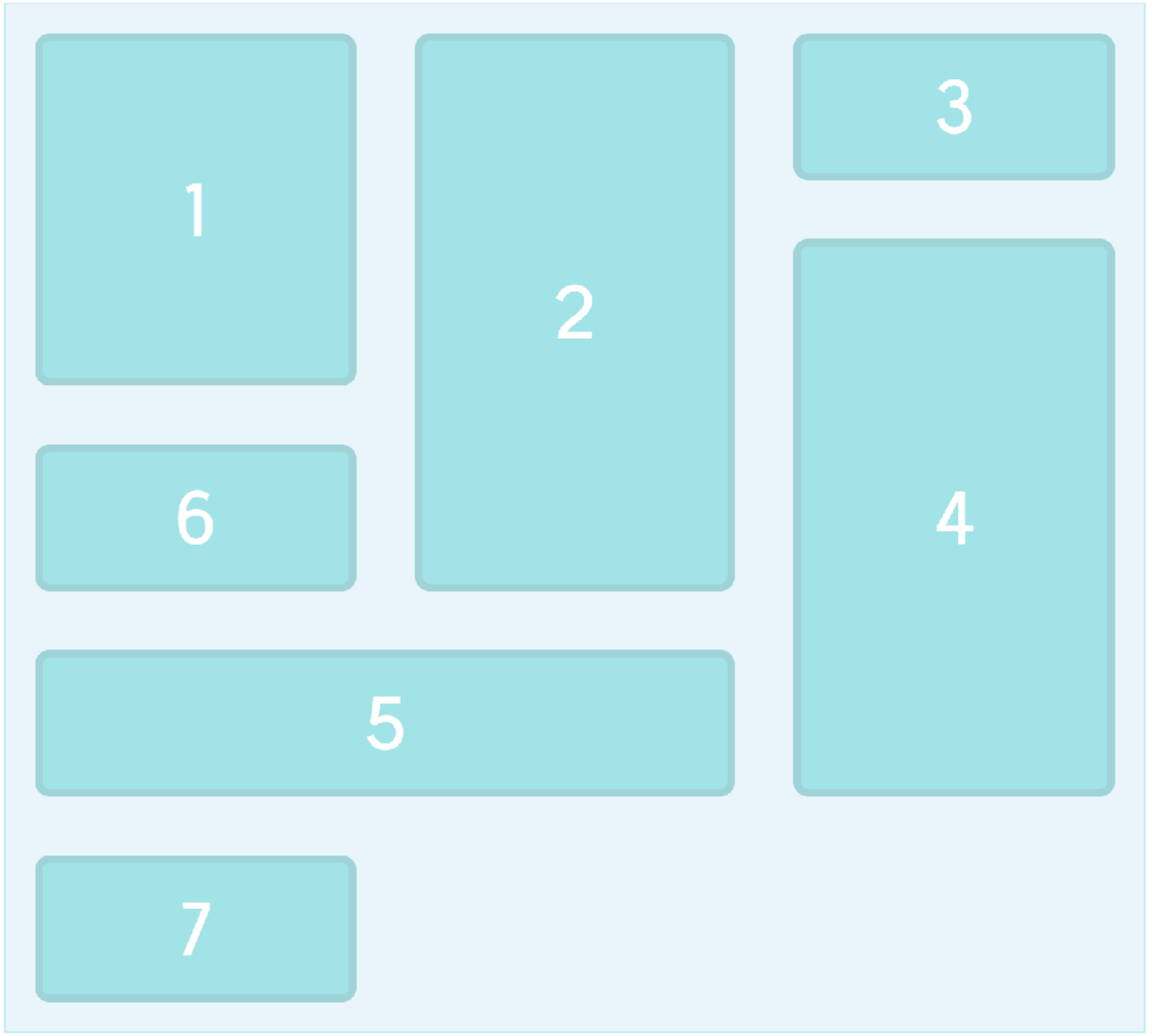

1
2
3
4
5
6
7

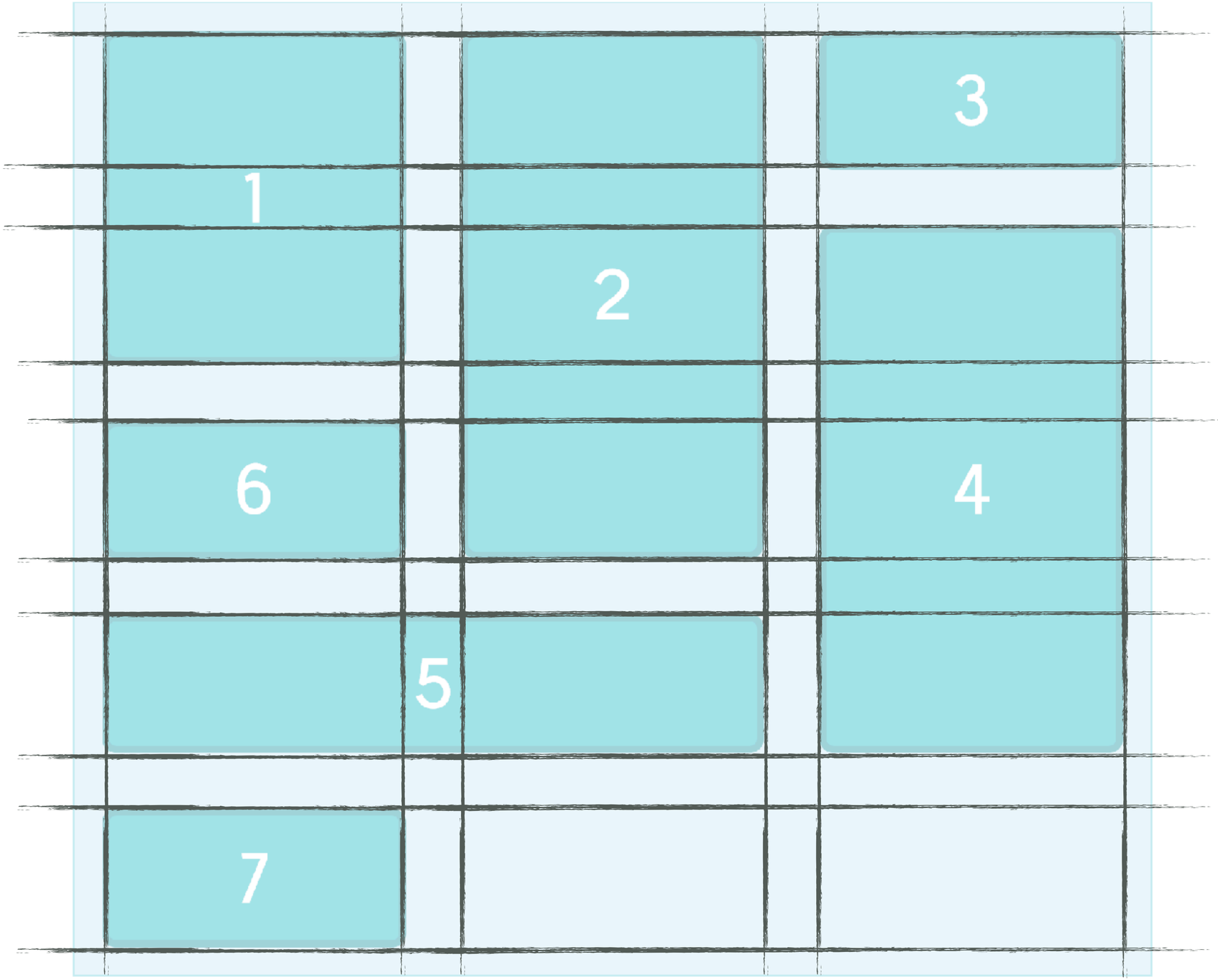
Set auto-flow to dense

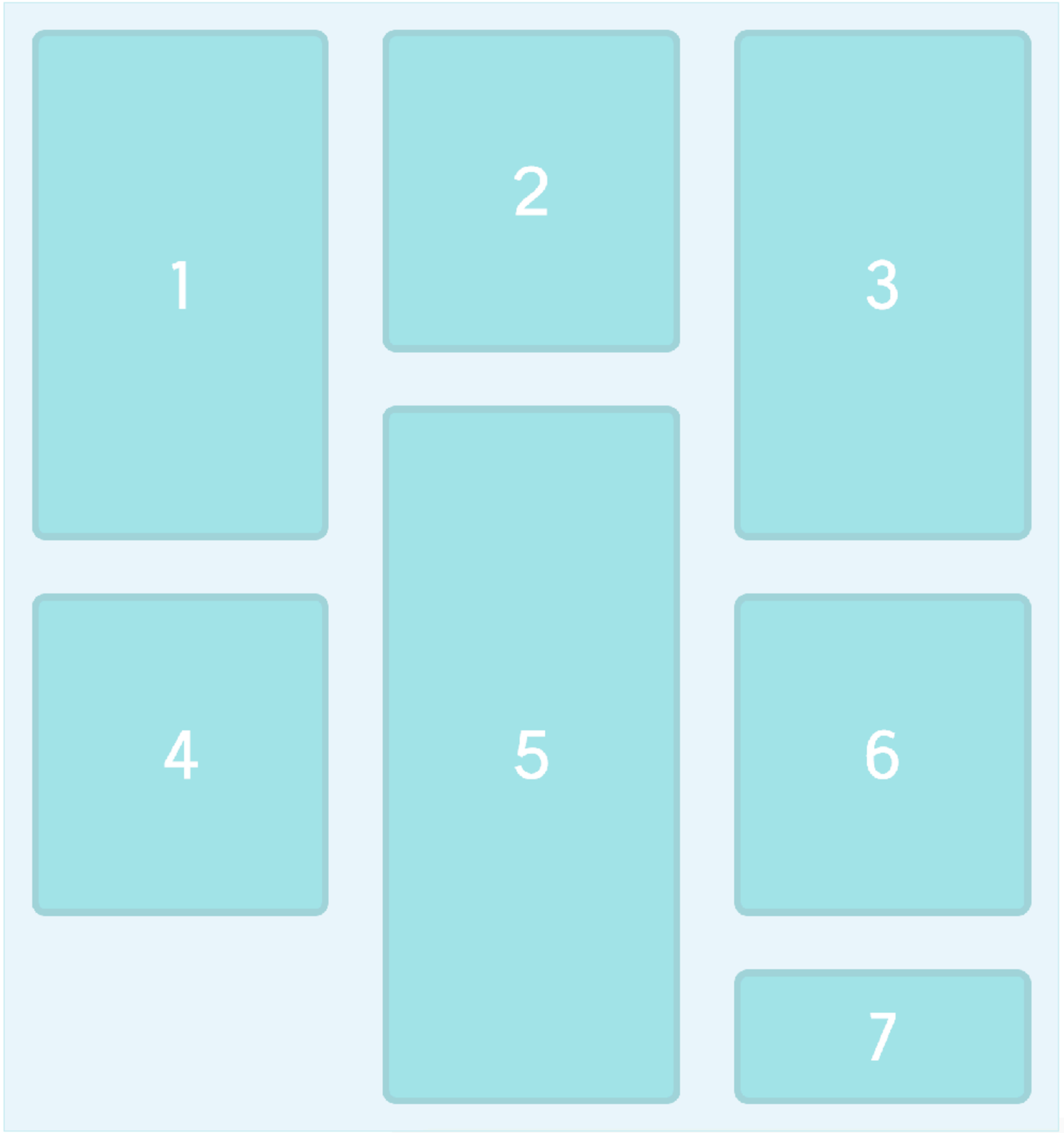
Grid will backfill gaps taking items out of document order visually.

```
.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;
}
```

https://codepen.io/rachelandrew/pen/WEWqWR


1
2
3
4
5
6
7


3
1
2
6
4
5
7


1
2
3
4
5
6
7

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

Next for Grid

Styling grid cells

[css-grid] Decorative grid-cell pseudo-elements #499

Edit New issue

Open **tabatkins** opened this issue on 19 Sep 2016 · 6 comments

tabatkins commented on 19 Sep 2016 Member

@jensimmons proposes adding a grid-cell pseudo so you can add (responsive) decorative elements to grids without having to add empty elements to your page.

Syntax like:

```
#grid::grid-area(1 / 2 / 3 / 4) {
  background-color: red; /* etc */
  /* grid-positioning properties are blacklisted */
}
```

Also some way of controlling whether or not that takes up auto-flow space. Might want to reserve space or not. Can we base this on 'content' or something? Or do we need a new property?

👍 4

tabatkins added the css-grid-2 label on 19 Sep 2016

jensimmons commented on 19 Sep 2016 Member

The slides from the presentation I just gave are at: https://speakerdeck.com/jensimmons/proposal-to-csswg-sept-2016

And actually, the syntax above is not what I proposed. This is what I proposed:

Assignees

No one—assign yourself

Labels

css-grid-2

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this repository.

7 participants

Lock conversation

Balloons in Bristol

Despite the fact that British weather is not ideal for modes of transport that are unusable during any significant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our content column and full width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

The layout

Using grid-template-areas - the ascii-art method of laying things out on the grid.

https://codepen.io/rachelandrew/pen/qjdzwR

```
.grid {
  display: grid;
  grid-template-columns: minmax(1em,
1fr) minmax(0, 660px) minmax(1em, 1fr);
  grid-template-areas:
    ". title ."
    ". content-top ."
    "full-width full-width full-width"
    ". content-bottom ."
}

h1 { grid-area: title; }

.content1 { grid-area: content-top; }

.content2 { grid-area: content-bottom; }

.gallery { grid-area: full-width; }
```

Balloons in Bristol

Despite the fact that British weather is not ideal for modes of transport that are unusable during any significant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our content column and full width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

Balloons in Bristol

Despite the fact that British weather is not ideal for modes of transport that are unusable during insignificant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our full-width and full-width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

Named Areas create Named Lines

Each named area creates 4 lines. These are named with the name of the area plus -start and -end for columns and rows.

The area **title** has **title-start** and **title-end** for row and column.

https://codepen.io/rachelandrew/pen/qjdzwR

```
.grid {
  display: grid;
  grid-template-columns: minmax(1em,
1fr) minmax(0, 660px) minmax(1em, 1fr);
  grid-template-areas:
    ". title ."
    ". content-top ."
    "full-width full-width full-width"
    ". content-bottom ."
}

h1 { grid-area: title; }

.content1 { grid-area: content-top; }

.content2 { grid-area: content-bottom; }

.gallery { grid-area: full-width; }
```

title-start

Balloons in Bristol

title-end content-top-start

Despite the fact that British weather is not ideal for modes of transport that are unusable during any significant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

content-top-end full-width-start

full-width-start

title-start content-top-start content-bottom-start

title-end content-top-end content-bottom-end

full-width-end

full-width-end content-bottom-start

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our content column and full width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

content-bottom-end

title-start

Balloons in Bristol

Despite the fact that British weather is not ideal for modes of transport that are unusable during any significant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

content-top-start

content-top-end

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our content column and full width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

content-bottom-end

Generated content

Positioned using the named lines created from our named areas.

https://codepen.io/rachelandrew/pen/qjdzwR

```
.grid::after {
  content: "";
  background-color: #fff;
  border: 4px solid rgb(182,222,211);
  grid-column:
    content-top-start / content-top-end;
  grid-row:
    title-start / content-bottom-end;
  z-index: -1;
}
```

Balloons in Bristol

Despite the fact that British weather is not ideal for modes of transport that are unusable during any significant weather, Bristol is known for hot air balloons. These make pleasing images to include in presentation code demos. Here are some.

The content of this demo has been placed using a mixture of named grid laines and auto-placement. The named lines mark out the start and end of our content column and full width grid. We have not specified rows, and are using auto-placement which will place the items in a new row.

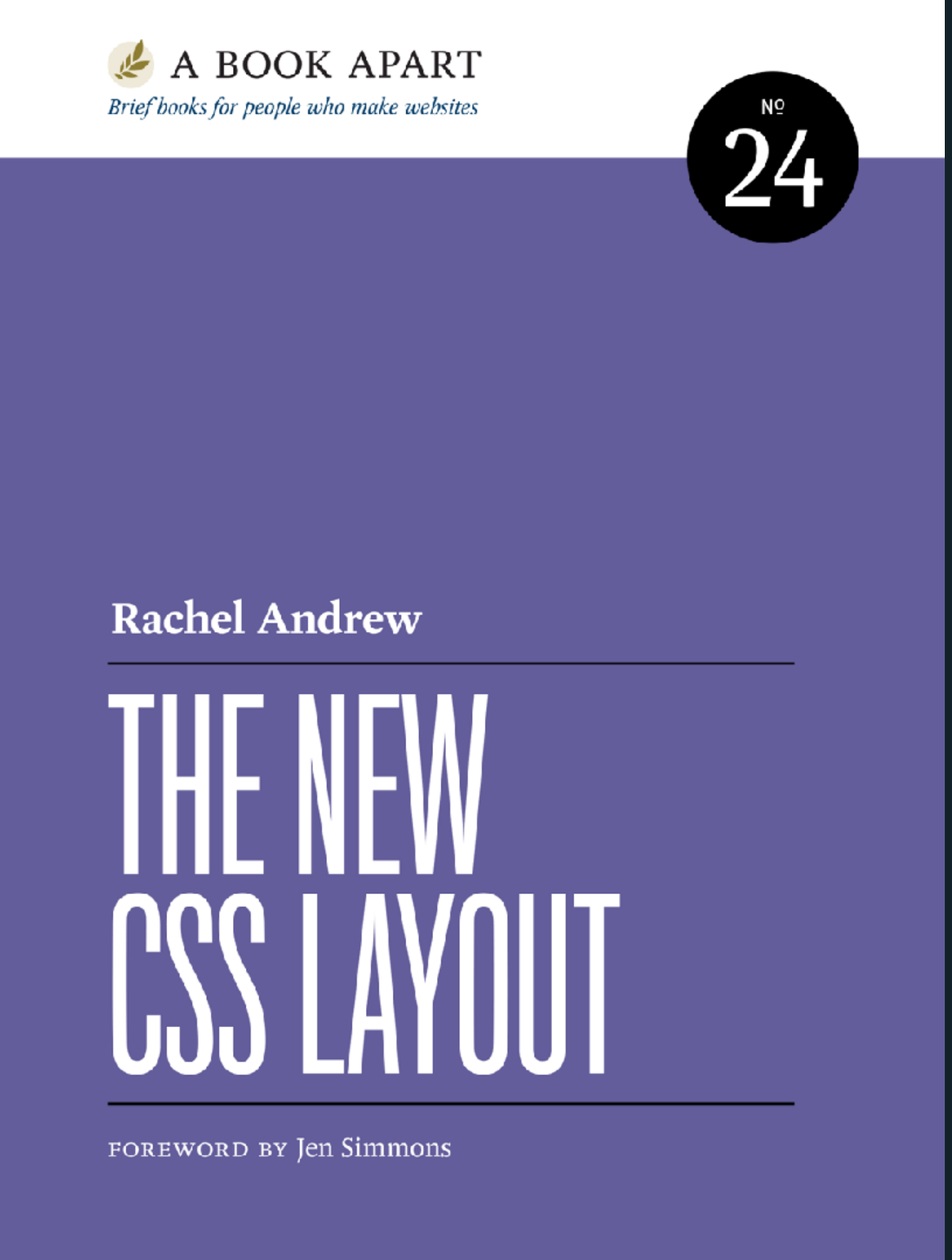
This is all new.

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



Out now!

The New CSS Layout

@rachelandrew

Slides & Resources: https://rachelandrew.co.uk/speaking/event/gdg-london-2017

Thank you!