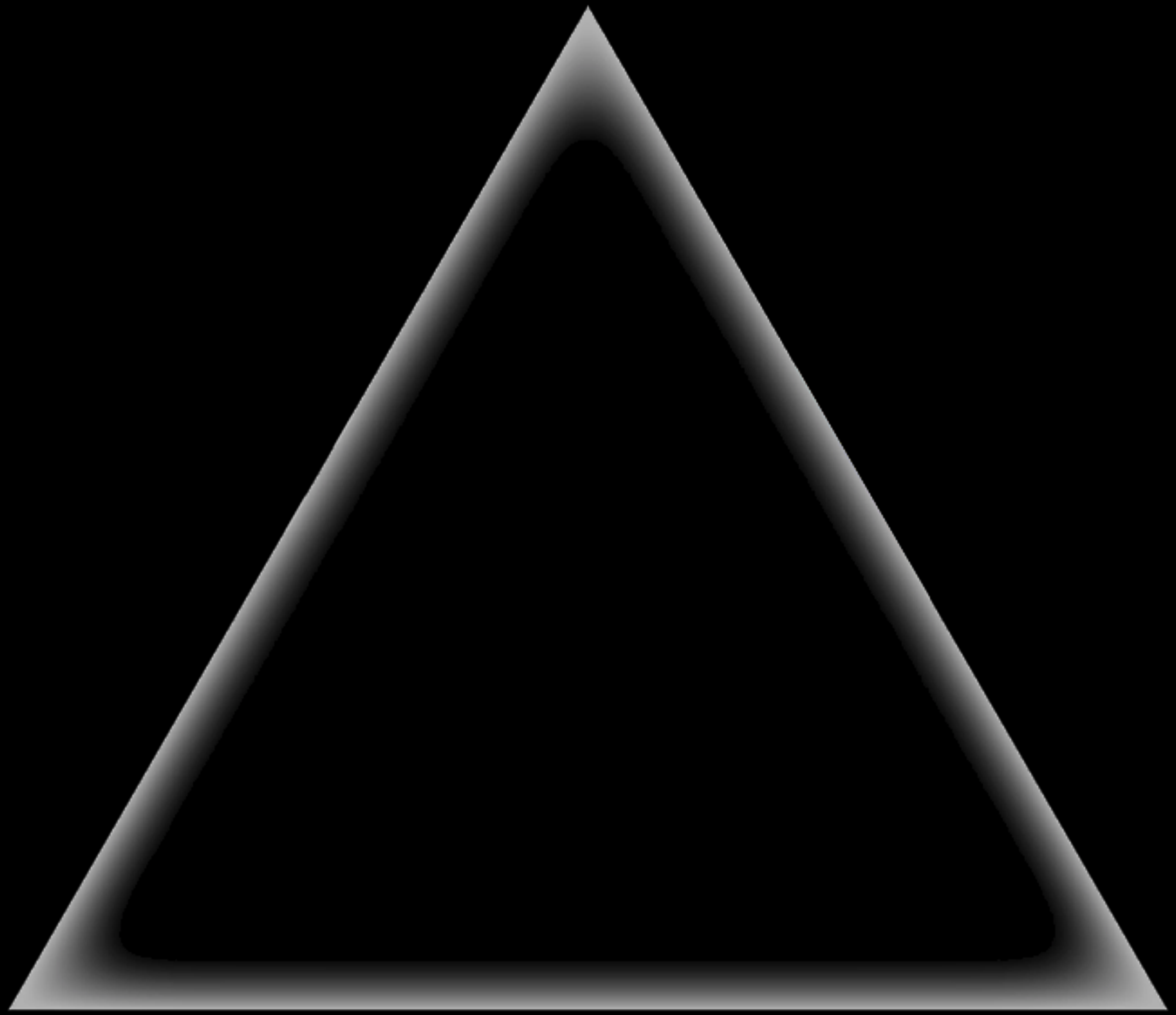


Manuel Matuzović

wien.gv.at - HTML, CSS, a11y ❤️

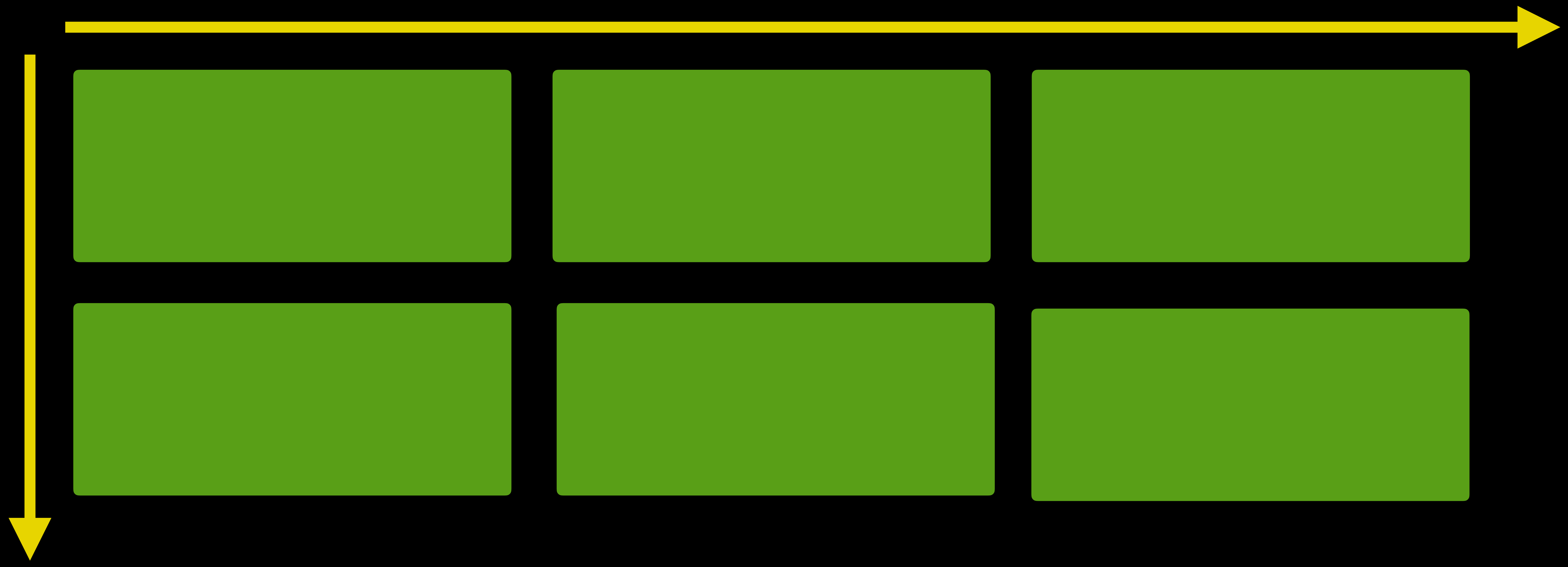
@mmatuzo

bit.ly/dark-grid-linz

The Dark Side of the Grid

CSS Grid Layout?

CSS Grid Layout is a grid-based layout system designed for two-dimensional layouts.



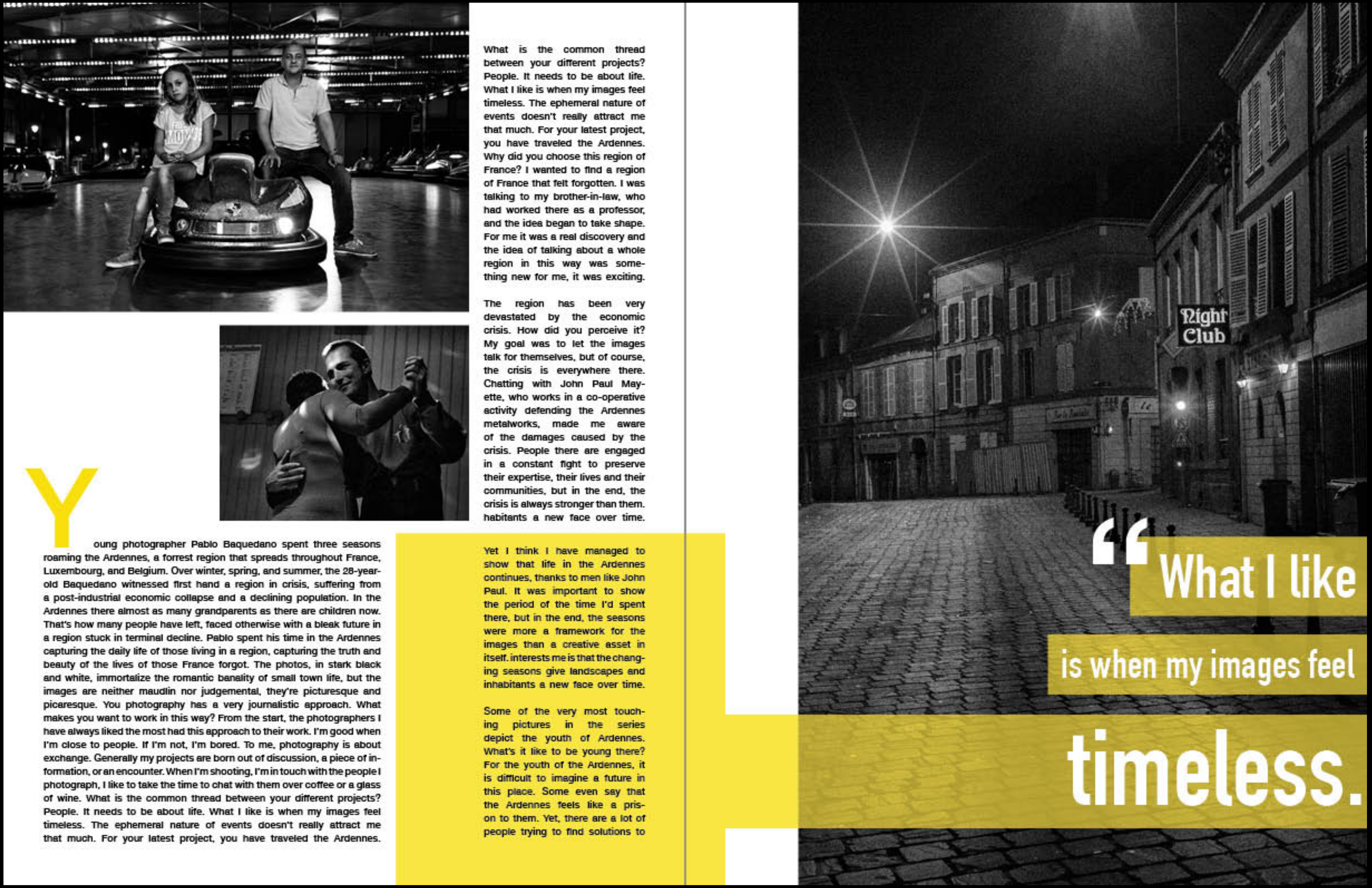
@mmatuzo

What’s special about Grid?

The first true layout method in CSS.

`float`, `display:inline-block`, `position`, `display:table, display: flex` not designed for building two-dimensional layouts.

Kinda like table layouts but responsive and flexible and described in CSS and not in HTML.



Young photographer Pablo Baquedano spent three seasons roaming the Ardennes, a forrest region that spreads throughout France, Luxembourg, and Belgium. Over winter, spring, and summer, the 28-year-old Baquedano witnessed first hand a region in crisis, suffering from a post-industrial economic collapse and a declining population. In the Ardennes there almost as many grandparents as there are children now. That's how many people have left, faced otherwise with a bleak future in a region stuck in terminal decline. Pablo spent his time in the Ardennes capturing the daily life of those living in a region, capturing the truth and beauty of the lives of those France forgot. The photos, in stark black and white, immortalize the romantic banality of small town life, but the images are neither maudlin nor judgemental, they're picturesque and picaresque. You photography has a very journalistic approach. What makes you want to work in this way? From the start, the photographers I have always liked the most had this approach to their work. I'm good when I'm close to people. If I'm not, I'm bored. To me, photography is about exchange. Generally my projects are born out of discussion, a piece of information, or an encounter. When I'm shooting, I'm in touch with the people I photograph, I like to take the time to chat with them over coffee or a glass of wine. What is the common thread between your different projects? People. It needs to be about life. What I like is when my images feel timeless. The ephemeral nature of events doesn't really attract me that much. For your latest project, you have traveled the Ardennes.

What is the common thread between your different projects? People. It needs to be about life. What I like is when my images feel timeless. The ephemeral nature of events doesn't really attract me that much. For your latest project, you have traveled the Ardennes. Why did you choose this region of France? I wanted to find a region of France that felt forgotten. I was talking to my brother-in-law, who had worked there as a professor, and the idea began to take shape. For me it was a real discovery and the idea of talking about a whole region in this way was something new for me, it was exciting.

The region has been very devastated by the economic crisis. How did you perceive it? My goal was to let the images talk for themselves, but of course, the crisis is everywhere there. Chatting with John Paul Mayette, who works in a co-operative activity defending the Ardennes metalworks, made me aware of the damages caused by the crisis. People there are engaged in a constant fight to preserve their expertise, their lives and their communities, but in the end, the crisis is always stronger than them. habitants a new face over time.

Yet I think I have managed to show that life in the Ardennes continues, thanks to men like John Paul. It was important to show the period of the time I'd spent there, but in the end, the seasons were more a framework for the images than a creative asset in itself. interests me is that the changing seasons give landscapes and inhabitants a new face over time.

Some of the very most touching pictures in the series depict the youth of Ardennes. What's it like to be young there? For the youth of the Ardennes, it is difficult to imagine a future in this place. Some even say that the Ardennes feels like a prison to them. Yet, there are a lot of people trying to find solutions to

“What I like is when my images feel timeless.

Article layout for a fictional magazine. Source: http://juliabialko.com/magarticle/

A Brief overview of Grids History

Early 90s: First ideas, but “too complex” to implement.

1996: “frame-based” layout model.

2005: Advanced Layout Module (later Template Layout Module).

Start
mat
honan
Interne
Mail
2
30
Wednesday
Internet Explorer
Store
San Francis
Area — Ne
Photos
Maps
SkyDrive
The New Yo
Messaging
:-)
Video
Music
Paris, France
Desktop
Weather
Xbox LIVE Games
Camera
Finance
TESLA

Joe Brown
Out of the office until Jun...
Hi there, I'm out of the o...

Sun Sentinel - Celtics' Allen mired in shooting slump

Grids in Microsofts Metro Design

A Brief overview of Grids History

2011: Microsoft ships a Grid implementation behind the `-ms-` vendor prefix in IE 10.

Microsoft presents a draft spec to the W3C in 2012.

CSS Working Group begins tweaking Microsoft’s proposal.

Bloomberg hires Igalia to implement CSS Grid for both Blink and WebKit.

A Brief overview of Grids History

January 2017: CSS Grid in Chromium 56 for Android.

Early March 2017: Chrome and Firefox.

End of March 2017: Opera and Safari.

October 17th, 2017: Edge.

Grid-related properties and values

display

grid-template-columns

grid-template-rows

grid-template-areas

grid-template

grid-column-gap

grid-row-gap

grid-gap

justify-items

align-items

place-items

justify-content

align-content

place-content

grid-auto-columns

grid-auto-rows

grid-auto-flow

grid

grid-column-start

grid-column-end

grid-row-start

grid-row-end

grid-column

grid-row

grid-area

justify-self

align-self

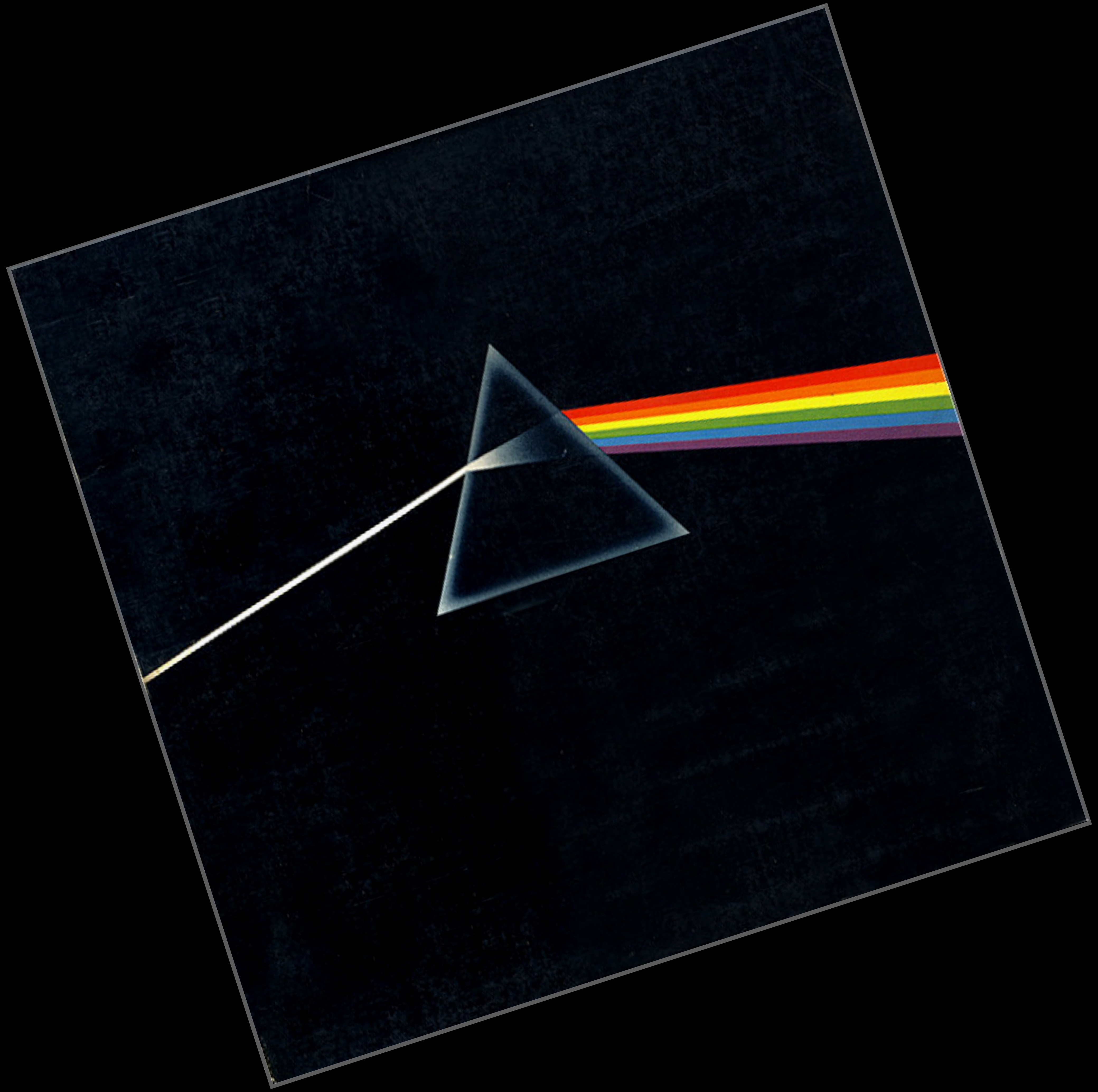
place-self

min-content, max-content, fit-content

fr

repeat()

minmax()



Cover of the LP “The Dark Side of the Moon” by Pink Floyd, released in 1973.

PINK FLOYD FUN FACT #1

45 million copies sold

The Dark Side of the Moon is, with over 45 million copies sold, the fourth best-selling album worldwide.

#1 Thriller by Michael Jackson (66 Million)
#2 Their Greatest Hits (1971-1975) by Eagles (51 Million)
#3 Back in Black by AC/DC (50 Million)

@mmatuzo

CHANGING VISUAL ORDER

Changing Visual Order

Explicit placement

`order` property

Absolute positioning

Auto flow

Areas

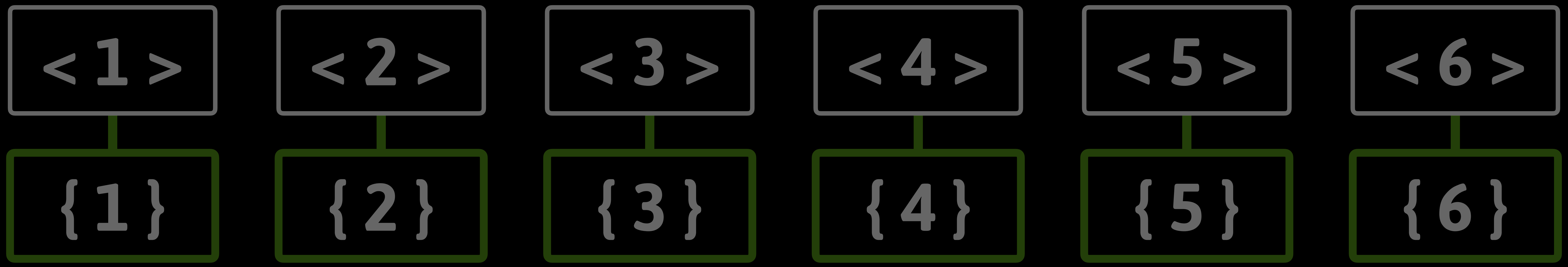
Visual Order

Both the tab order and the order in which screen readers read content follow DOM order.

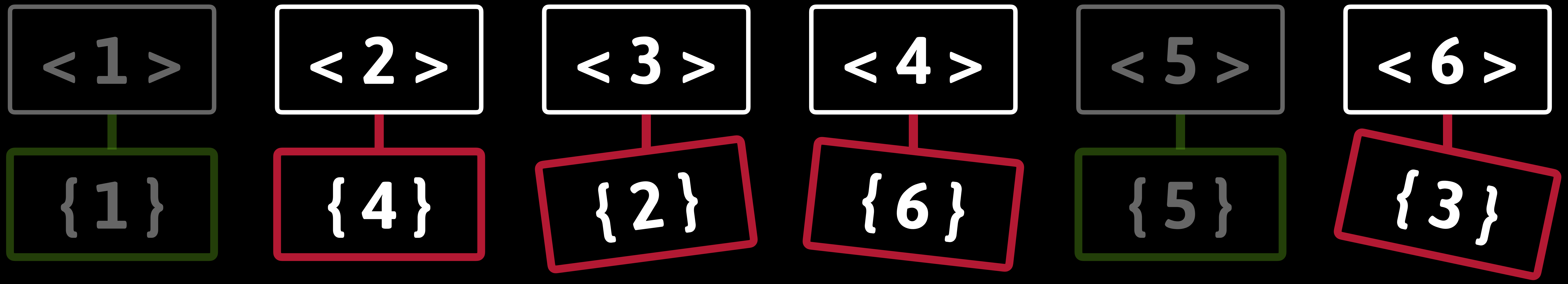
Changing visual order in CSS has no effect on tab and screen reader order.

Visual Order

Visual order matches DOM order



Visual order doesn’t match DOM order



If visual order doesn’t match DOM order.

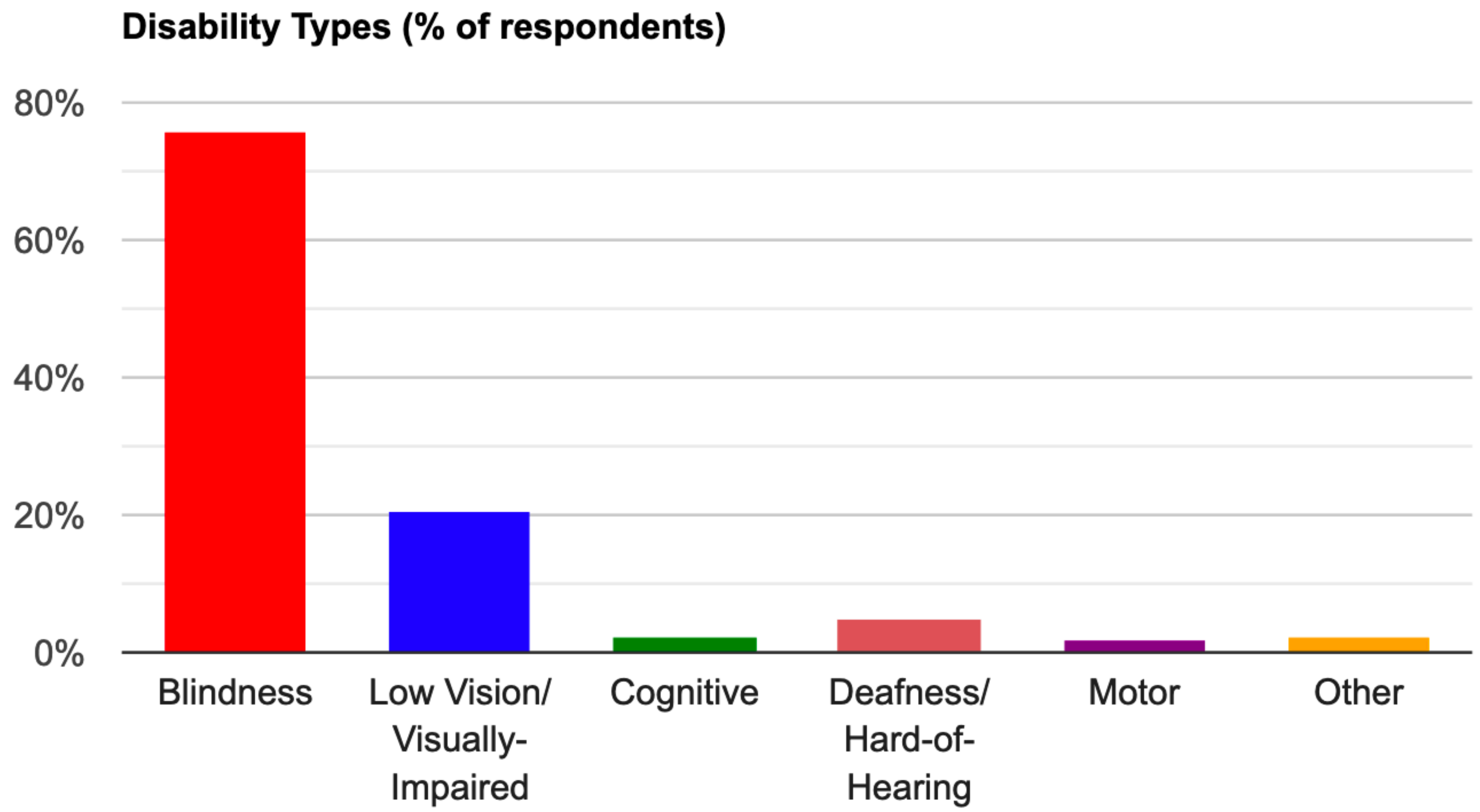
1. A keyboard user may have trouble predicting where focus will go next.

If visual order doesn’t match DOM order.

2. Users of screen magnifiers may be confused when the display detail skips around constantly.

If visual order doesn’t match DOM order.

3. If a blind user is working with a sighted user who reads the page in visual order, they may be confused when they encounter information in different order.



Which of the following disabilities do you have?

Response	# of Respondents	% of Respondents
Blindness	1,358	75.8%
Low Vision/Visually-Impaired	366	20.4%
Cognitive	39	2.2%
Deafness/Hard-of-Hearing	90	5.0%
Motor	33	1.8%
Other	41	2.3%

75.8% users are blind, 20.4% have low vision.

Who uses the keyboard for navigation?

People with physical disabilities who cannot use the mouse.

People with chronic conditions, such as repetitive stress injuries (RSI), who should limit or avoid use of a mouse.

People who temporarily can’t use the mouse due to an injury.

Powerusers (e.g. developers or designers 🤓).

Who uses screen readers for navigation?

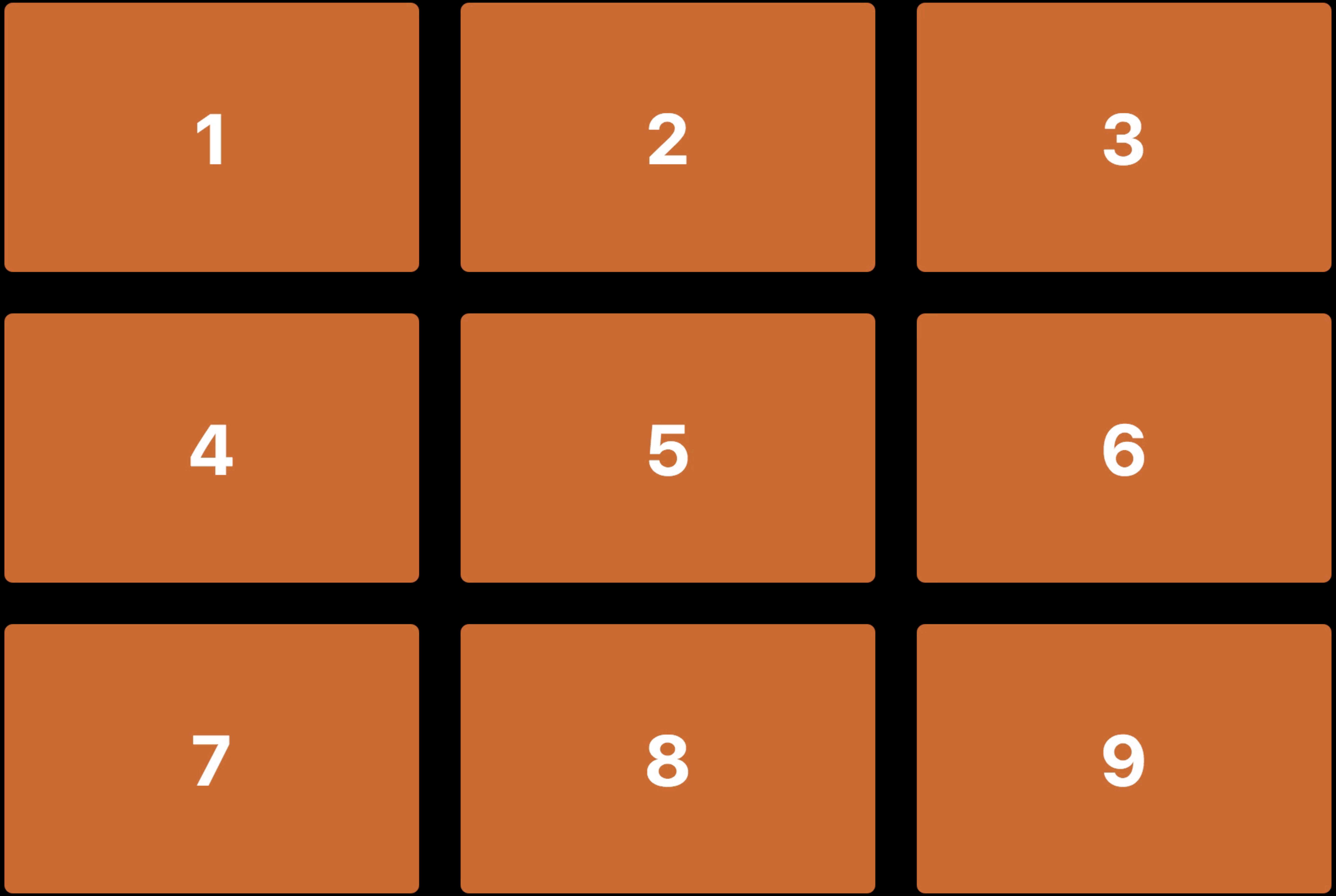
Blind people.

People with low vision to supplement what they see on the screen.

People with learning disabilities.

VISUAL ORDER

EXPLICIT PLACEMENT



1	2	3
4	5	6
7	8	9

https://codepen.io/matuzo/pen/xaOgGg

Explicit Placement

Grid gives us the ability to place items in every cell we want in our grids by specifying in which column and row they start or end.

`grid-column`

`grid-column-start`

`grid-column-end`

`grid-row`

`grid-row-start`

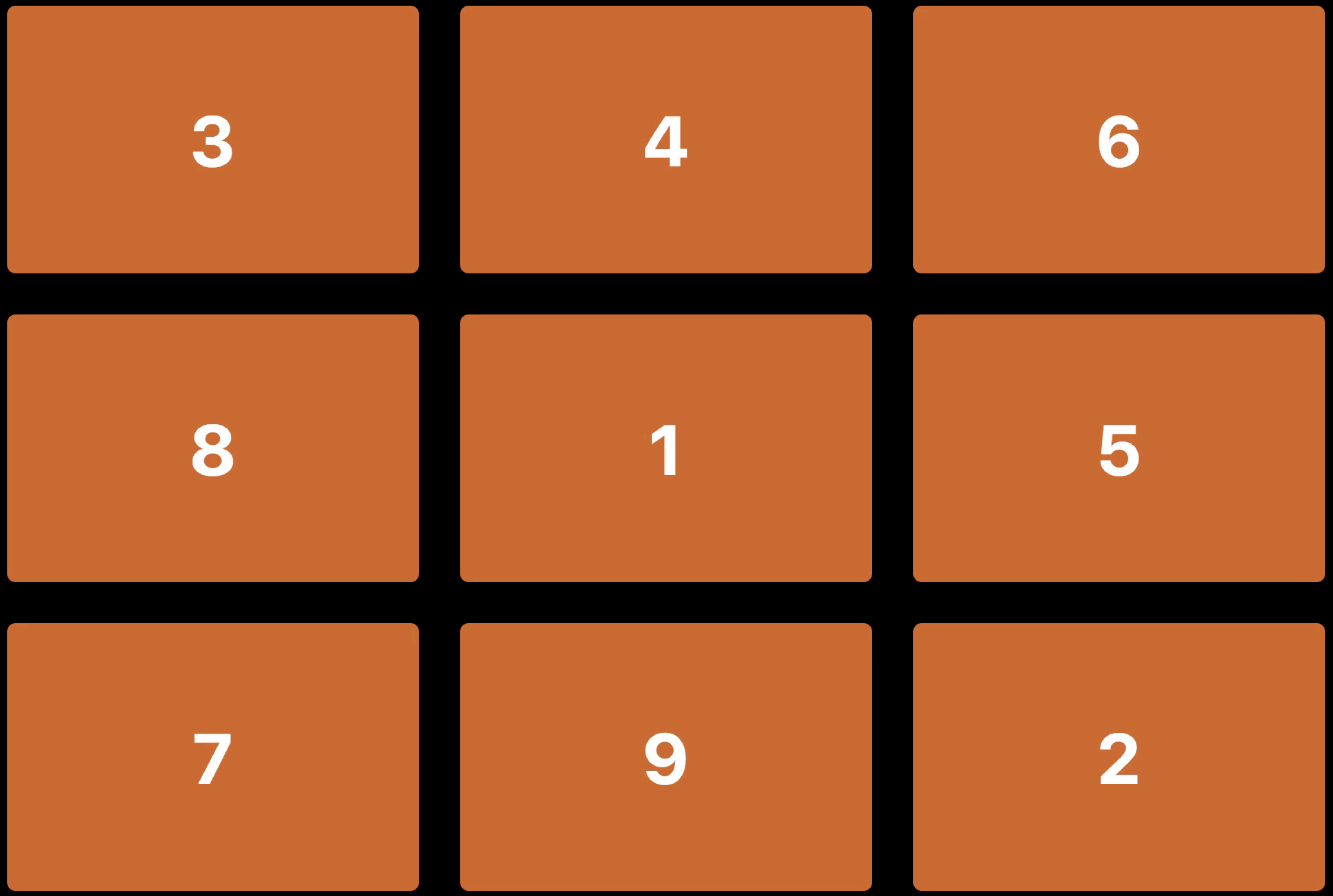
`grid-row-end`

$ explicit placement

```
.item:nth-child(1) {
  grid-column: 2;
  grid-row: 2;
}
```

```
.item:nth-child(2) {
  grid-column: 3;
  grid-row: 3;
}
```

```
 ...
```



3	4	6
8	1	5
7	9	2

https://codepen.io/matuzo/pen/xaOgGg

VISUAL ORDER

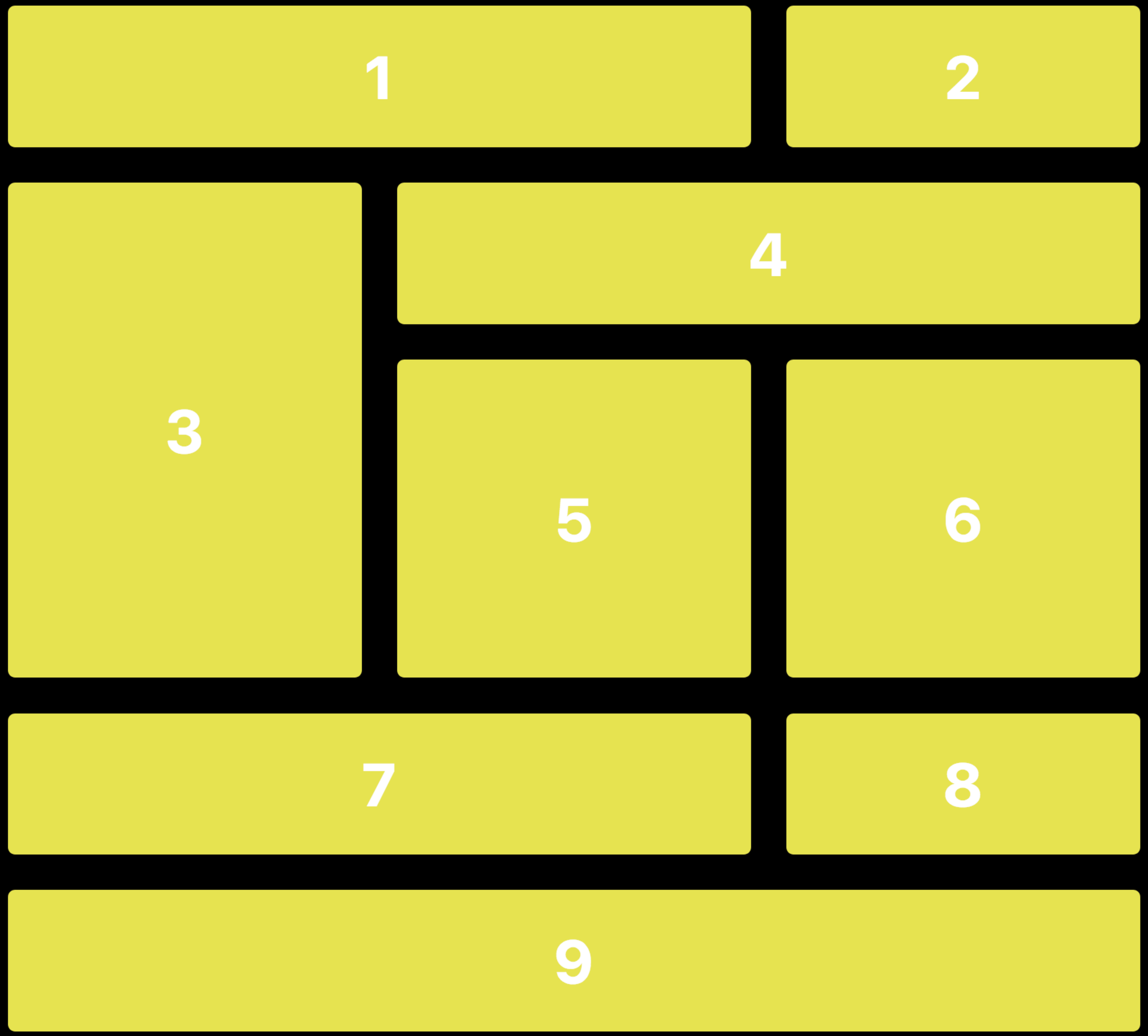
ORDER PROPERTY

$ order property

```
.item:nth-child(2) {
  order: 9;
}

.item:nth-child(5) {
  order: 6;
}

 …
```

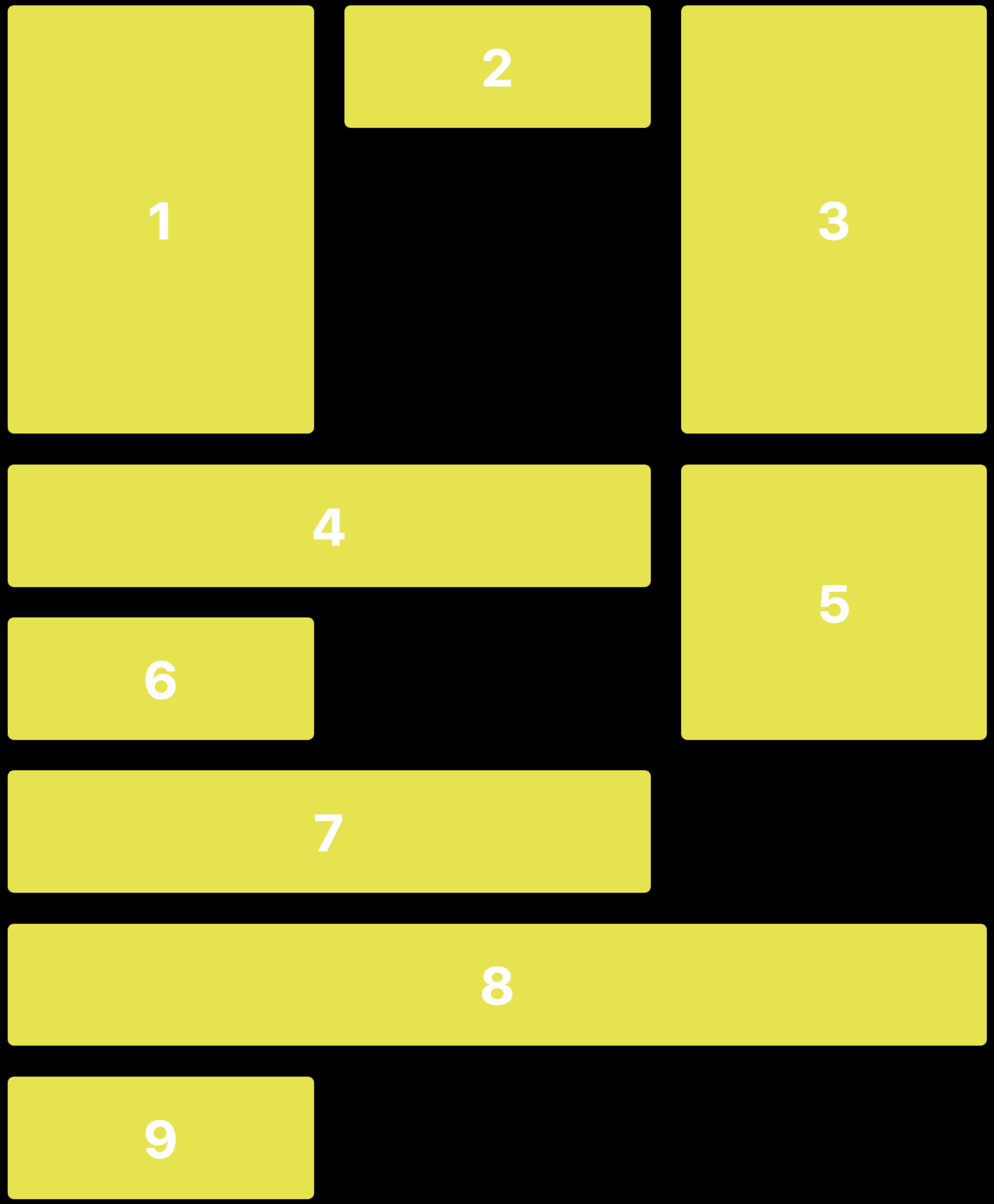
VISUAL ORDER

AUTO-FLOW



https://s.codepen.io/matuzo/pen/pONEzJ

@mmatuzo



https://s.codepen.io/matuzo/pen/pONEzJ

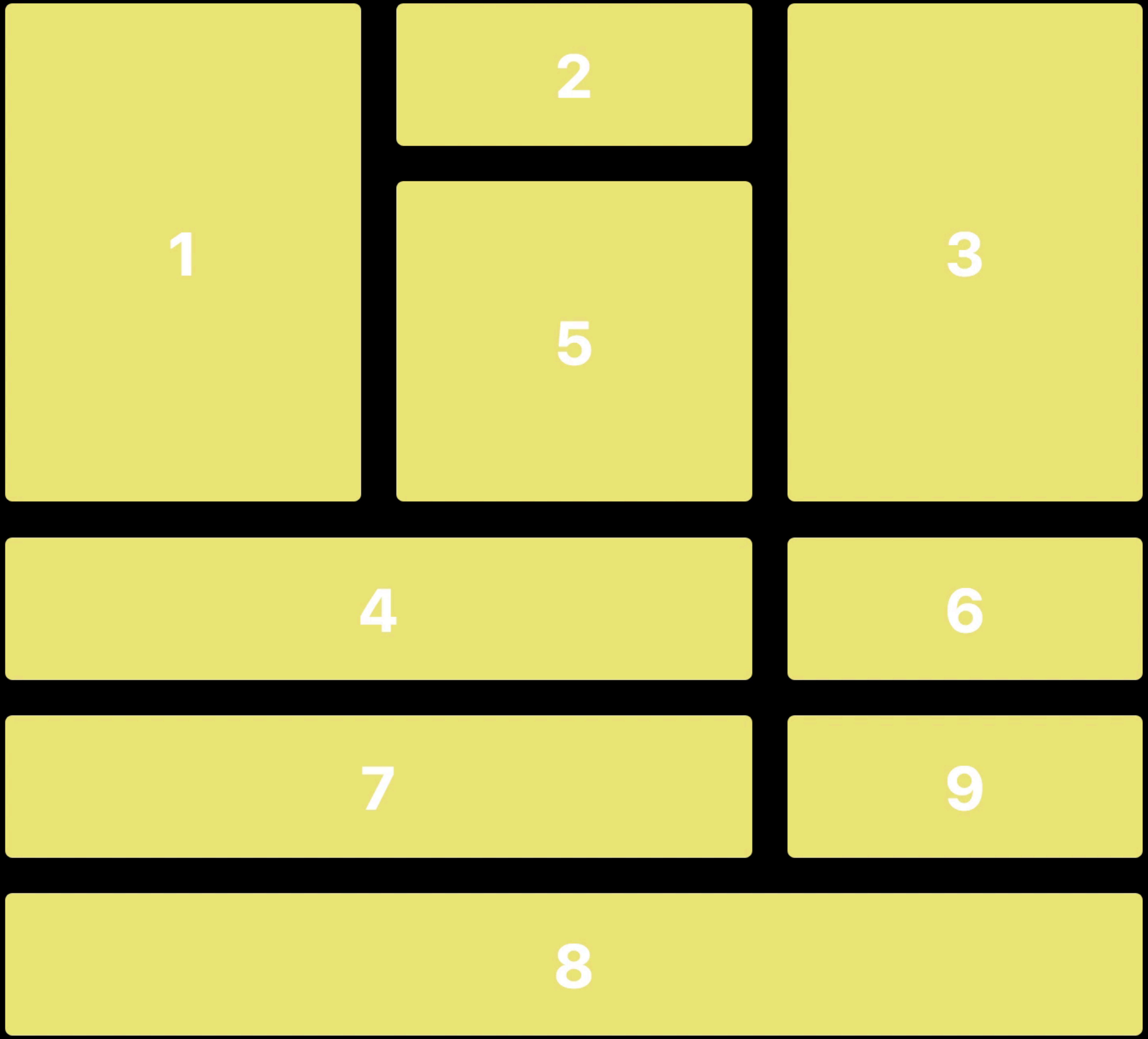
@mmatuzo

$ auto-flow

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-auto-rows: 80px;
  grid-gap: 20px;

}
```

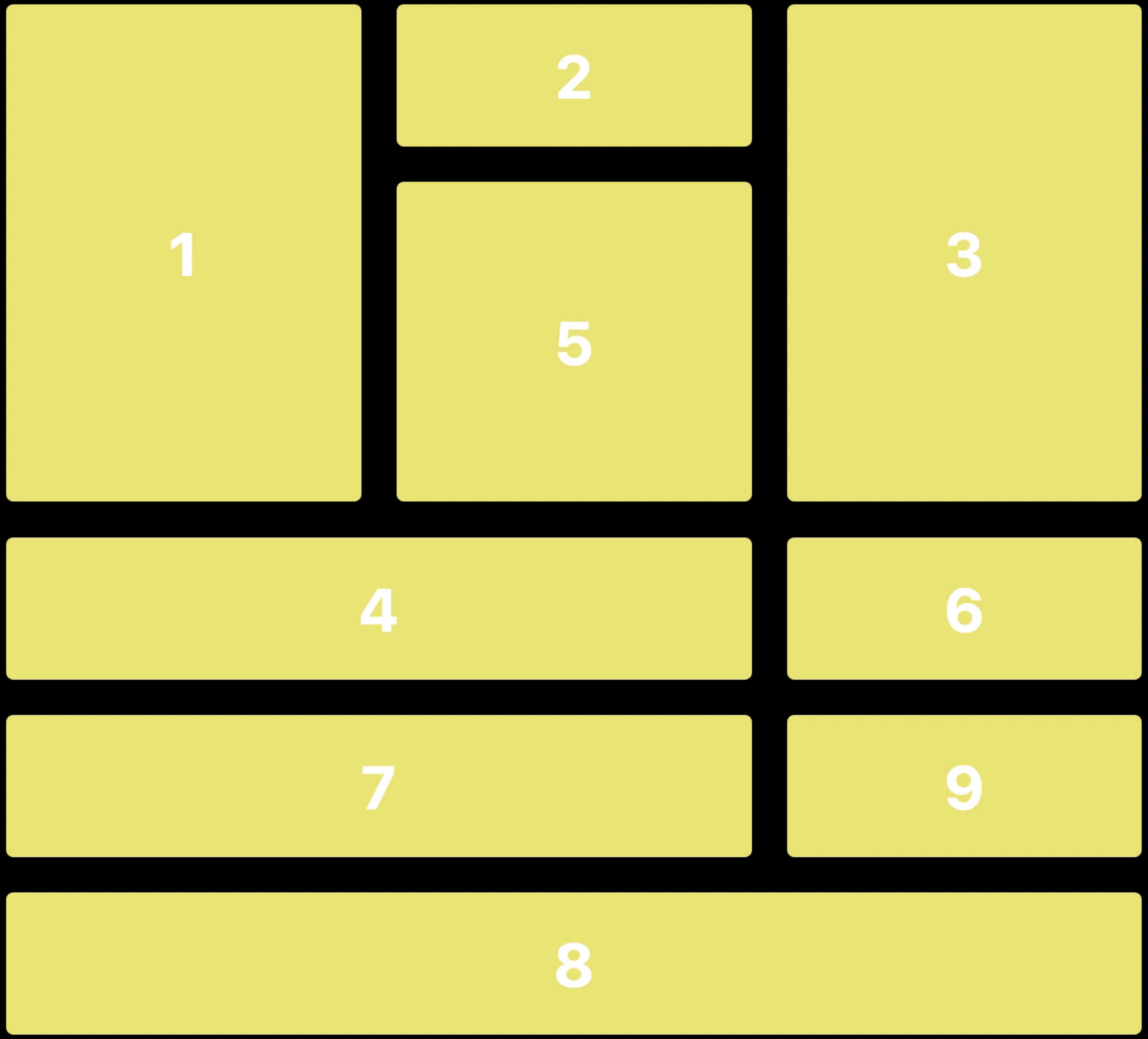
$ auto-flow

```css
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-auto-rows: 80px;
  grid-gap: 20px;
  grid-auto-flow: dense;
}
```



https://s.codepen.io/matuzo/pen/pONEzJ

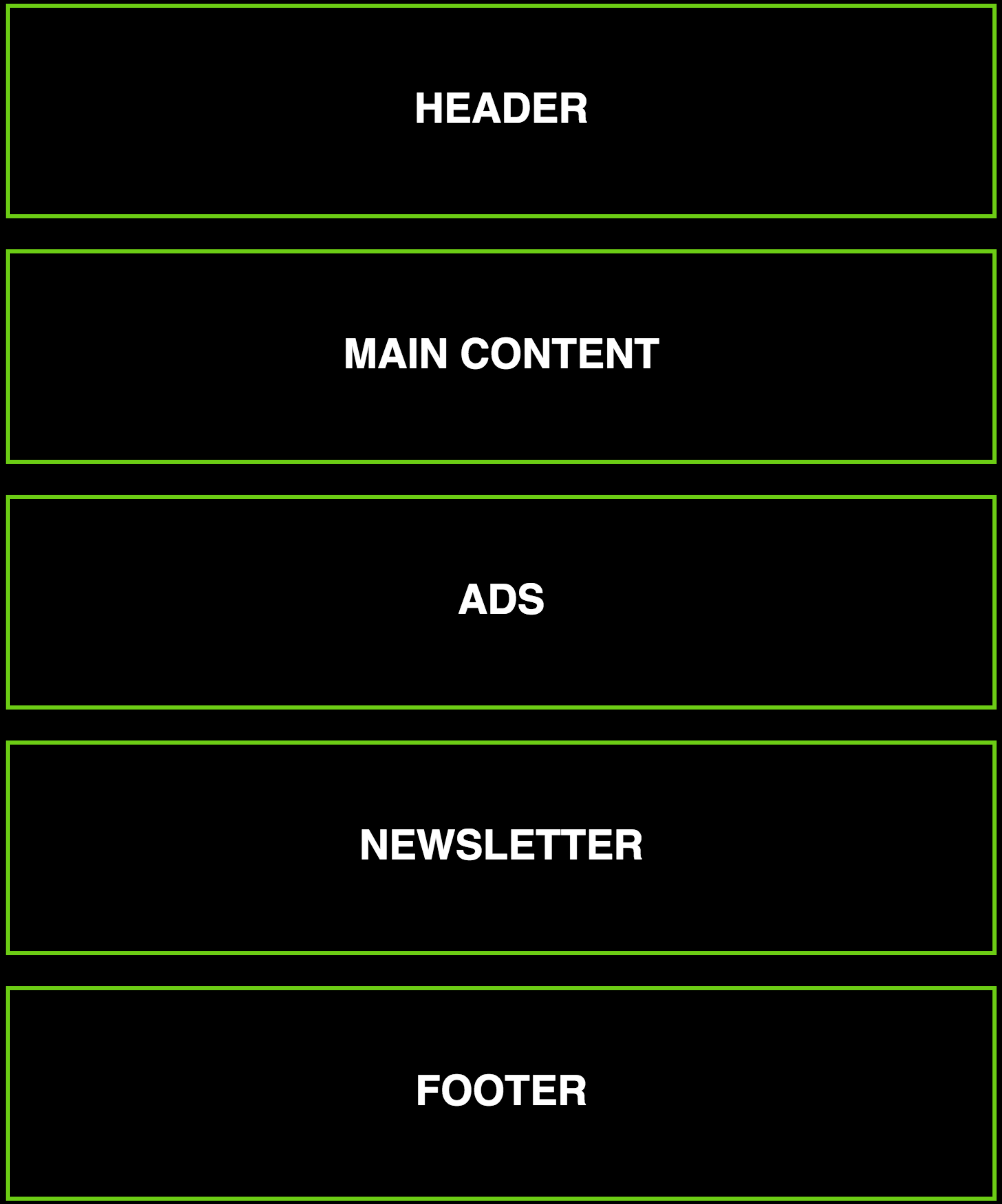
@mmatuzo



https://s.codepen.io/matuzo/pen/pONEzJ

VISUAL ORDER

AREAS



HEADER

MAIN CONTENT

ADS

NEWSLETTER

FOOTER

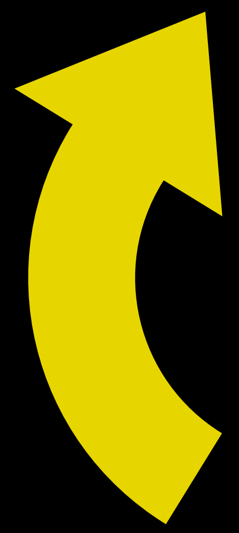
Example: Mobile layout

$ areas

```
body {
  display: grid;
  grid-template-areas:
    "header"
    "content"
    "ads"
    "newsletter"
    "footer";
}
```

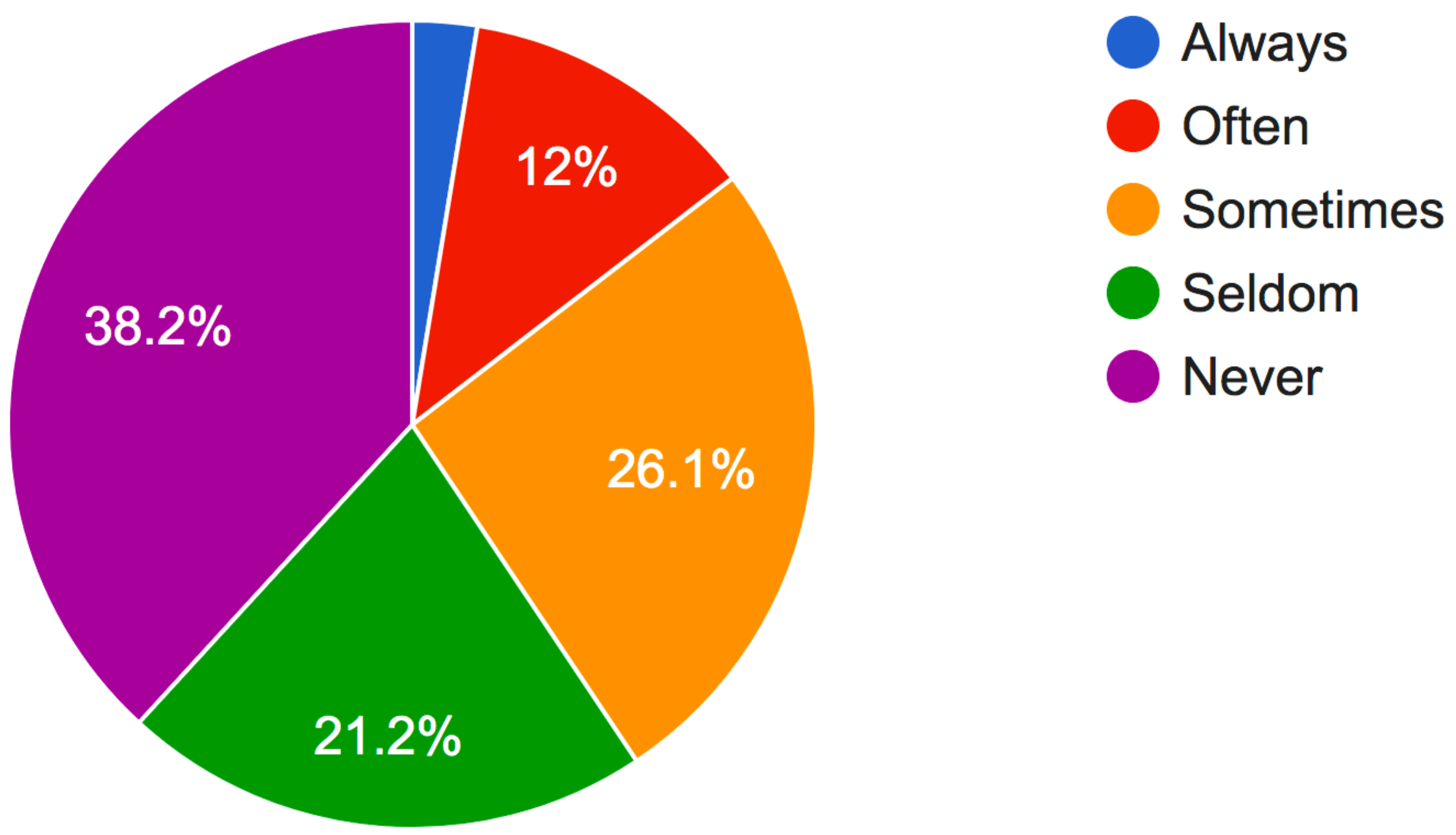
$ areas

```
body {
  display: grid;
  grid-template-areas:
    "header"
    "content"
    "ads"
    "newsletter"
    "footer";
}
```



$ areas

```
body {
  display: grid;
  grid-template-areas:
    "header"
    "newsletter"
    "content"
    "ads"
    "footer";
}
```



When using a mobile screen reader how often do you use an external keyboard?

Response	# of Respondents	% of Respondents
Always	39	3.9%
Often	181	11.8%
Sometimes	394	25.7%
Seldom	320	20.9%
Never	577	37.7%

15.7% use a keyboard always or often when using their mobile devices.

$ areas

```
body {
  grid-template-columns: 1fr 400px;
  grid-template-areas:
    "header header"
    "newsletter newsletter"
    "content ads"
    "content related"
    "comments comments"
    "footer footer";
}
```

WHOSE RESPONSIBILITY IS IT?

Our options

Dynamically changing source order for every breakpoint with JavaScript. 🙄

Applying `tabindex` or `aria-flow-to` to all elements and reorder them accordingly. 😭

Browser sniffing - Send different HTML sources to each client/devices type. 🤢

Our options

Mitigate the problem by providing skip links. 😐

Don’t change the visual order. 🙄

> „Authors must use order and the grid-placement properties only for visual, not logical, reordering of content.“

- https://drafts.csswg.org/css-grid/#order-accessibility

> Brilliant on the one hand for them trying to at least advise everybody in the right direction but really — come on. […] Suggesting that we don’t all use it just because of this, I think, is wishful thinking.

-Léonie Watson FF Conf 2016 https://www.youtube.com/watch?v=spxT2CmHoPk

So…what do we do?

Think about source order before writing HTML and CSS.
Work closely with designers early on.

Start with a well structured HTML document.

Write CSS mobile first and change the layout according to screen sizes.

So…what do we do?

Test what you've built by tabbing through the layout at different screen sizes.

Return to the source if there is an order mismatch across screen sizes.

PINK FLOYD FUN FACT #2

736 weeks on the Billboard chart

The Dark Side of The Moon spent 736 weeks on the American Billboard chart. That’s more than 14 years!

COMPROMISING ON SEMANTICS

@mmatuzo

> “I believe there will be a strong temptation, especially with Grid, to flatten out document structure in order that all elements become a child of the element with the Grid declared.
> Making layout simple, but at what cost?”

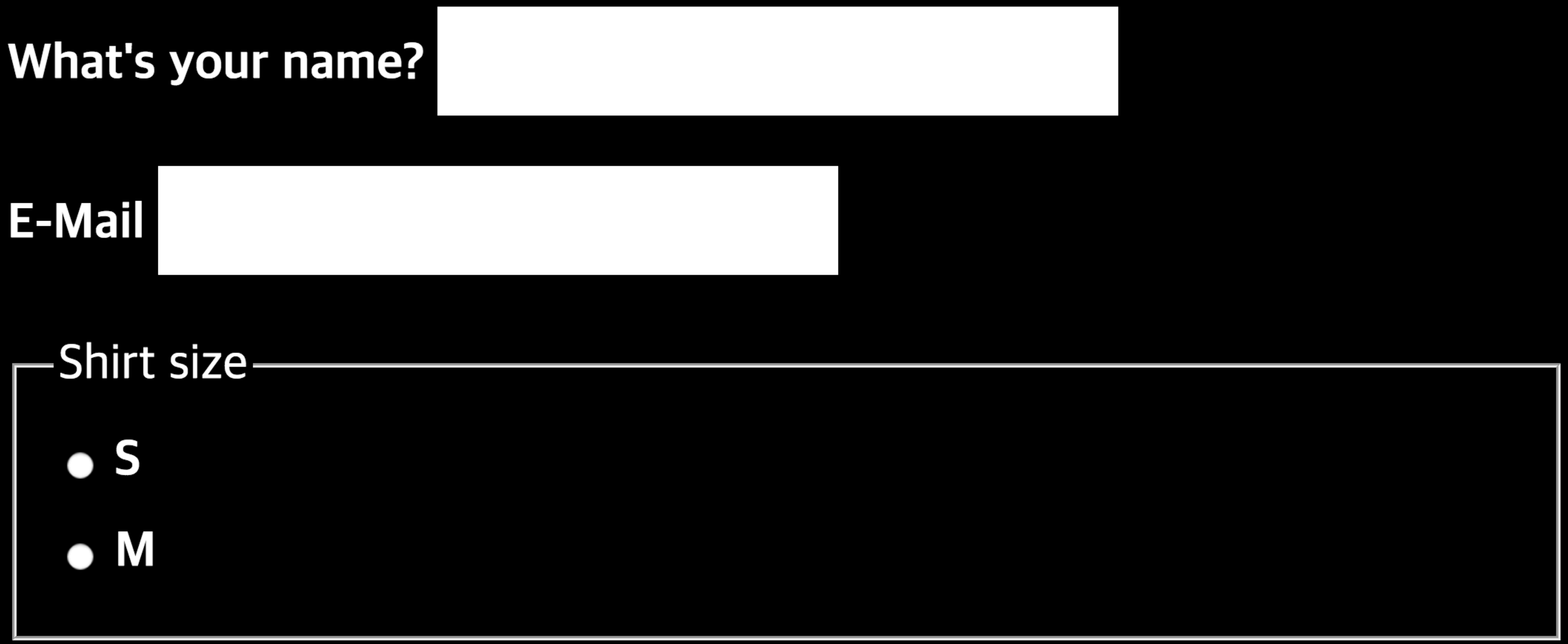
Rachel Andrew
(rachelandrew.co.uk/archives/2015/07/28/modern-css-layout-power-and-responsibility)

$ compromising on semantics

```
<form>
  <div>
    <label for="name">What's your name?</label>
    <input type="text" id="name" />
  </div>
  <div>
    <label for="email">E-Mail</label>
    <input type="email" id="email" />
  </div>
  …
```

$ compromising on semantics

```
  …
  <fieldset>
    <legend>Shirt size</legend>
    <div>
      <input type="radio" id="s" name="r">
      <label for="s">S</label>
    </div>
    <div>
      <input type="radio" id="m" name="r">
      <label for="m">M</label>
    </div>
  </fieldset>
</form>
```



What's your name?

E-Mail

Shirt size

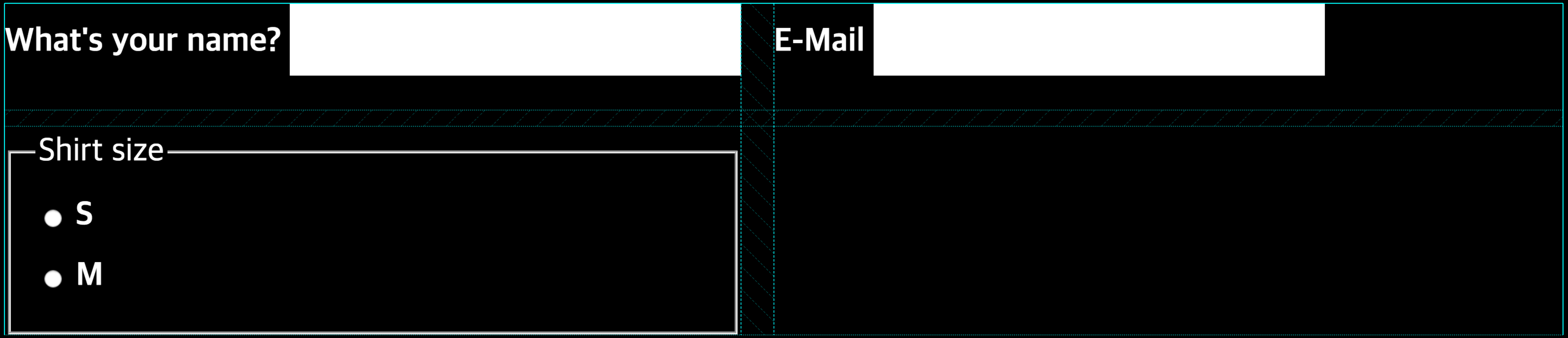
○ S

○ M

https://s.codepen.io/matuzo/pen/BOWYLB?editors=1100

$ compromising on semantics

```
form {
  display: grid;
  grid-template-columns: max-content
                         minmax(auto, 600px);
  grid-gap: 10px 20px;
}
```



What's your name?

E-Mail

Shirt size

○ S

○ M

https://s.codepen.io/matuzo/pen/BOWYLB?editors=1100

@mmatuzo

$ compromising on semantics

```html
<div>
  <label for="name">What's your name?</label>
  <input type="text" id="name" />
</div>
<div>
  <label for="email">E-Mail</label>
  <input type="email" id="email" />
</div>
```

$ compromising on semantics

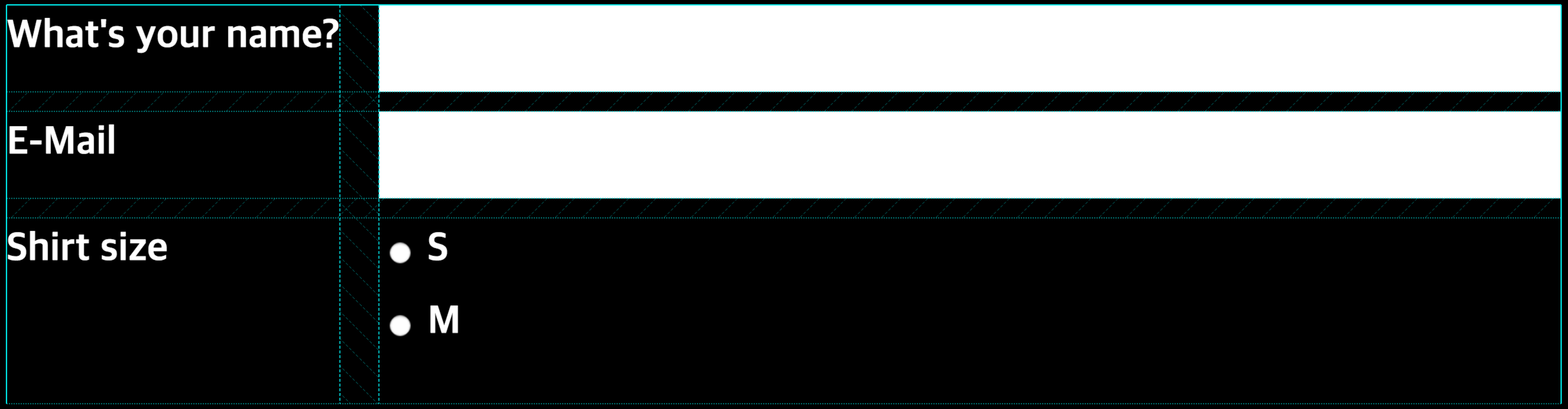
```
~~<div>~~
  <label for="name">What's your name?</label>
  <input type="text" id="name" />
~~</div>~~
~~<div>~~
  <label for="email">E-Mail</label>
  <input type="email" id="email" />
~~</div>~~
```

$ compromising on semantics

```
<fieldset>
  <legend       >Shirt size</legend       >

    <div>
      <input type="radio" id="s">
      <label for="s">S</label>
    </div>
    <div>
      <input type="radio" id="m">
      <label for="m">M</label>
    </div>

</fieldset>
```

$ compromising on semantics

```
~~<fieldset>~~
  <~~legend~~ strong>Shirt size</~~legend~~ strong>
  <div>
    <div>
      <input type="radio" id="s">
      <label for="s">S</label>
    </div>
    <div>
      <input type="radio" id="m">
      <label for="m">M</label>
    </div>
  </div>
~~</fieldset>~~
```



What's your name?

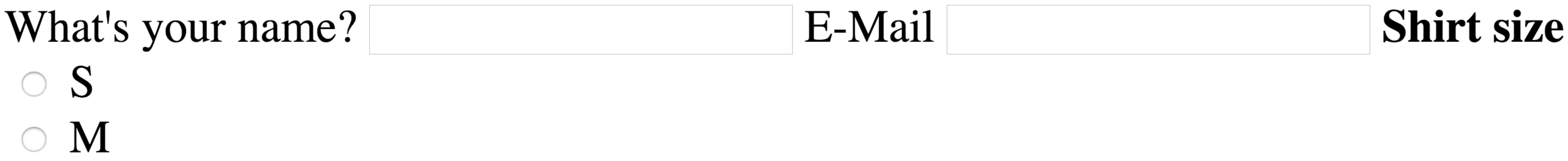
E-Mail

Shirt size

○ S

○ M

https://s.codepen.io/matuzo/pen/BOWYLB?editors=1100



What's your name? E-Mail **Shirt size**

○ S

○ M

The form displayed without CSS

COMPROMISING ON SEMANTICS

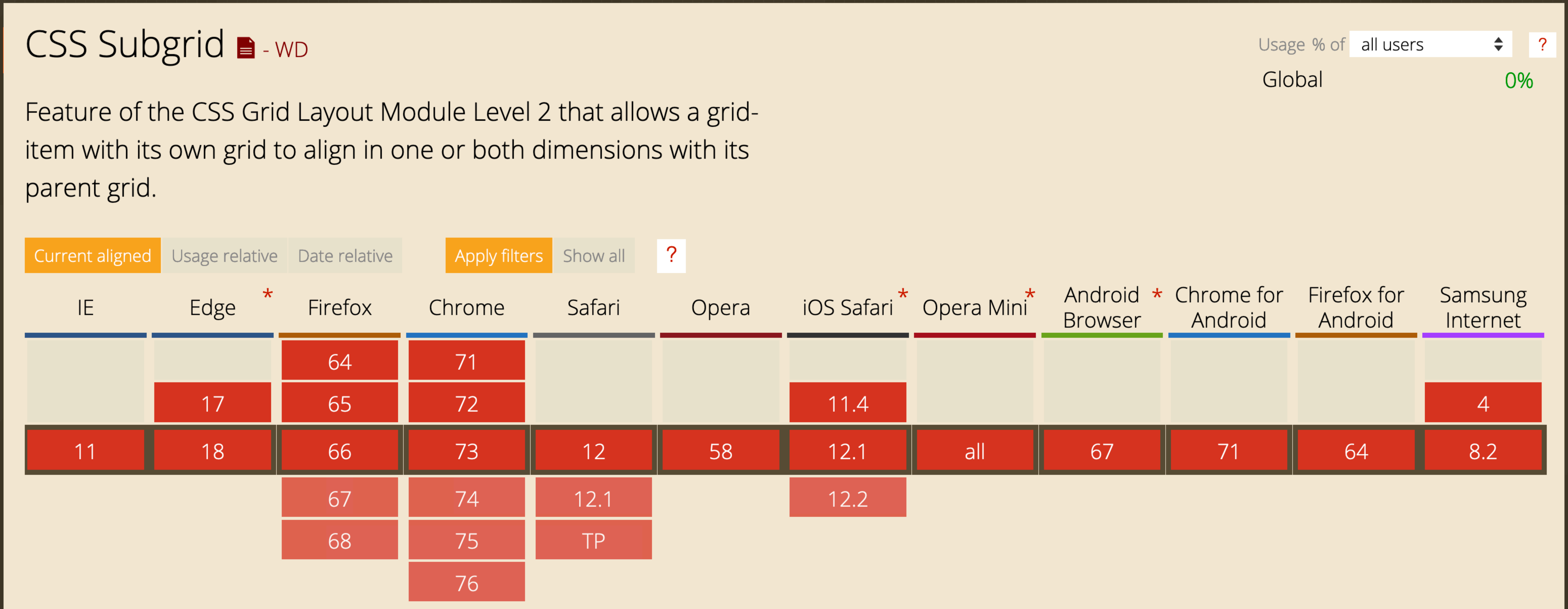
SUBGRID

> “A grid container that is itself a grid item can defer the definition of its rows and columns to its parent grid container, making it a subgrid.”

- https://drafts.csswg.org/css-grid-2/

$ compromising on semantics

```css
form > div,
fieldset {
  display: grid;
  grid-template-columns: subgrid;
}
```



CSS Subgrid - WD

Usage % of all users ?

Global 0%

Feature of the CSS Grid Layout Module Level 2 that allows a grid-item with its own grid to align in one or both dimensions with its parent grid.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android	Firefox for Android	Samsung Internet
		64	71								
	17	65	72			11.4					4
11	18	66	73	12	58	12.1	all	67	71	64	8.2
		67	74	12.1		12.2					
		68	75	TP							
			76								

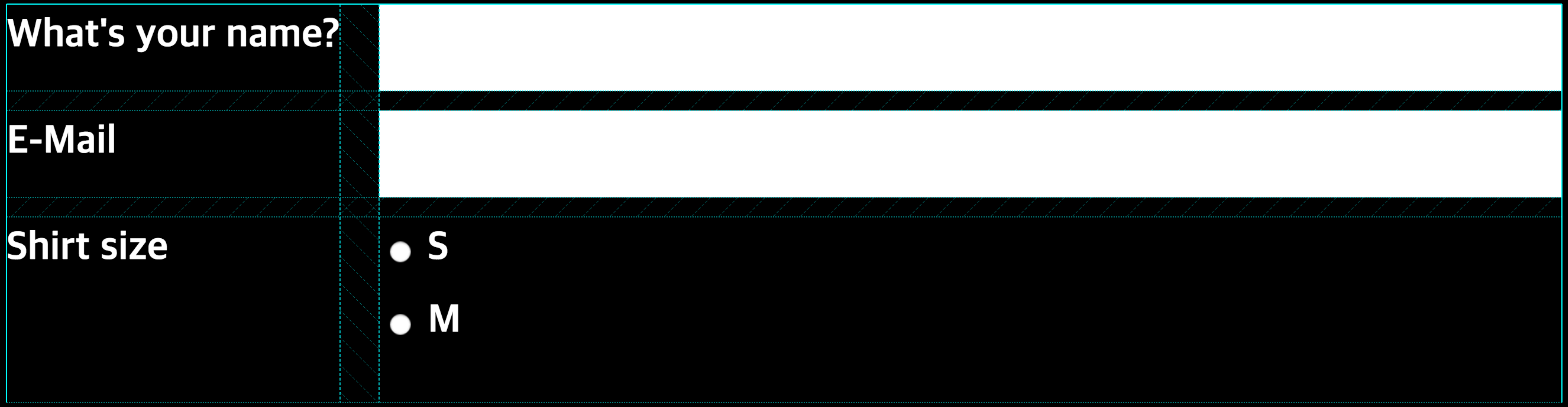
Subgrid will ship with level 2 of the specification

> “`display: contents` causes an element's children to appear as if they were direct children of the element's parent, ignoring the element itself.”

https://caniuse.com/#search=contents

$ compromising on semantics

```
form > div,
fieldset {
  display: contents;
}
```



What's your name?

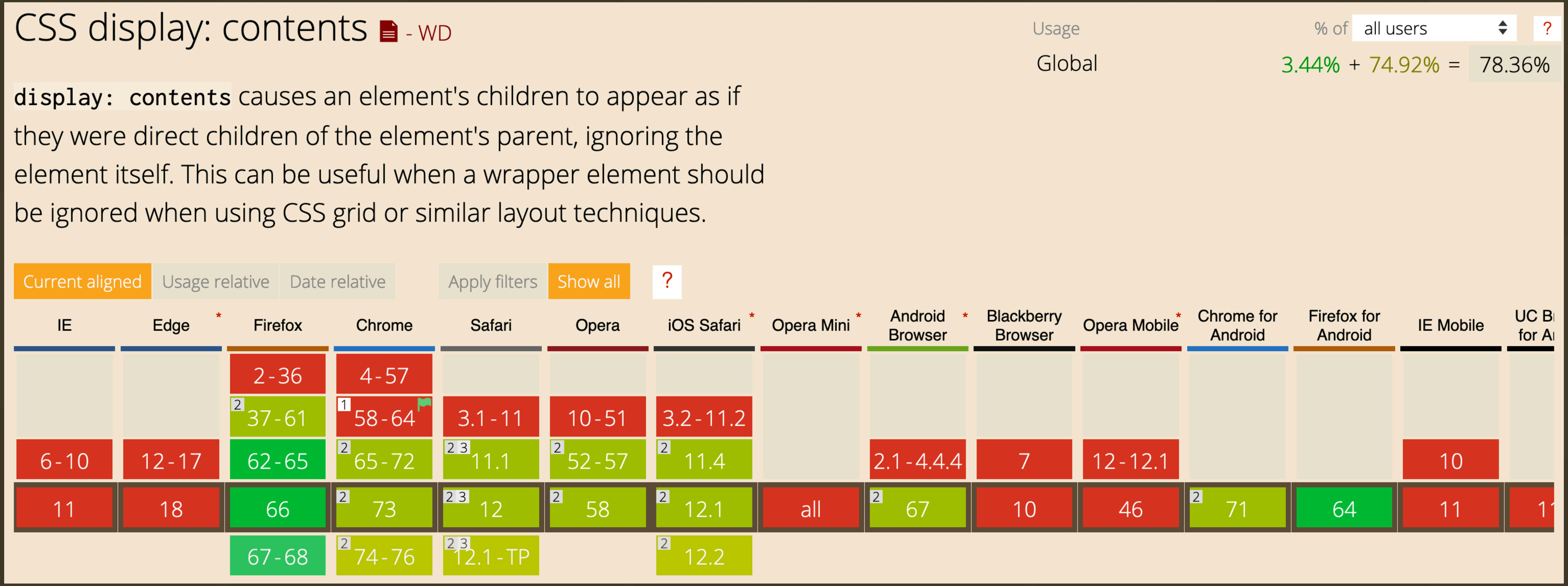
E-Mail

Shirt size

○ S

○ M

https://codepen.io/matuzo/pen/Mqrpbm



CSS display: contents - WD

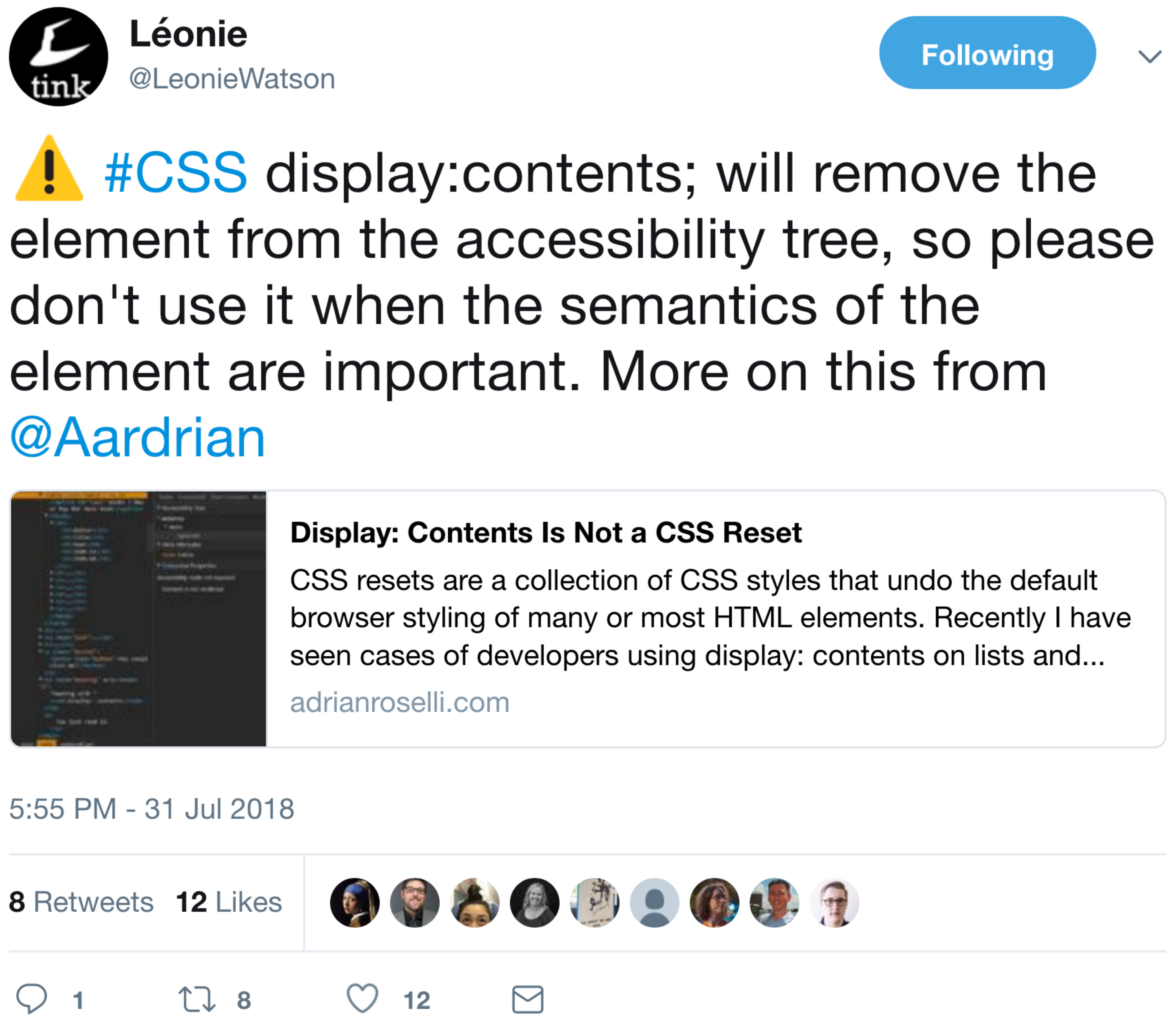
Usage % of all users ?

Global 3.44% + 74.92% = 78.36%

`display: contents` causes an element's children to appear as if they were direct children of the element's parent, ignoring the element itself. This can be useful when a wrapper element should be ignored when using CSS grid or similar layout techniques.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Br for A
		2 - 36	4 - 57											
		[2] 37 - 61	[1] 58 - 64	3.1 - 11	10 - 51	3.2 - 11.2								
6 - 10	12 - 17	62 - 65	[2] 65 - 72	[2 3] 11.1	[2] 52 - 57	[2] 11.4		2.1 - 4.4.4	7	12 - 12.1			10	
11	18	66	[2] 73	[2 3] 12	[2] 58	[2] 12.1	all	[2] 67	10	46	[2] 71	64	11	11
		67 - 68	[2] 74 - 76	[2 3] 12.1 - TP		[2] 12.2								

display: contents; is supported by all major desktop browsers except for Edge



Léonie
@LeonieWatson

⚠️ #CSS display:contents; will remove the element from the accessibility tree, so please don't use it when the semantics of the element are important. More on this from @Aardrian

Display: Contents Is Not a CSS Reset

CSS resets are a collection of CSS styles that undo the default browser styling of many or most HTML elements. Recently I have seen cases of developers using display: contents on lists and...

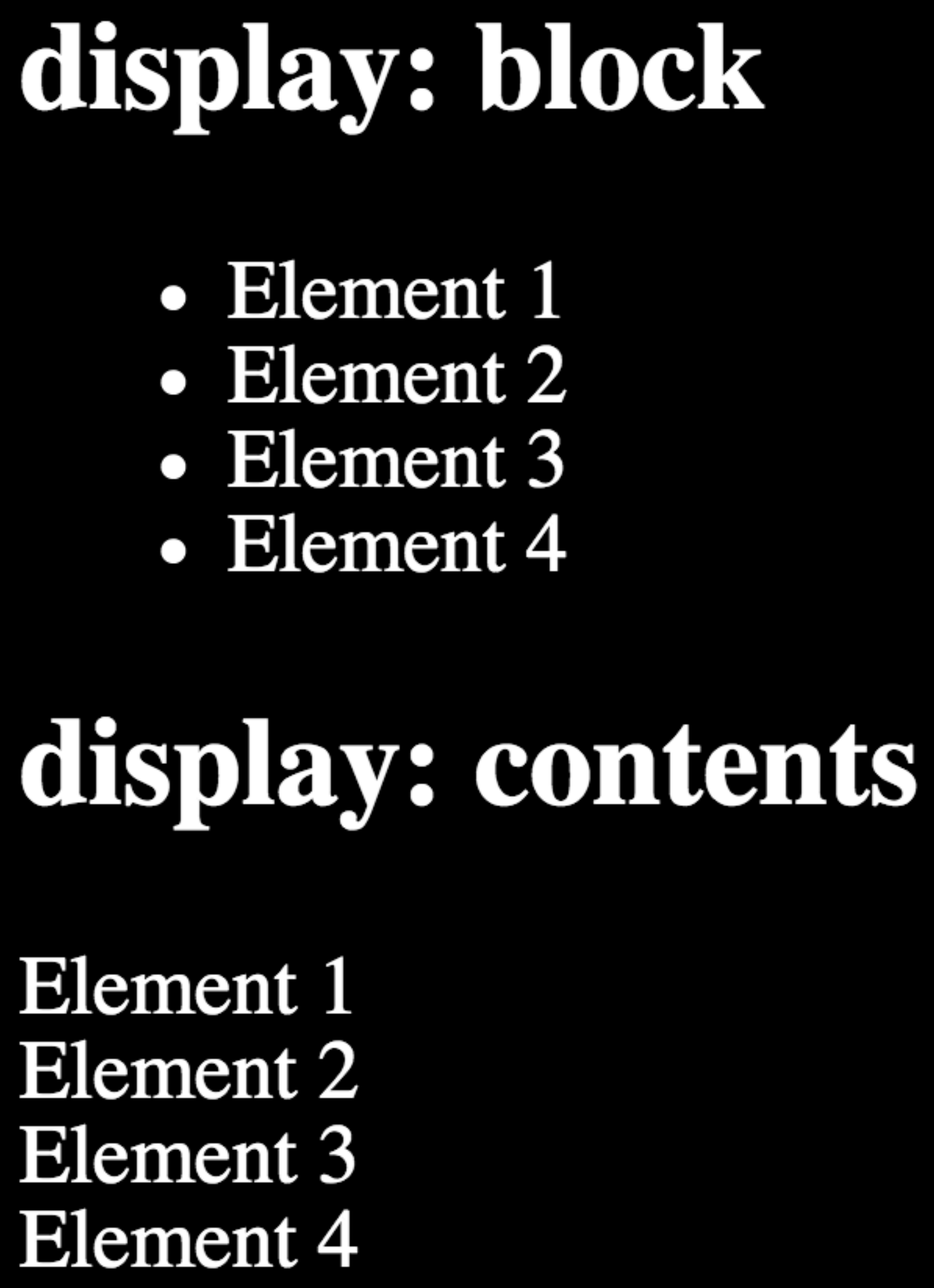
adrianroselli.com

5:55 PM - 31 Jul 2018

8 Retweets **12** Likes

Due to a bug display: contents; removes items from the accessibility tree.

$ compromising on semantics



display: block

- Element 1
- Element 2
- Element 3
- Element 4

display: contents

Element 1
Element 2
Element 3
Element 4

```
ul {
  display: contents;
}
```

PINK FLOYD FUN FACT #3

Monty Python and the Holy Grail

In 1975 Pink Floyd helped to finance the movie “Monty Python and the Holy Grail” by the comedy group Monty Python.

Some of the other investors were Led Zeppelin and Genesis.

@mmatuzo

CROSS BROWSER SUPPORT

Grids History

Microsoft shipped a Grid implementation behind the `-ms-` vendor prefix in IE 10 in 2011.

Chrome, Firefox, Opera, Safari and Edge shipped a different implementation throughout 2017.

Properties: Level 1 vs. MS Spec

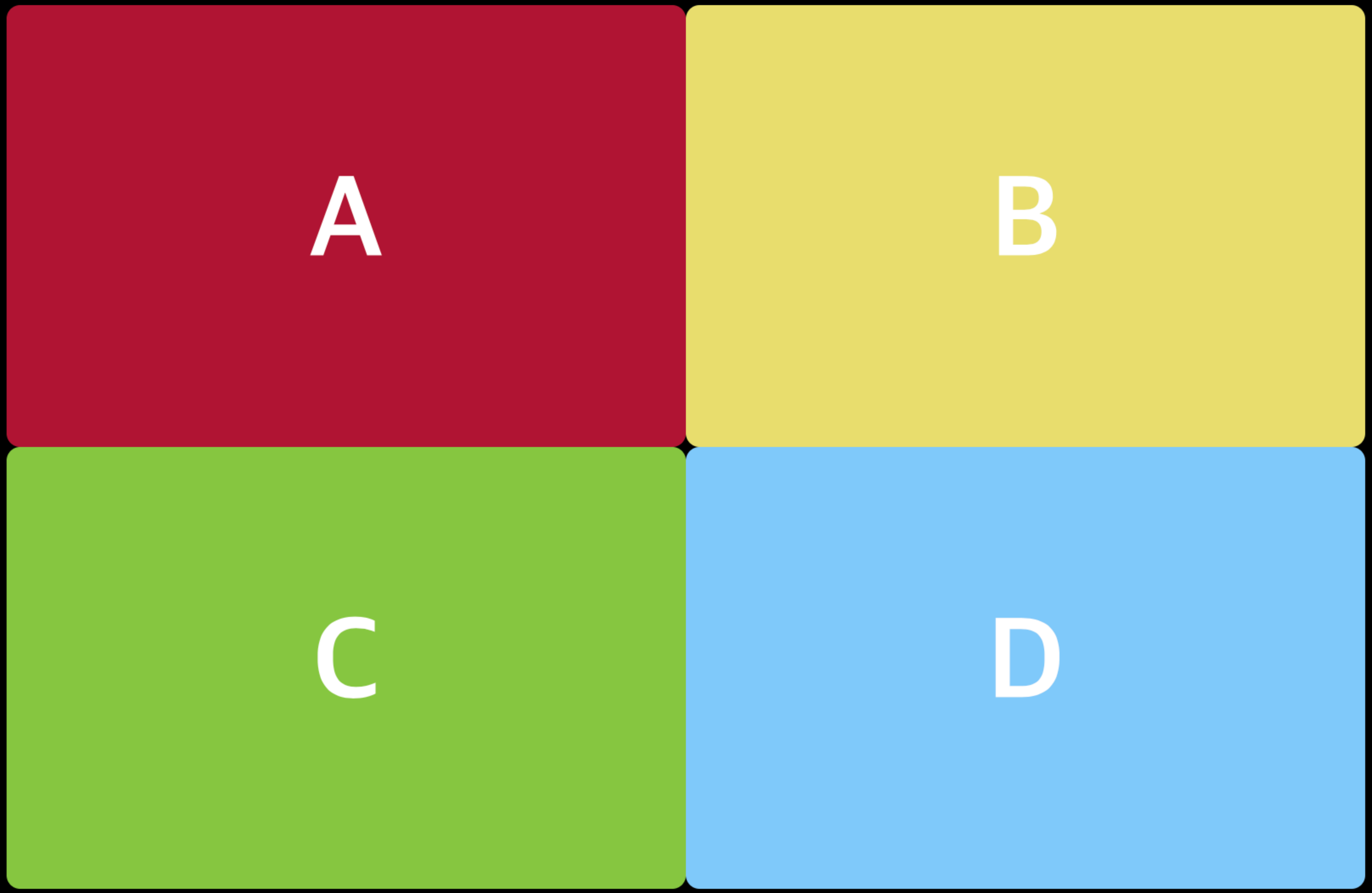
CR Level 1 property	IE10 implementation
grid-template-columns	-ms-grid-columns
grid-template-rows	-ms-grid-rows
grid-template-areas	x
grid-template	x
grid-auto-columns	x
grid-auto-rows	x
grid-auto-flow	x
grid	x
grid-row-start	-ms-grid-row
grid-column-start	-ms-grid-column
grid-row-end	x
grid-column-end	x

CR Level 1 property	IE10 implementation
grid-row	x
grid-column	x
grid-area	x
grid-row-gap	x
grid-column-gap	x
grid-auto-flow	x
grid-gap	x
-	-ms-grid-column-span
-	-ms-grid-row-span
align-self	-ms-grid-column-align
justify-self	-ms-grid-row-align

$ browser support

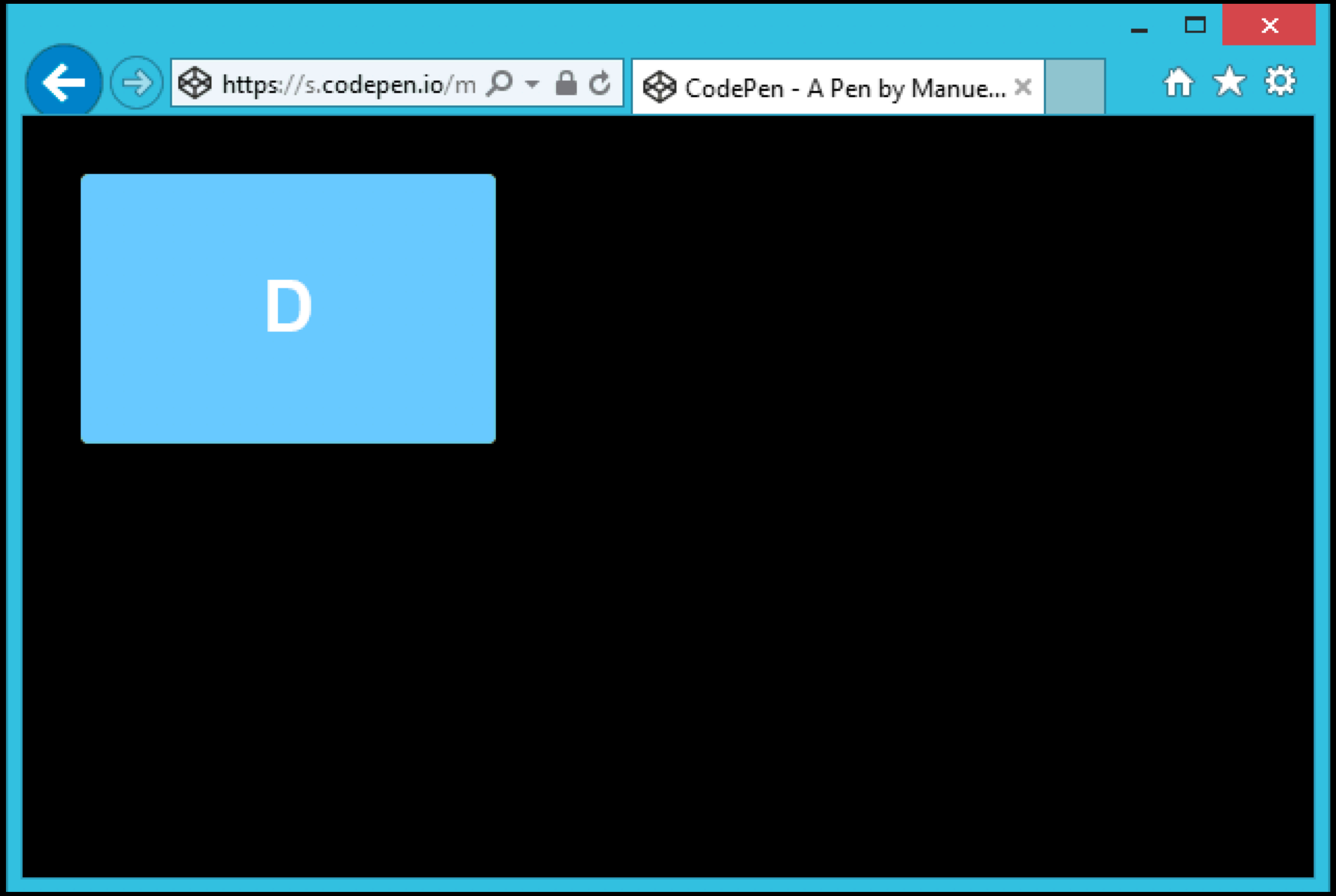
```
.grid {
  display: -ms-grid;
  -ms-grid-columns: 200px 200px;
  -ms-grid-rows: 130px 130px;

  display: grid;
  grid-template-columns: 200px 200px;
  grid-template-rows: 130px 130px;
}
```

Chrome



A	B
C	D

IE 11

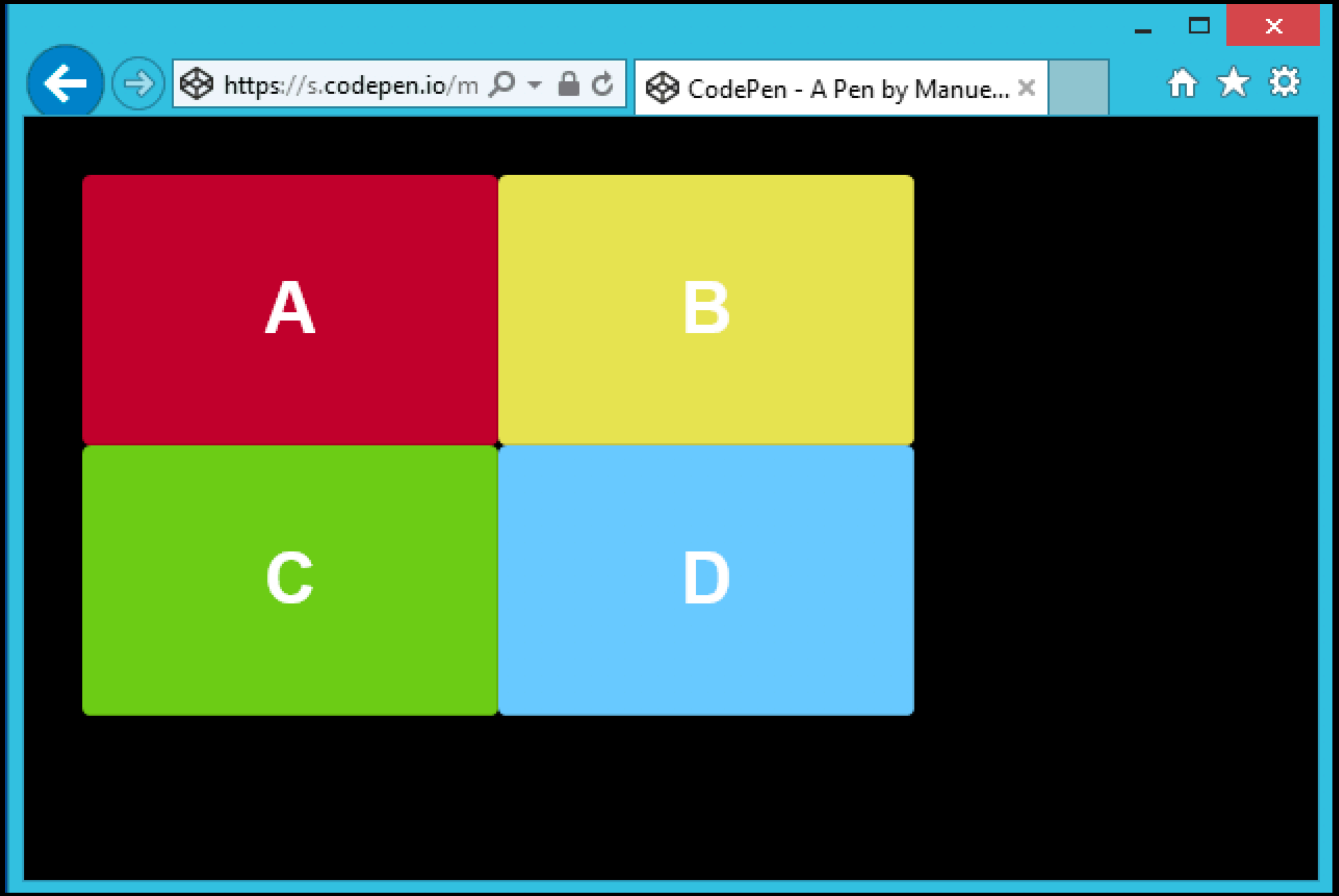


4 items in a 2 x 2 grid without explicit placement in Chrome and IE11

$ browser support

```
.item:nth-child(1) {
  -ms-grid-column: 1;
  -ms-grid-row: 1;
}

.item:nth-child(2) {
  -ms-grid-column: 2;
  -ms-grid-row: 1;
}
```

```
.item:nth-child(3) {
  -ms-grid-column: 1;
  -ms-grid-row: 2;
}

.item:nth-child(4) {
  -ms-grid-column: 2;
  -ms-grid-row: 2;
}
```



4 items in a 2 x 2 grid with explicit placement in IE11

BROWSER SUPPORT

AUTOPREFIXER?

@mmatuzo

```
$ autoprefixer
```

```
module.exports = {
  plugins: [
    require('autoprefixer')({
      grid: "autoplace",
    })
  ]
}
```

main.prefixed.css — darksideofthegrid
# main.prefixed.css ●
Ln 1, Col 8
Spaces: 2
UTF-8
LF
CSS
Prettier

```
: 100px 100px 100px;
00px 200px 200px;
```

```css
.grid {
  display: -ms-grid;
  display: grid;

  -ms-grid-columns: 100px 100px 100px;
  grid-template-columns: 100px 100px 100px;

  -ms-grid-rows: 200px 200px 200px;
  grid-template-rows: 200px 200px 200px;

  grid-gap: 10px;
}

```

Autoprefixer with the grid setting set to "autoplace".

main.prefixed.css — darksideofthegrid
# main.prefixed.css
0px 200px 200px;

```css
.grid {
  display: -ms-grid;
  display: grid;

  -ms-grid-rows: 100px 10px 100px 10px 100px;
  -ms-grid-columns: 200px 10px 200px 10px 200px;

  grid-template: 100px 100px 100px / 200px 200px 200px;
  grid-gap: 10px;
}

```

Ln 5, Col 1 (94 selected)   Spaces: 2   UTF-8   LF   CSS   Prettier

Autoprefixer fakes gaps by adding rows and columns in IE 10 and IE 11.

Autoprefixer gotchas

`grid-template` or **`grid-template-areas`** + **`grid-template-columns`** mandatory for gap support.

The usage of duplicate area names is limited.

Negative line numbers, e.g. **`grid-column: 1 / -1;`** are not supported.

Autoprefixer gotchas

`grid` shorthand not supported.

`align-content`, `justify-content`, and `place-content` don’t work.

It’s not possible to define end-lines only.

Limited auto placement.

Autoplacement in Autoprefixer

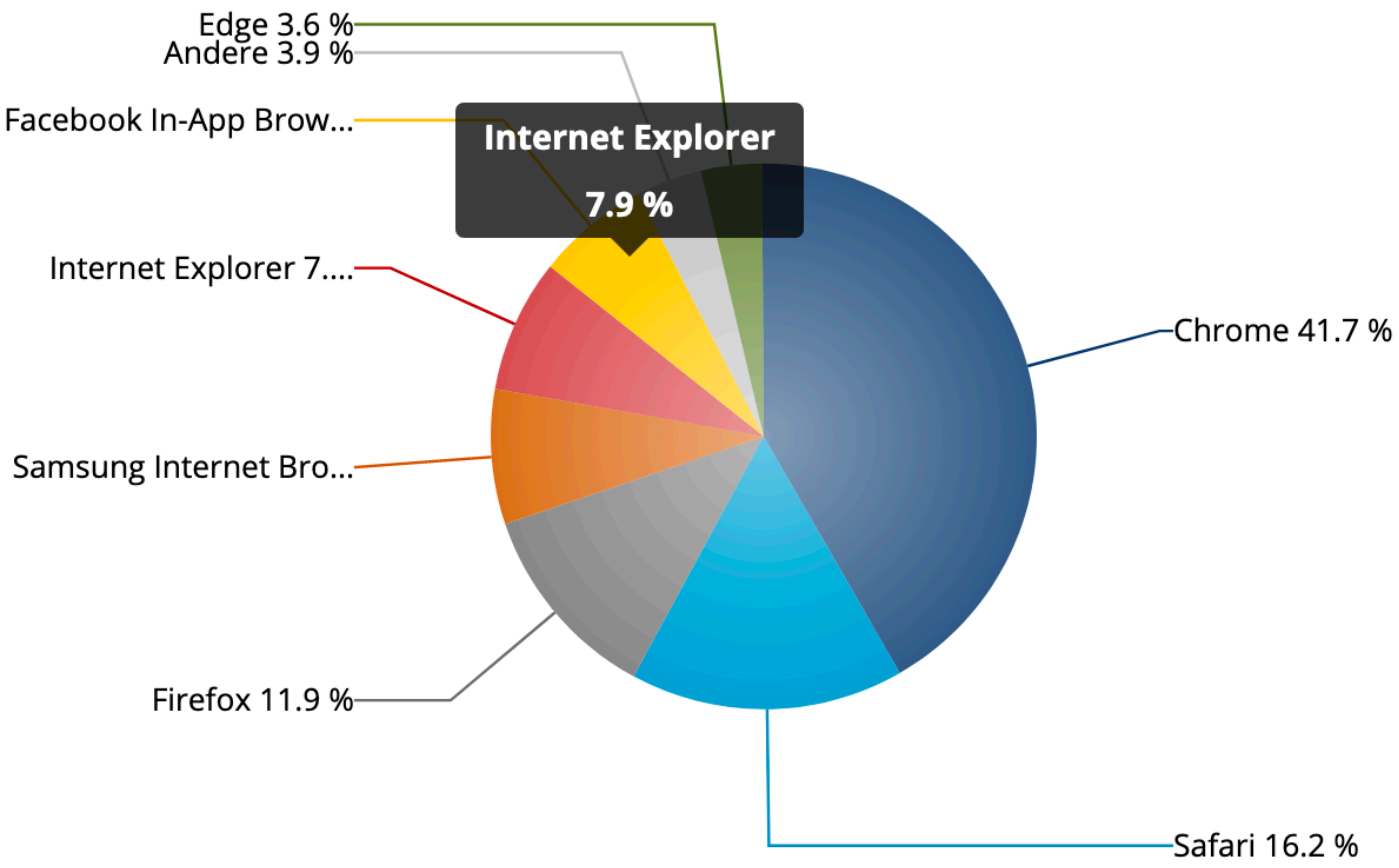
Both columns and rows must be defined.

Repeat `auto-fit` and `auto-fill` are not supported.

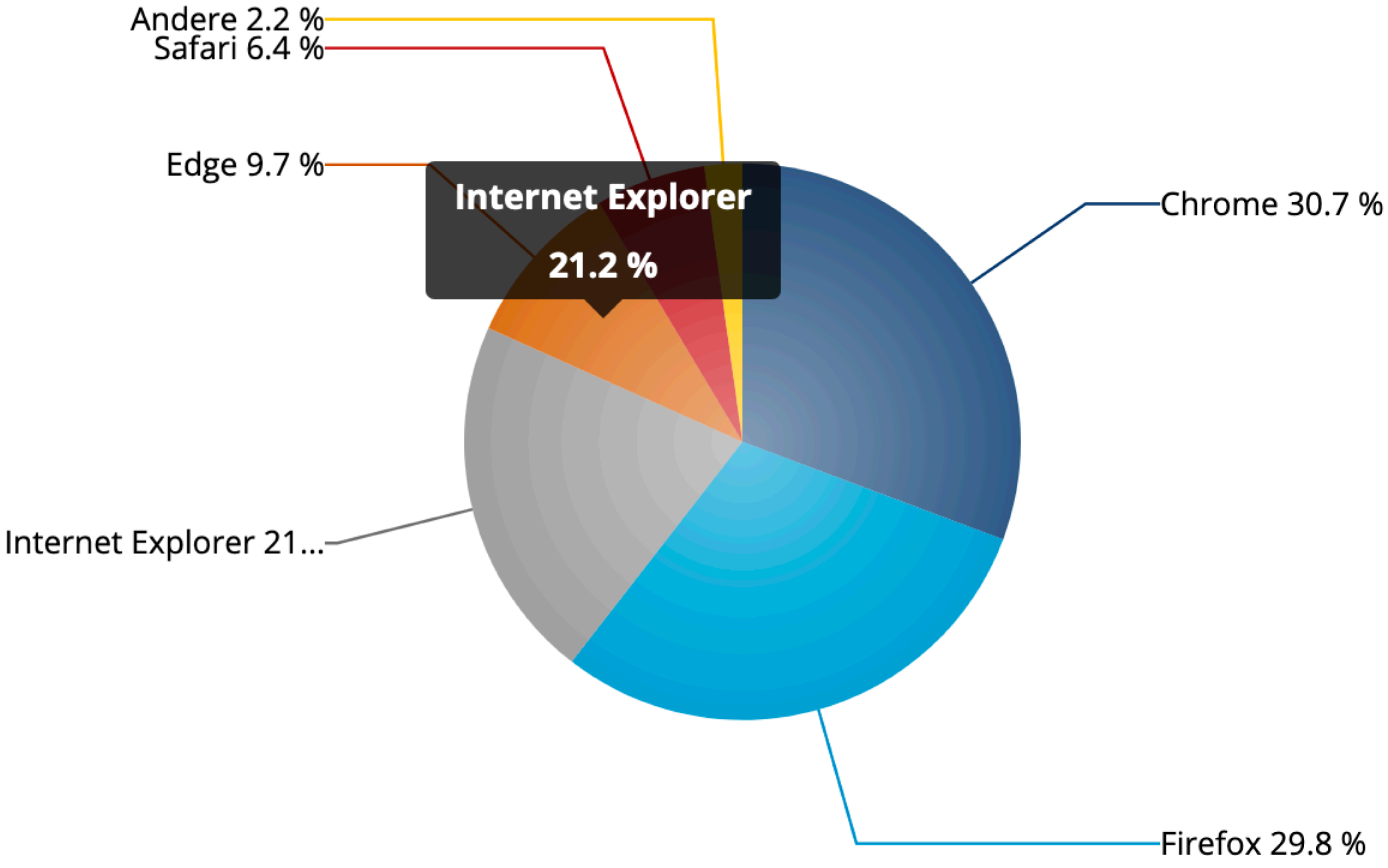
Explicit and implicit placement must not be mixed.

Auto-placement of the pseudo elements `::before` and `::after` is not supported.

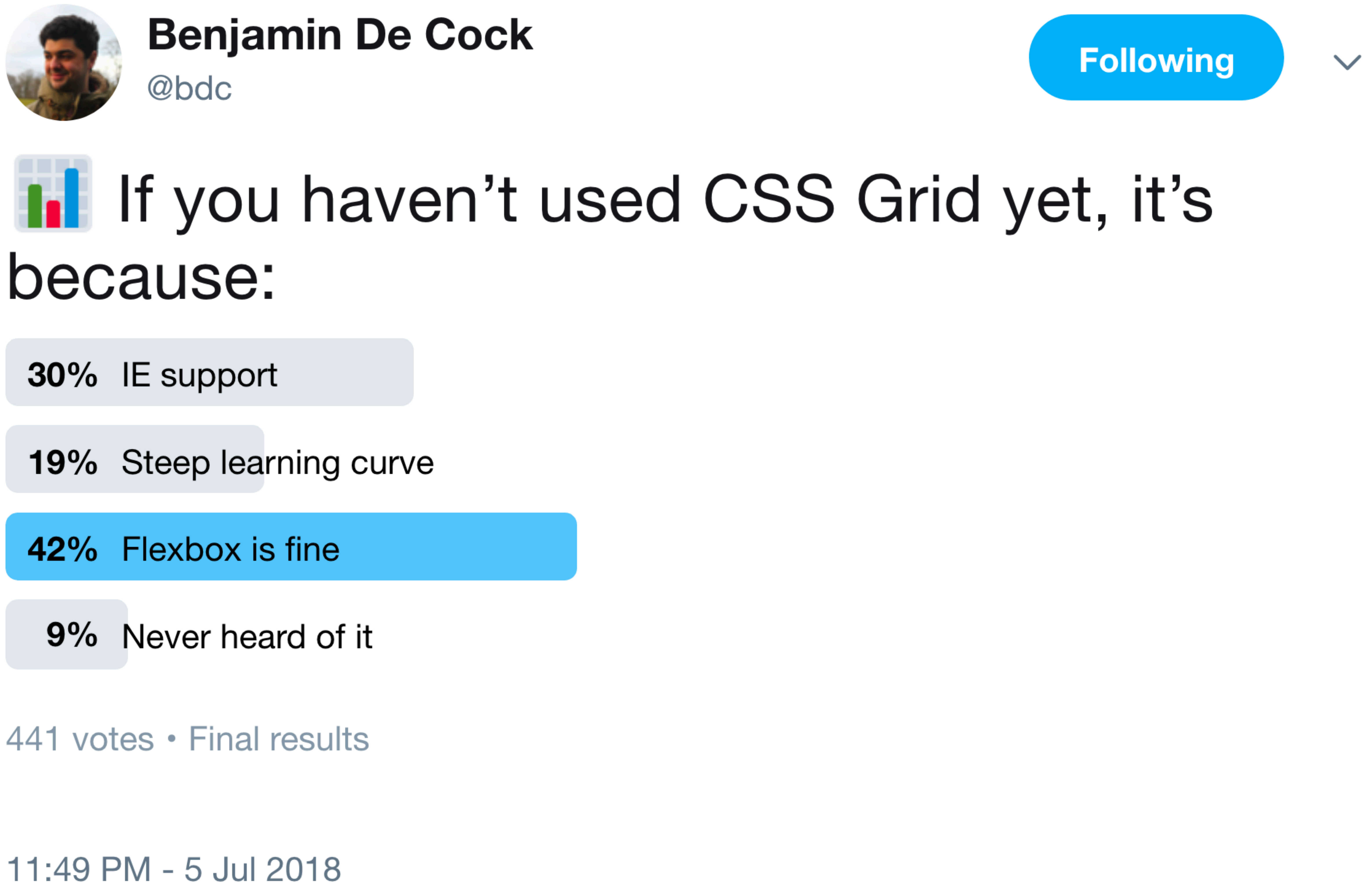
When changing the `grid-gap` value, columns and rows must be re-declared.



7.9% IE users in Austria (total).



21.2% IE users in Austria (Desktop users only).



Benjamin De Cock
@bdc

📊 If you haven't used CSS Grid yet, it's because:

30% IE support

19% Steep learning curve

42% Flexbox is fine

9% Never heard of it

441 votes • Final results

11:49 PM - 5 Jul 2018

Internet Explorer support is a barrier for many developers.

> “I don’t think we owe it to any users to make it all exactly the same. Therefore we can get away with keeping fallbacks very simple.
> My hypothesis:
> users don’t mind, they’ve come for the content.”

-Hidde de Vries
hiddedevries.nl/en/blog/2018-08-11-lets-serve-everyone-good-looking-content

There is no dark side of the ~~moon~~ grid really.
Matter of fact it's all dark.

PINK FLOYD: ECLIPSE

Responsible use of Grid

Avoid `grid-template-areas` or any other reordering techniques on interaction sensitive components.

Test with your keyboard.

Don’t compromise on semantics.

Responsible use of Grid

If you’re using autoprefixer make sure everyone in your team is aware of its limitations and what it does behind the scenes.

Make sure your components work in older and less capable browsers.

Embrace the power of progressive enhancement.

Slides: bit.ly/dark-grid-linz

Danke! 🥰

manuel@matuzo.at

🐦 @mmatuzo