

2017

Rachel Andrew @ View Source London

Solving Layout problems
with CSS Grid & friends

Rachel Andrew - I do things on the web.

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

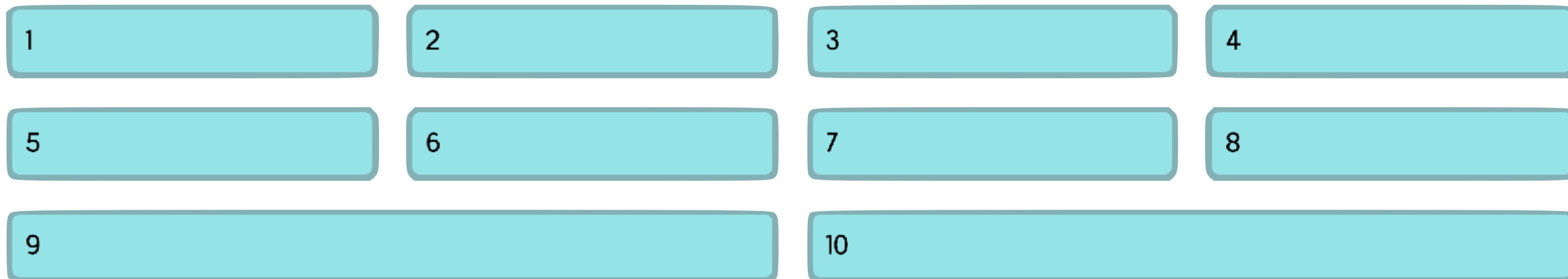
So, about this Grid thing ...

... why not
use Flexbox?

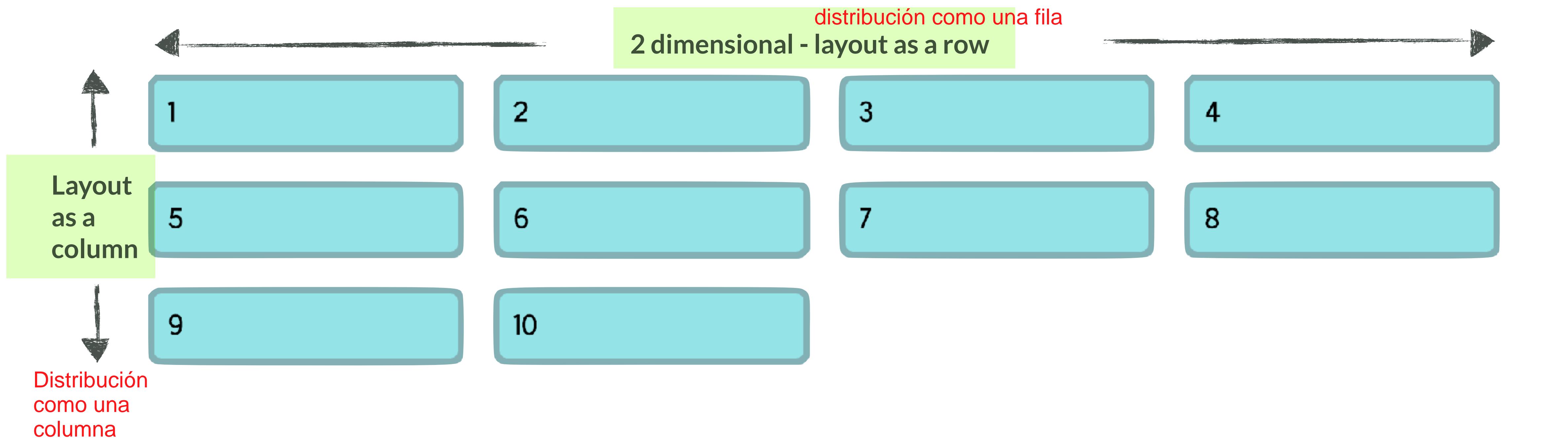
Do you need layout in
one dimension or two?

FLEXBOX

1 dimensional layout as a row



GRID LAYOUT



Grid trabaja desde dentro del contenedor

Grid works from the container in

cada otro metodo

de creación un GRID

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

envuelve dimensionando el elemento individual

un flotante basado en 'grid'

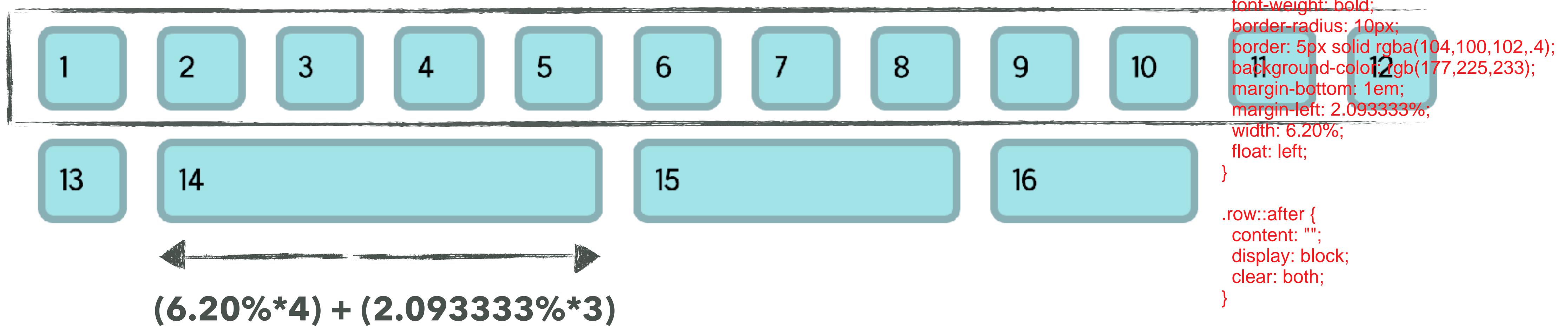
A float based “grid”

tenemos que darle un ancho al elemento

We have to give the items a width. By
Por apilando cuidadosamente estos artículos de tamaño
stacking these carefully sized items up
obtenemos la apariencia de grid
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



```
* { box-sizing: border-box; }
body {
  padding: 40px;
  background-color: #fff;
  font: 1.2em/1.4 'Pontano sans', sans-serif;
}

.wrapper {
  max-width: 980px;
  margin: 0 auto;
  padding-right: 2.093333%;
}

.col {
  padding: 10px;
  font-weight: bold;
  border-radius: 10px;
  border: 5px solid rgba(104,100,102,.4);
  background-color: #17c2e2;
  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%);
}

.col.span3 {
  width: calc((6.20%*3) + (2.093333%*2));
}

.col.span4 {
  width: calc((6.20%*4) + (2.093333%*3));
}
```

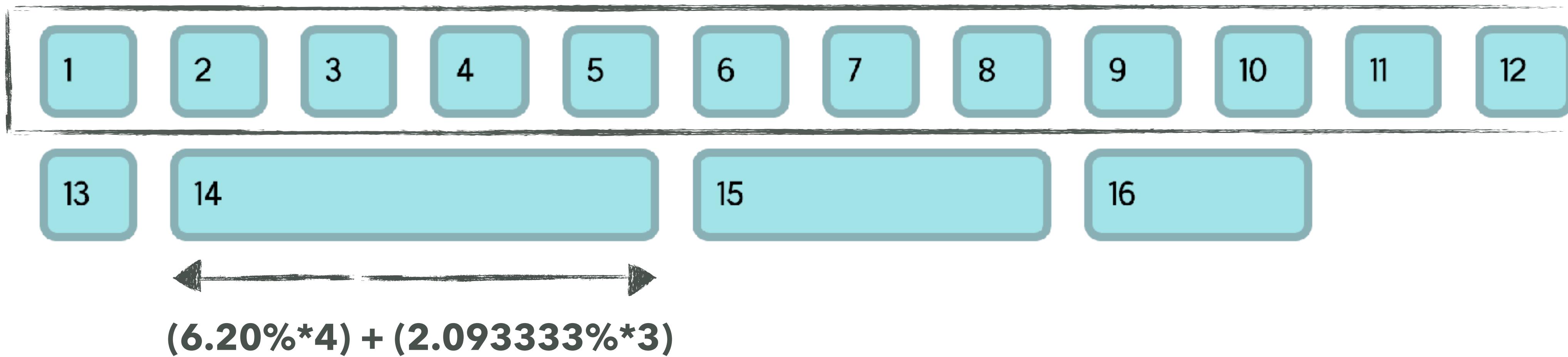
There is no grid. We made it look like
there is a grid by the fact things line up.

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 filas envueltas

sin dimensionado informacion
sobre los items (grid-element)

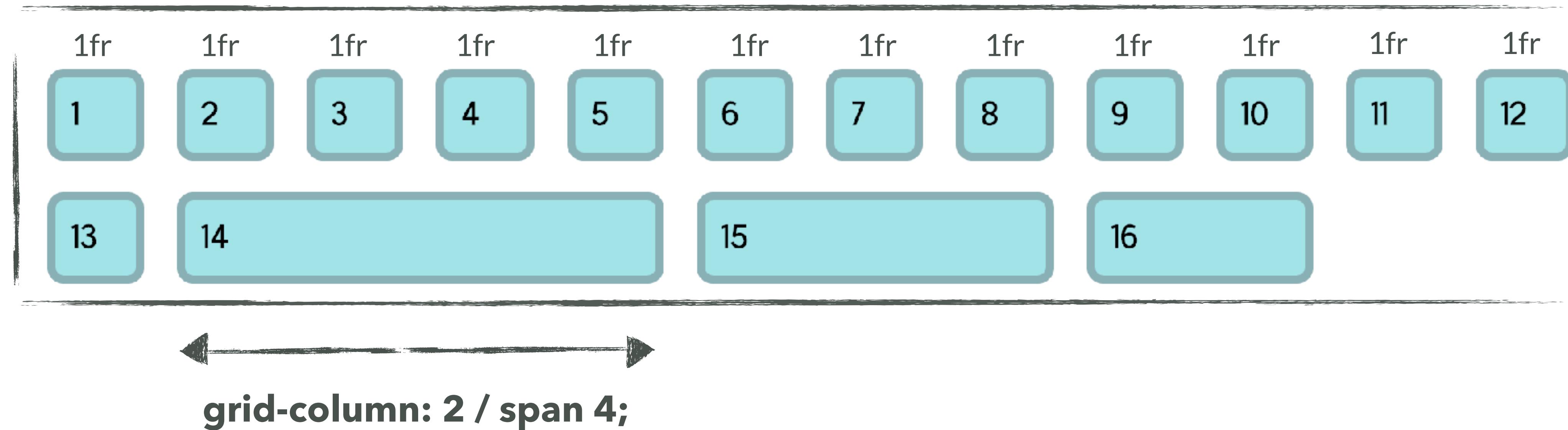
No row wrappers. No sizing information

on the items, just an instruction on how
many columns to span.
solo una instrucción sobre como cuantas columnas
a abarcar

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

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

Grid container



CREA ALGUNA COSA QUE

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

QUE SE VE COMO UN GRID CONTROLANDO TAMAÑO ITEM

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

colocas

elementos

dentro de el

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

Precision & Flexibility

Dimensionando Las PISTAS GRID

Sizing Grid Tracks

Grid container width



← 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));  
}
```

LOS GRID-ITEMS
se ajustan a la cantidad
de elementos que hay
no al ancho del contenedor

minmax()

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

minimo

maximo

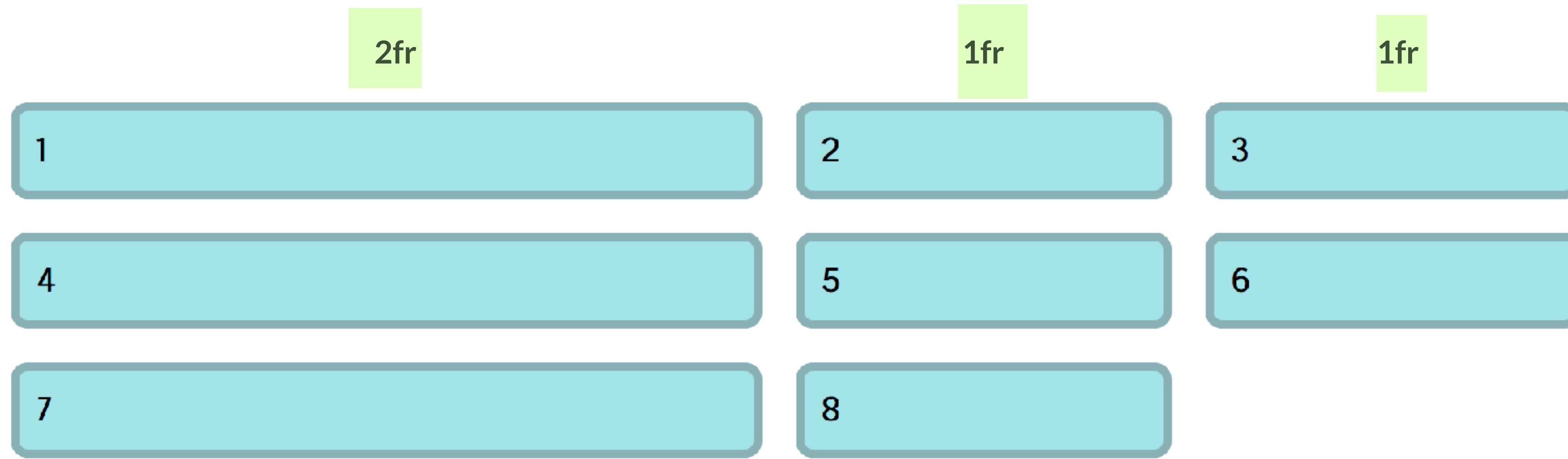
Fracción

The fr unit - distributing available space

The fr unit

pista , via , ruta ,
With this track listing the available
spaces divided into 4.

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



The fr unit

Mix absolute length units and **fr** units.

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

1fr

1

1fr

2

400px

3

4

5

6

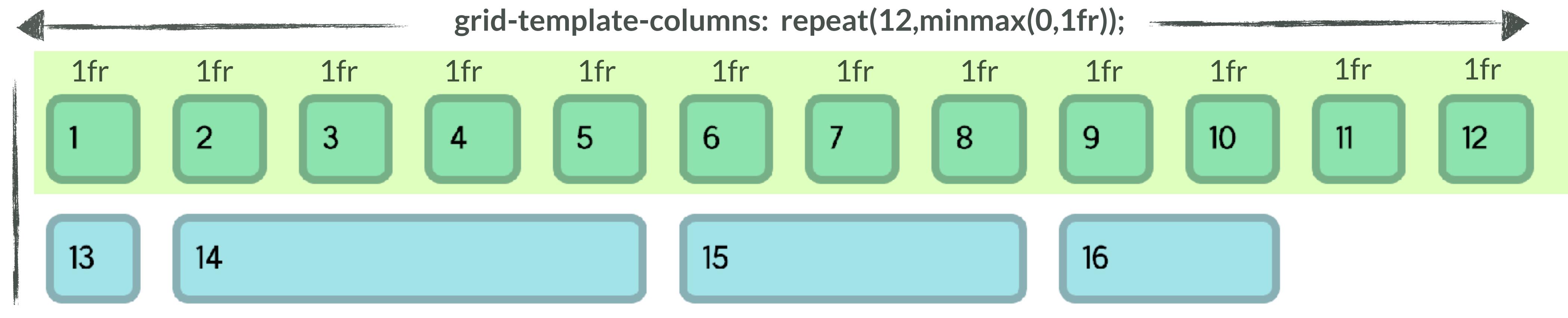
7

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;  
}
```



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

```
.col.span3 {  
    grid-column: auto / span 3;  
}
```

```
.col.span4 {  
    grid-column: auto / span 4;  
}
```

REEMPLAZA PORCENTAJES

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

200px



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.



grid-auto-rows: minmax(200px, auto);
minmax(200px, auto)



grid-auto-rows: minmax(200px, auto);
minmax(200px, auto)



grid-auto-rows: minmax(200px, auto);
minmax(200px, auto)



grid-auto-rows: 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 **se convierte en** a **grid container**, so we can make use of the ability to layer items on the grid.

```
figure {  
  → display: grid; CONVIERTE EN GRID-CONTAINER el GRID-ITEM  
  grid-template-rows: 1fr minmax(100px,  
  auto);  
}
```

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

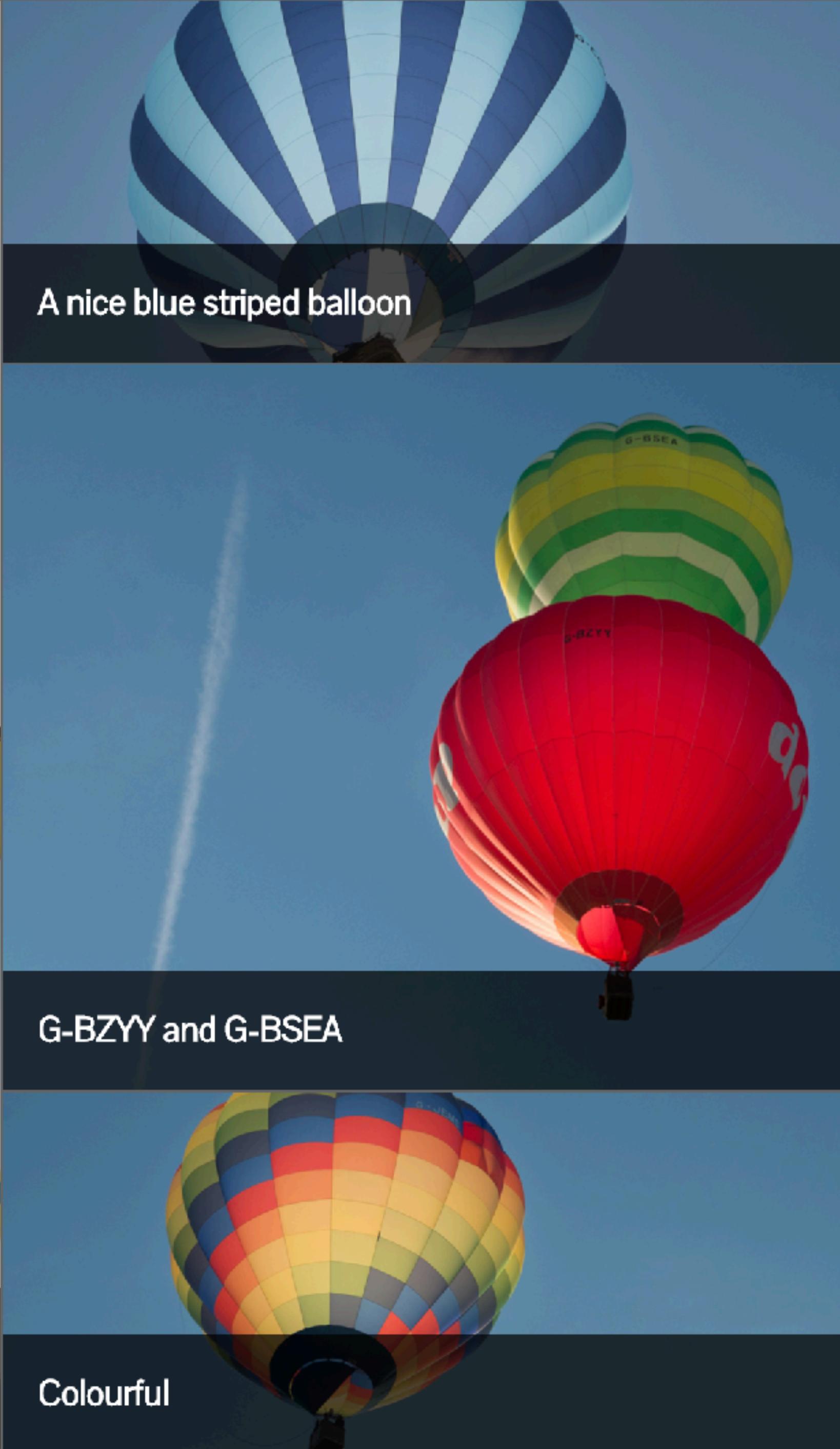
```
figure figcaption {  
  grid-row: 2;  
  grid-column: 1;  
  padding: 20px;  
}
```

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



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

aproximadamente la linea tamaño que podria formar

Roughly, the inline size that would fit
alrededor ellos el contenido si todo flexible
around its contents if all soft wrap
oportunidades oportunidades within the box were
taken.

Aproximadamente, el tamaño en línea
que encajaría alrededor
de su contenido si se tomaran
todas las oportunidades de envoltura
suave dentro de la caja

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

min-content

I am
sized
with
min-
content

1fr

2

1fr

3

4

5

6

7

8

max-content

más estrecho

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.

Por lo general, el tamaño en línea más estrecho que podría tomar mientras se ajustaba alrededor de su contenido si ninguna de las oportunidades de envoltura suave dentro de la caja se tomaron

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

max-content

I am sized with max-content

4

7

1fr

2

5

8

1fr

3

6

fit-content

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

Si el espacio disponible en un eje dado es finito, igual a $\min(\text{tamaño de contenido máximo}, \max(\text{tamaño de contenido mínimo}, \text{tamaño de ajuste}))$. De lo contrario, igual al tamaño de contenido máximo en ese eje

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

fit-content(200px)

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

2

3

1fr

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

un elemento inline-block que convierte en grid item

An inline-block item that becomes a grid
pierde algunos atributos que debería aplicar cuando era inline-block
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.

La creación de elementos anónimos no se produce si un elemento con display: table-cell u otra propiedad table-* se convierte en un elemento de cuadrícula

```
.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

Puede ser usado como una retirada para el

Can be used as a fallback for the Box

alineamiento propiedades box dentro inline-block o table

Alignment properties in inline-block or

table layout and stops applying when

y parar aplicando cuando el

elemento se convierte un

the item becomes a grid item.

Se puede utilizar como reserva para las propiedades
Box

Alignment en el diseño de bloque en línea o

tabla y deja de aplicarse cuando

el elemento se convierte en un elemento de cuadrícula

```
.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.

retirada

```
.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.

retirada

```
.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.

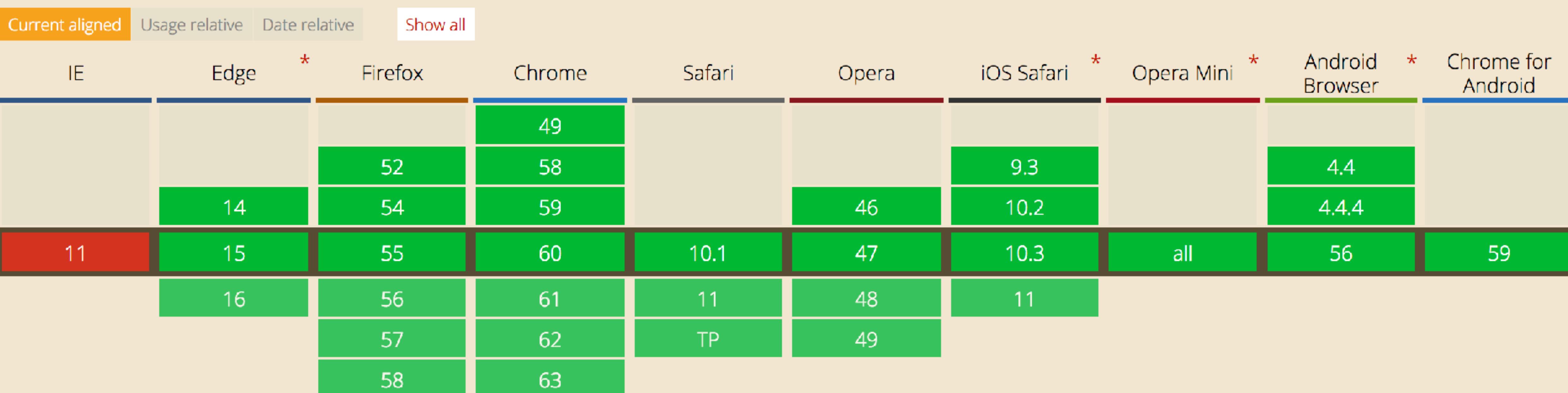
A problem!

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.



Notes

Known issues (2)

Resources (7)

Feedback

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

Feature Queries

Test for support of a property and value.

Dentro de

Inside the Feature Query add code only
for browsers that claim support.
reclamar

<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

distribucion
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;  
}
```

Grid Container

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

Permitir el interior crecer

empujando el pie hacia abajo

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;  
}
```

<p>1</p> <p>My title</p>  <p>The contents.</p> <p>Footer contents</p>	<p>2</p> <p>My title</p>  <p>The contents. I have a lot of content, more content than the other ones.</p> <p>Footer contents</p>	<p>3</p> <p>My title</p>  <p>The contents.</p> <p>Footer contents, I have a mahoosive footer. Just look at this stuff.</p>
<p>4</p> <p>My title</p>  <p>The contents.</p> <p>Footer contents</p>	<p>My title</p>  <p>The contents.</p> <p>Footer contents</p>	<p>My title this one is very long indeed</p>  <p>The contents.</p> <p>Footer contents</p>

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.
usa la pista de los grid padre 'grid-container'

```
.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

permitiendo a los elementos auto-colocar con

Allowing items to auto-place with
certos artículos espandiendo las filas y columnas
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);  
}  
  
/* Definimos la longitud del elemento hijo - Grid-Items */  
.grid > div:nth-child(1) {  
    grid-row-end: span 2;  
} /* Alargamos el contenido 2 espacios verticalmente */  
  
.grid > div:nth-child(2) {  
    grid-row-end: span 3;  
}  
  
.grid > div:nth-child(4) {  
    grid-row-end: span 3;  
}  
/* Alargamos el contenido 2 espacios horizontalmente */  
.grid > div:nth-child(5) {  
    grid-column-end: span 2;  
}
```

1

2

3

4

Espacio
VACIO

5

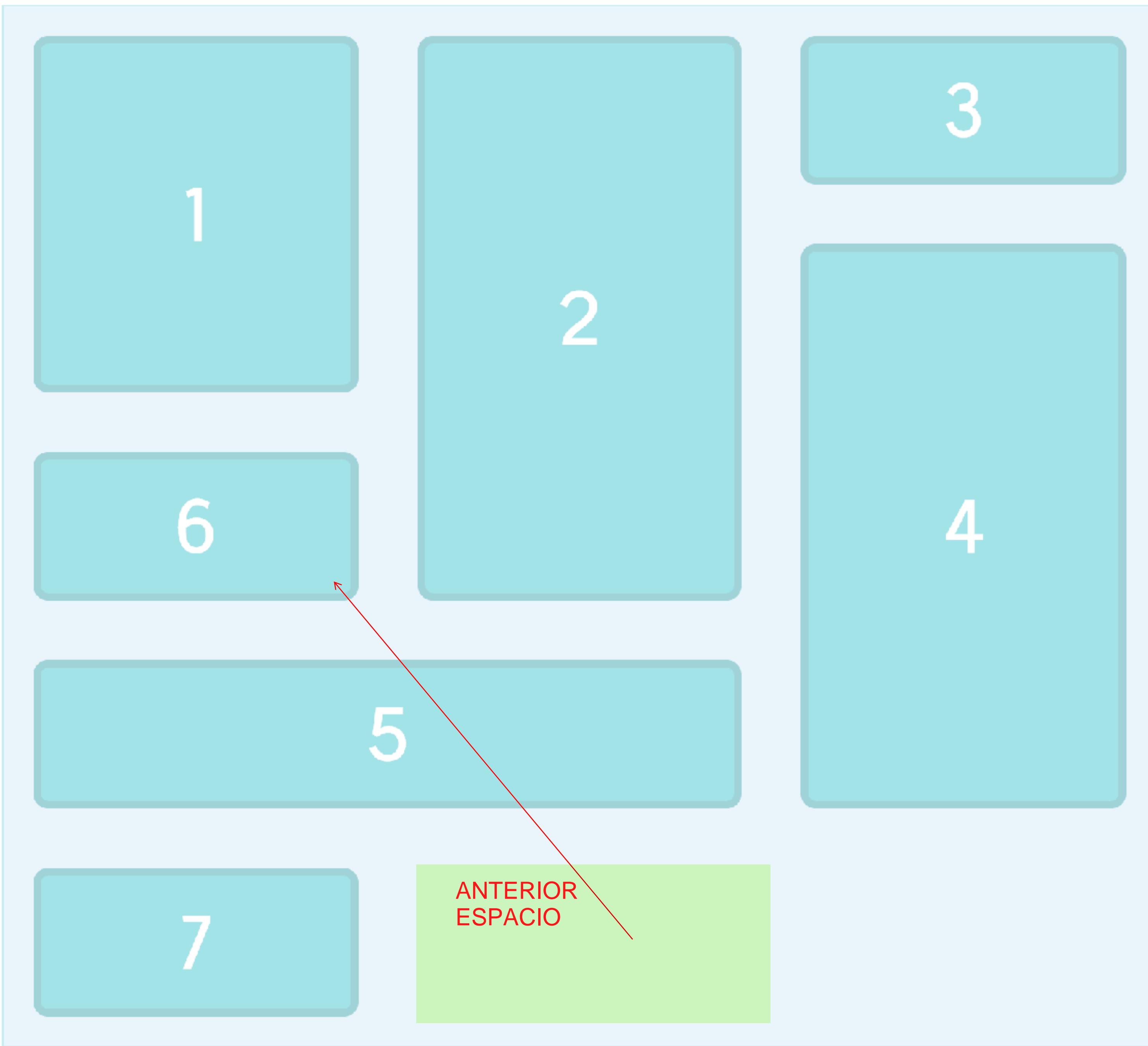
6

7

Set auto-flow to dense

relleno vacío/brecha fuera del
orden visual del documento
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;  
}
```



1	2	3
6	2	4
7	5	

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>

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

Out now!

The New CSS Layout

@rachelandrew

Slides & Resources: <https://rachelandrew.co.uk/speaking/event/view-source-2017>

Thank you!