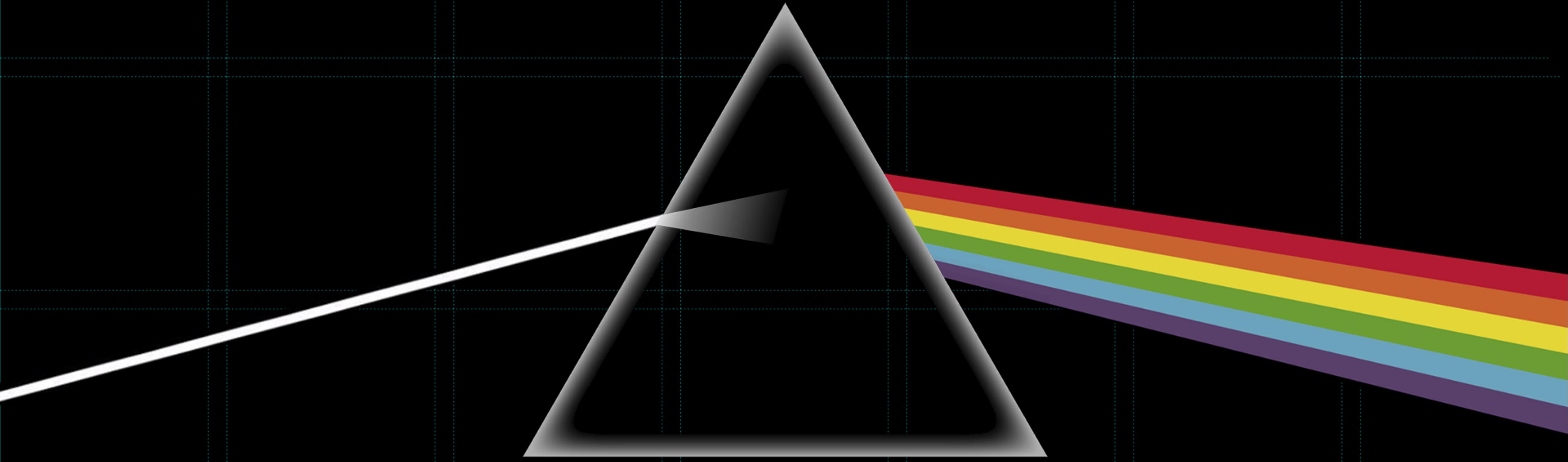


An aerial photograph of Vienna's Museum Quarter. In the foreground, the ornate facade of the Kunsthistorisches Museum (Museum of Fine Arts) is visible, featuring its characteristic domes and classical architecture. To the right, the Naturhistorisches Museum (Museum of Natural History) is partially visible. A large, well-maintained green space, the Volksgarten, stretches across the middle ground, filled with trees, manicured lawns, and paved walkways. In the background, the city's skyline extends towards the horizon under a clear blue sky.

Manuel Matuzović

@mmatuzo

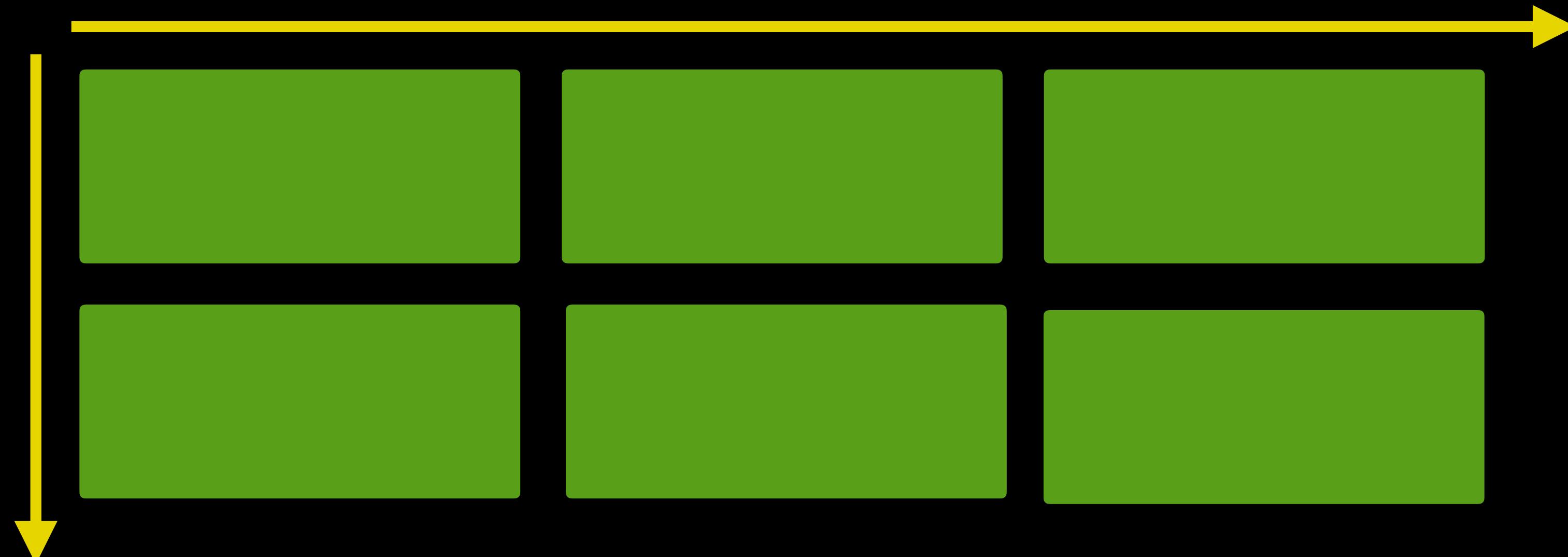
Museum of Fine Arts, Museum of Natural History, Volksgarten und parliament in Vienna



The dark side of the grid

CSS Grid Layout?

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

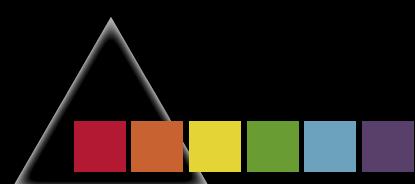


What's special about Grid?

The first true layout method in CSS.

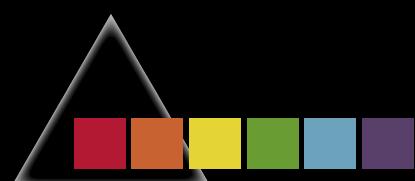
float, display:inline-block, position,
display:table, display: flex not designed for
building two-dimensional layouts.

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



Grid-related properties and values

display	justify-content	grid-column
grid-template-columns	align-content	grid-row
grid-template-rows	place-content	grid-area
grid-template-areas	grid-auto-columns	justify-self
grid-template	grid-auto-rows	align-self
grid-column-gap	grid-auto-flow	place-self
grid-row-gap	grid	min-content, max-content, fit-content
grid-gap	grid-column-start	fr
justify-items	grid-column-end	repeat()
align-items	grid-row-start	minmax()
place-items	grid-row-end	





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.



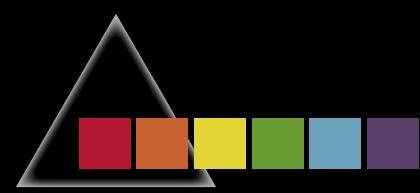
#1 Thriller by Michael Jackson (66 Million)

#2 Their Greatest Hits (1971-1975) by Eagles (51 Million)

#3 Back in Black by AC/DC (50 Million)



CHANGING VISUAL ORDER



Changing Visual Order

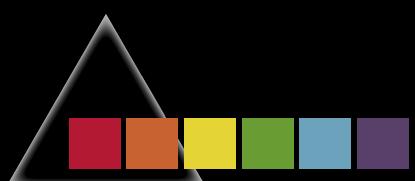
Explicit placement

order property

Absolute positioning

Auto flow

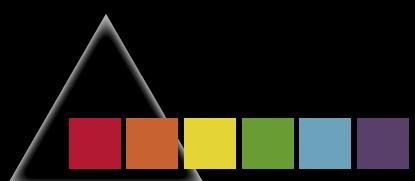
Areas



Visual Order

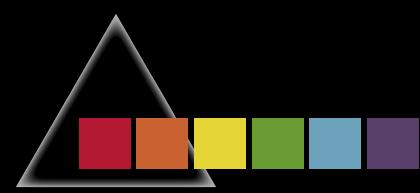
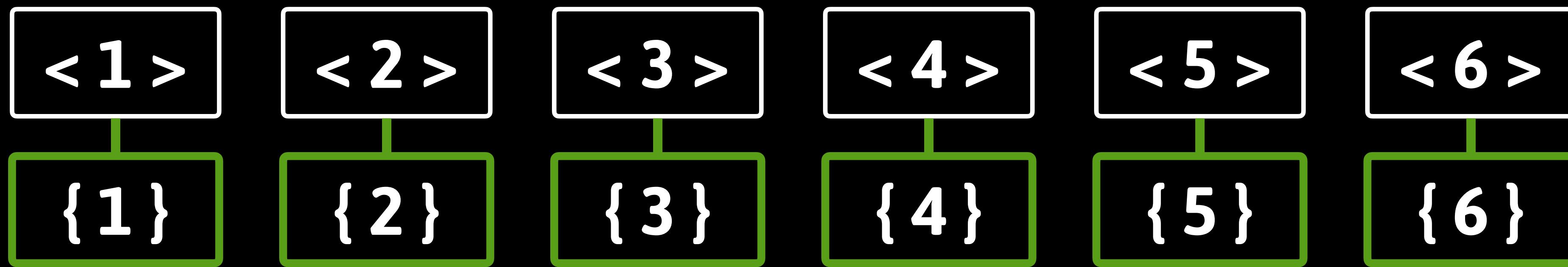
Both the tab order and the order in which screen readers read content follow DOM order.

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



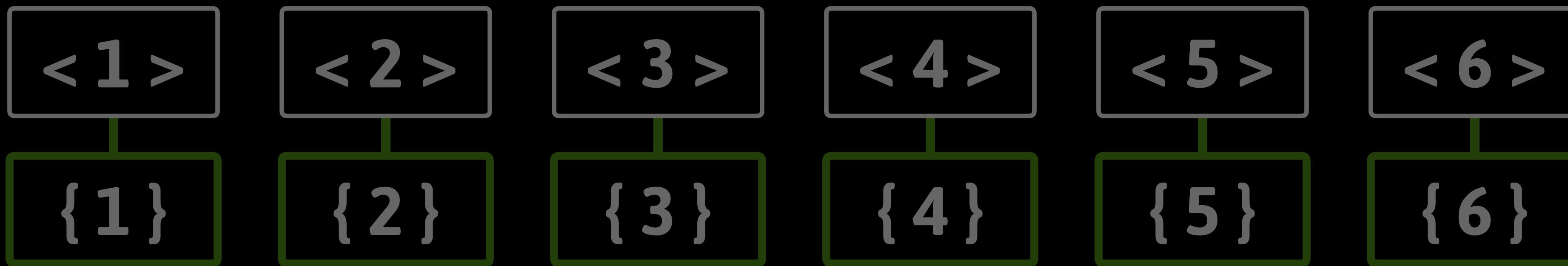
Visual Order

Visual order matches DOM order

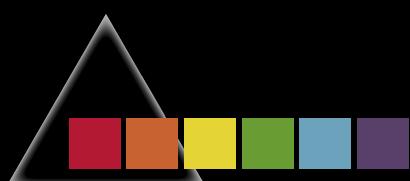
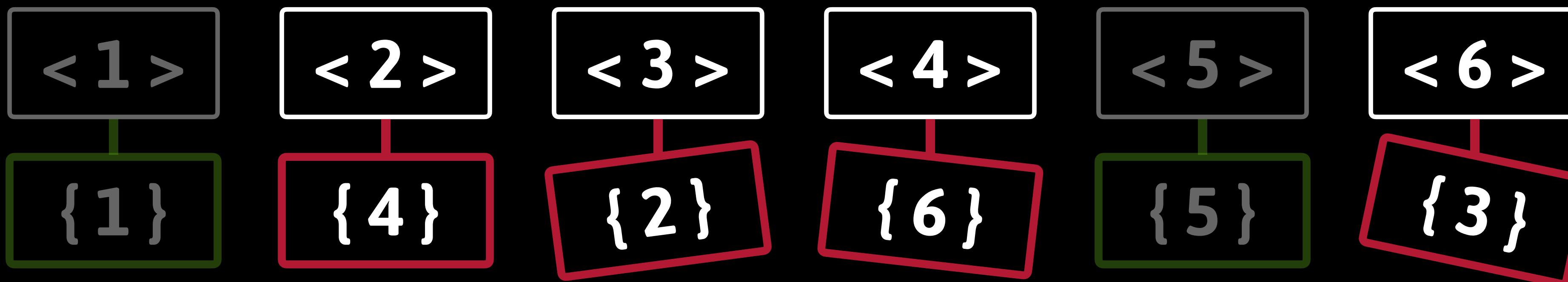


Visual Order

Visual order matches DOM order



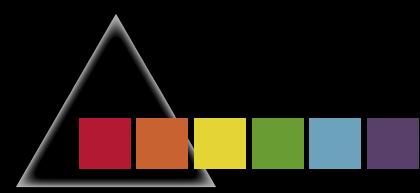
Visual order doesn't match DOM order



V I S U A L O R D E R

E X P L I C I T

P L A C E M E N T



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

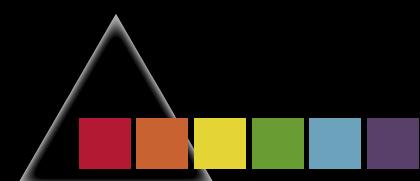
grid-column-start

grid-column-end

grid-row

grid-row-start

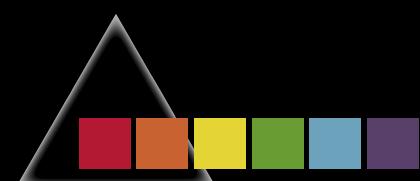
grid-row-end



\$ explicit placement

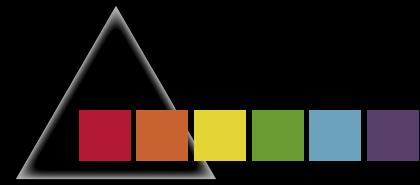
```
.item:nth-child(1) {  
    grid-column: 2;  
    grid-row: 2;  
}  
...
```

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



VISUAL ORDER

AREAS



HEADER

MAIN CONTENT

ADS

NEWSLETTER

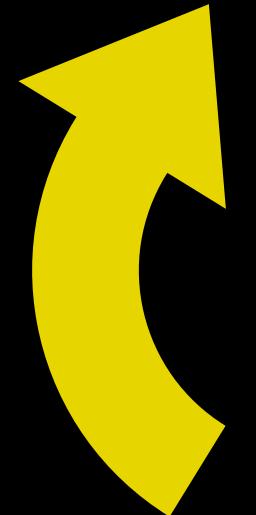
FOOTER

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";  
}
```



HEADER

NEWSLETTER

MAIN CONTENT

ADS

FOOTER

<https://s.codepen.io/matuzo/debug/ZMLpML>



@mmatuzo

```
$ areas  
body {  
  grid-template-columns: 1fr 1fr;  
  grid-template-areas:  
    "header header"  
    "content content"  
    "ads newsletter"  
    "footer footer";  
}
```

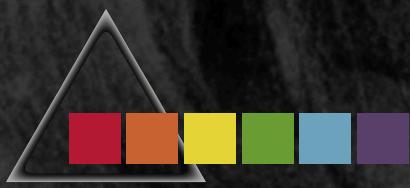


A black and white photograph of the rock band Pink Floyd. From left to right, it shows Roger Waters, Nick Mason, Syd Barrett (partially visible), and David Gilmour. They are all wearing dark clothing and have their eyes closed, suggesting they are performing or in a meditative state. The background is dark and textured.

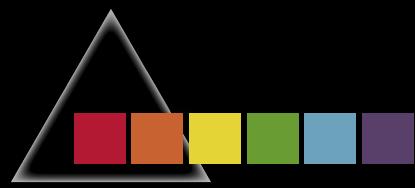
PINK FLOYD FUN FACT #2

736 weeks on the Billboard chart

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



COMPROMISING ON SEMANTICS



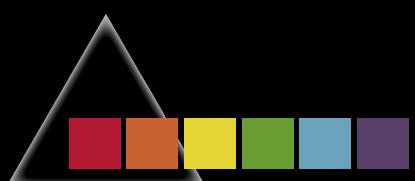
“ I believe there will be a strong temptation,
especially with Grid, to flatten out document structure
in order that all elements become a child of the
element with the Grid declared.
Making layout simple, but at what cost? ”

Rachel Andrew

[\(rachelandrew.co.uk/archives/2015/07/28/modern-css-layout-power-and-responsibility\)](http://rachelandrew.co.uk/archives/2015/07/28/modern-css-layout-power-and-responsibility)

\$ compromising on semantics

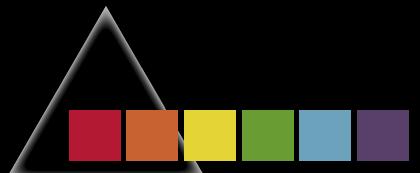
```
<form>  
  <div>  
    <label for="name">What's your name?</label>  
    <input type="text" id="name" />  
  </div>  
  <div>  
    <label for="email">E-Mail</label>  
    <input type="email" id="email" />  
  </div>  
  ...
```



\$ 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

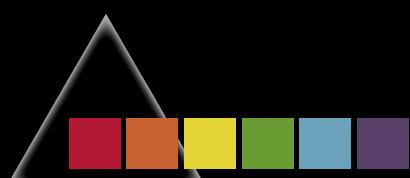
<https://s.codepen.io/matuzo/pen/BOWYLB?editors=1100>



@mmatuzo

\$ compromising on semantics

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



What's your name?

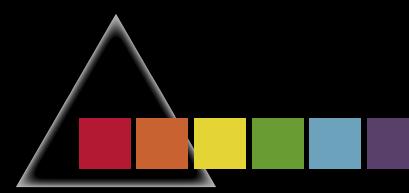
E-Mail

—Shirt size—

- S
- M

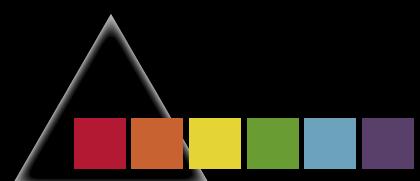
<https://s.codepen.io/matuzo/pen/BOWYLB?editors=1100>

@mmatuzo



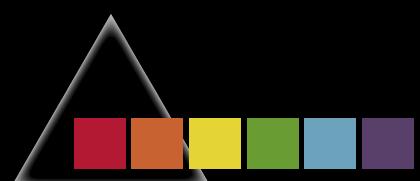
\$ compromising on semantics

```
<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>
```



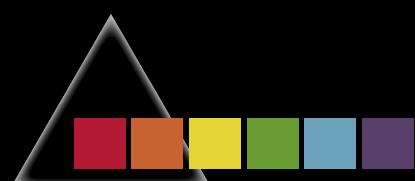
\$ compromising on semantics

```
<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>
```



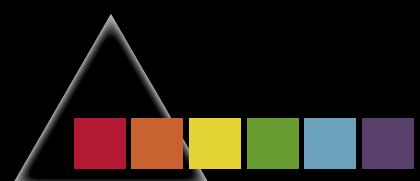
\$ compromising on semantics

```
<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>
```



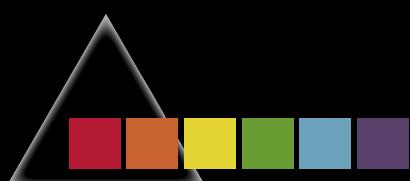
What's your name?

E-Mail

Shirt size

- S
- M

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



@mmatuzo

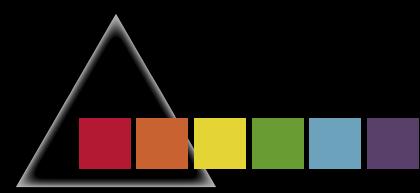
What's your name? E-Mail **Shirt size**

S
 M

The form displayed without CSS

C O M P R O M I S I N G O N S E M A N T I C S

S U B G R I D



“ 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

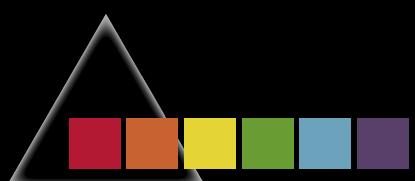
form > div,

fieldset {

display: grid;

grid-template-columns: subgrid;

}



Can I use

subgrid

?  Settings

0 results found.

Why not suggest this feature be added?

Detected your country as "Austria". Would you like to import usage data for that country?

Import

No thanks

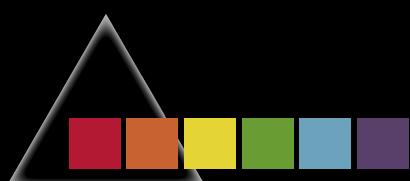
Subgrid will ship with level 2 of the specification

“ 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;  
}
```



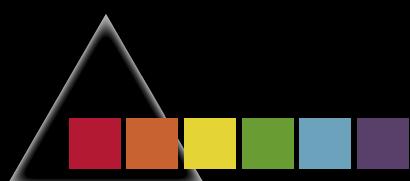
What's your name?

E-Mail

Shirt size

- S
- M

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



@mmatuzo

CSS display: contents WD

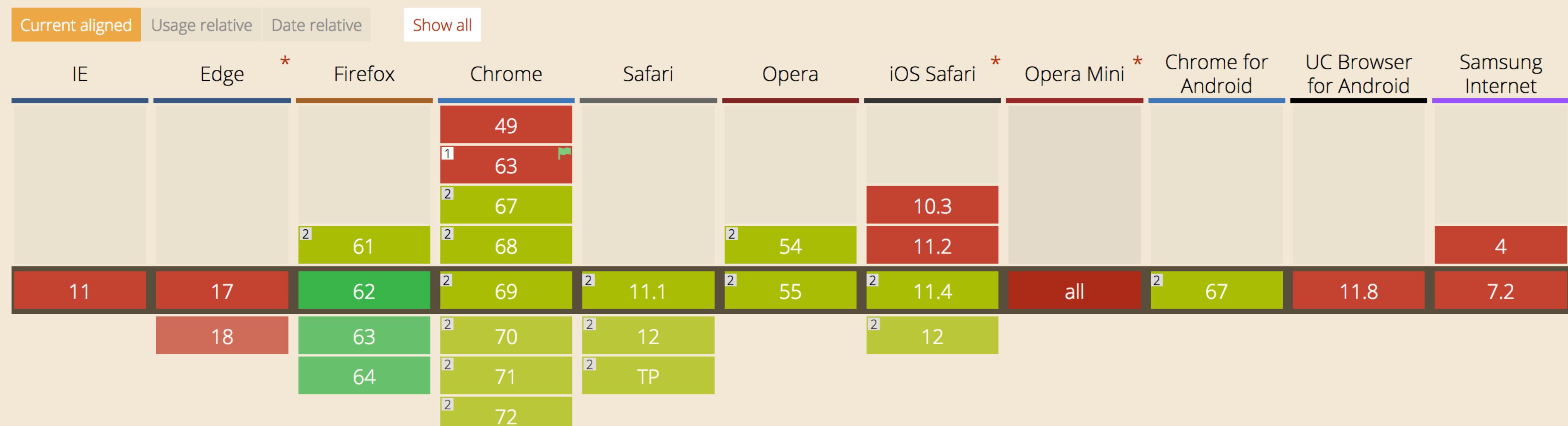
Usage

% of all users

Global

0.1% + 72.91% = 73.01%

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.



Notes

Known issues (1)

Resources (6)

Feedback

¹ Enabled in Chrome through the "experimental Web Platform features" flag in chrome://flags

² Partial support refers to severe implementation bug that renders the content inaccessible. <https://hiddedevries.nl/en/blog/2018-04-21-more-accessible-markup-with-display-contents>

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



Léonie

@LeonieWatson

Following

⚠ #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



1



8



12



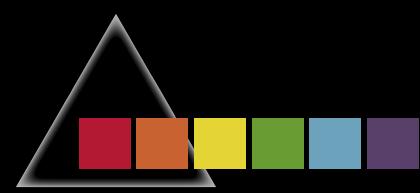
Due to a bug `display: contents;` removes items from the accessibility tree.

Old at AdrianRoselli.com

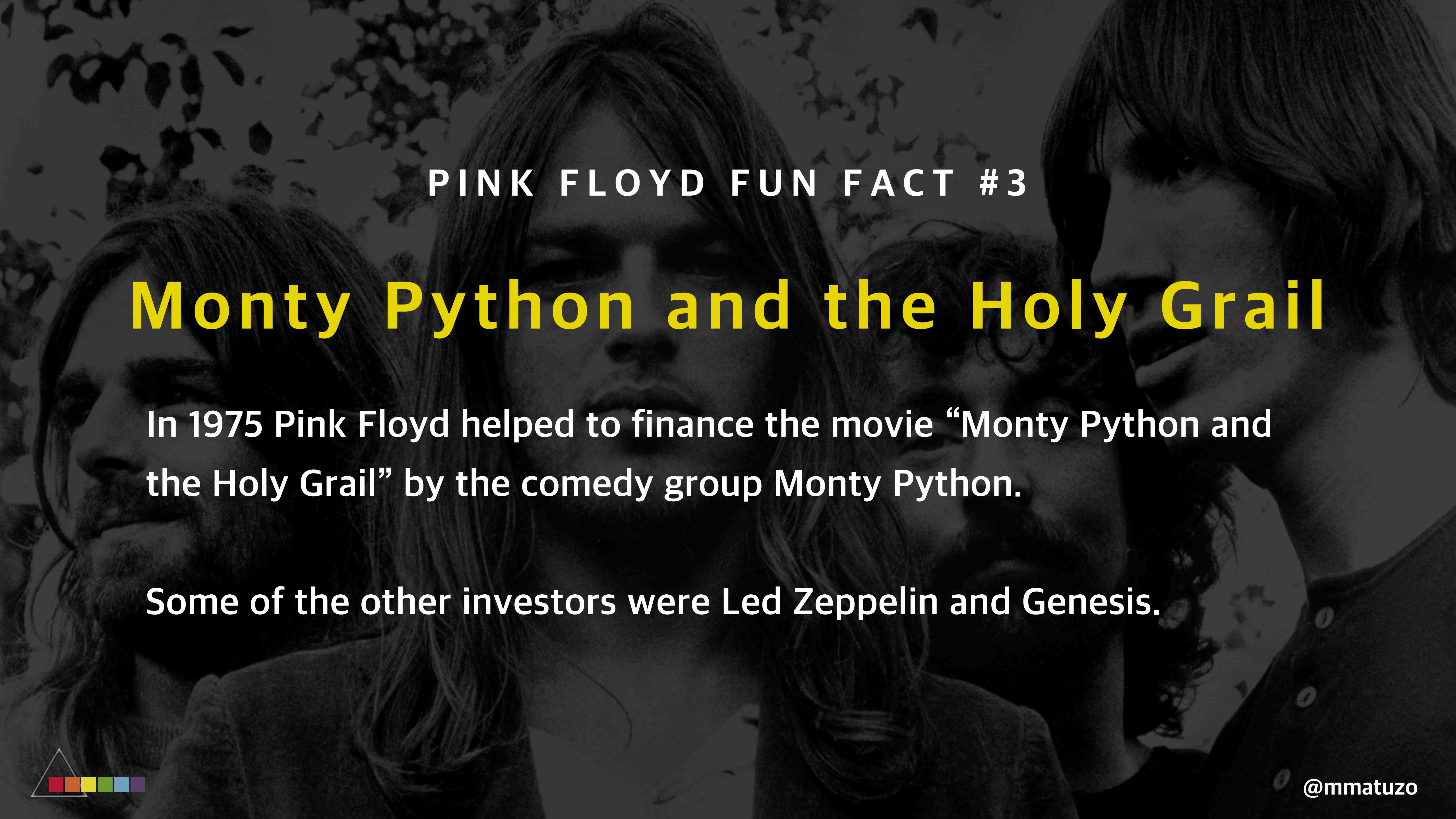
**Do not use
display:contents as it removes
semantics from your HTML**

<http://adrianroselli.com>

<https://twitter.com/aardrian/status/1052027923786620928>



@mmatuzo

A black and white photograph of the rock band Pink Floyd. The members are shown from the chest up, looking slightly downwards and to the right. They are wearing dark clothing. The background is dark and textured.

PINK FLOYD FUN FACT #3

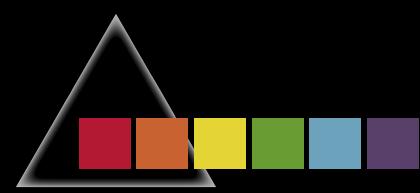
Monty Python and the Holy Grail

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.



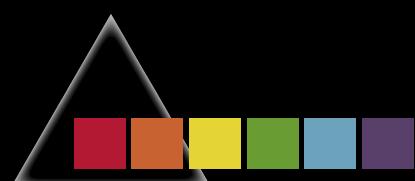
CROSS BROWSER SUPPORT



Grids History

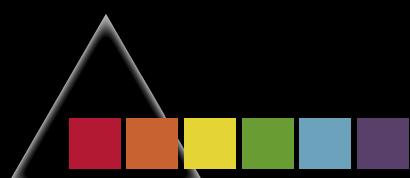
Microsoft shipped a Grid implementation behind the
-ms- vendor prefix in IE 10 in 2011.

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

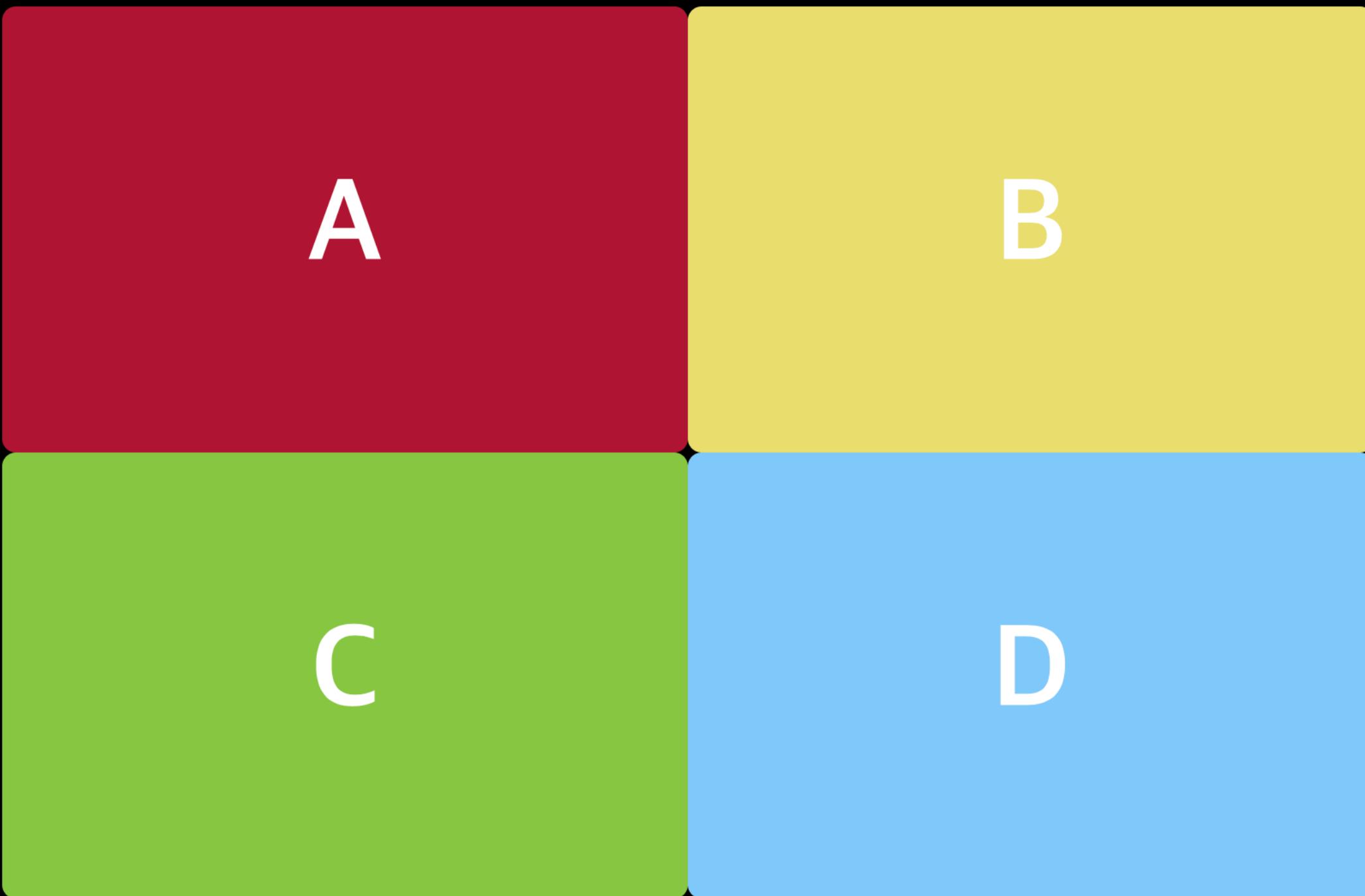


```
$ browser support
```

```
.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



IE 11

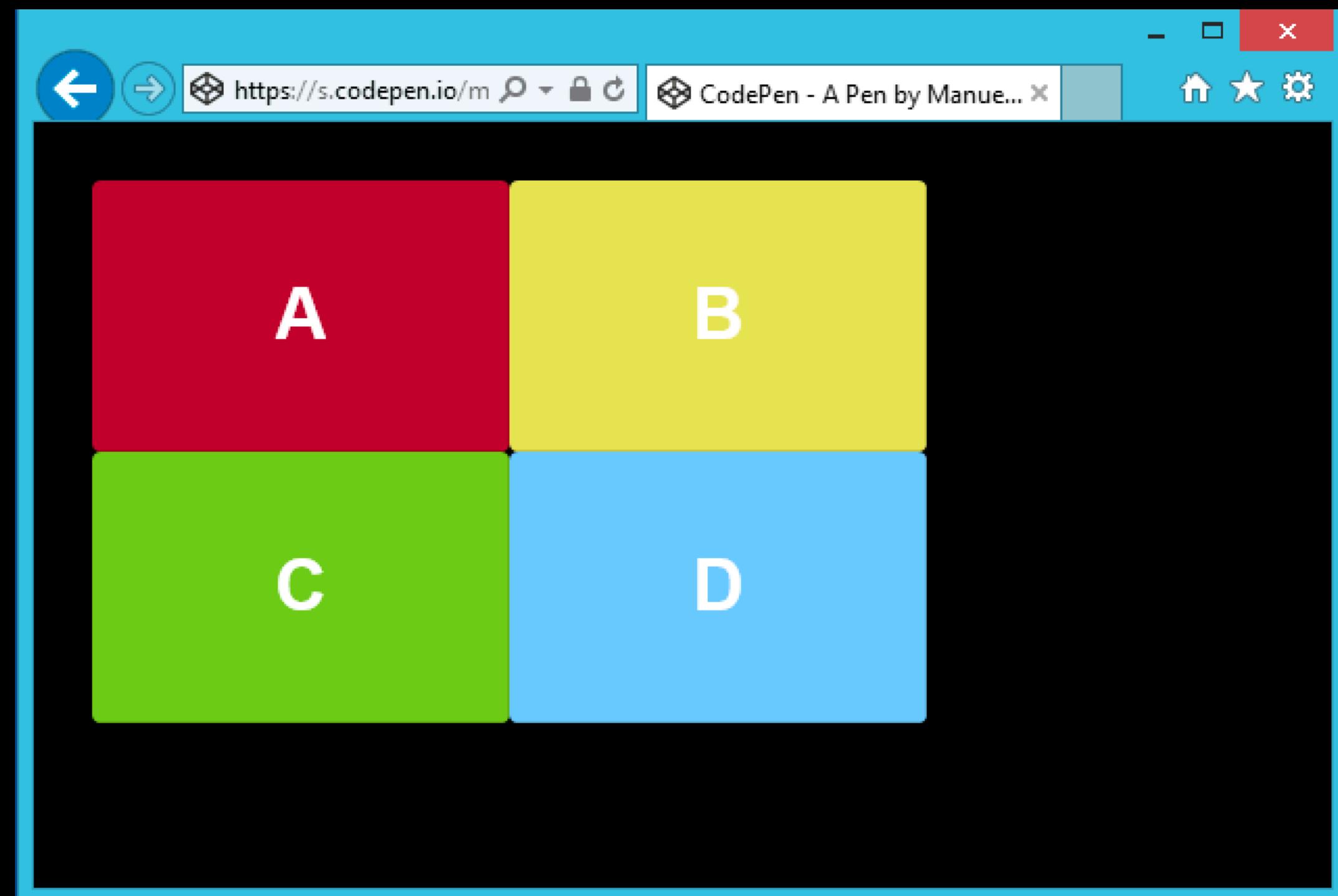


4 items in a 2×2 grid without explicit placement in Chrome and IE11

```
$ browser support
```

```
.item:nth-child(1) { .item:nth-child(3) {  
    -ms-grid-column: 1;  
    -ms-grid-row: 1;  
}  
  
.item:nth-child(2) { .item:nth-child(4) {  
    -ms-grid-column: 2;  
    -ms-grid-row: 1;  
}  
}
```

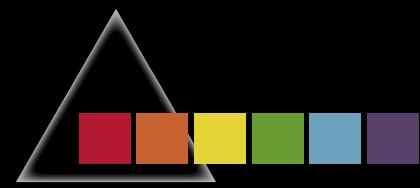




4 items in a 2×2 grid with explicit placement in IE11

BROWSER SUPPORT

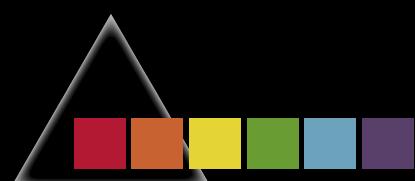
AUTOPREFIXER?



@mmatuzo

```
$ autoprefixer
```

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



```
# main.css
1 .grid {
2   display: grid;
3   grid-template-columns: 100px 100px 100px;
4   grid-template-rows: 200px 200px 200px;
5   grid-gap: 10px;
6 }
```

```
# main.prefixed.css
1 .grid {
2   display: -ms-grid;
3   display: grid;
4   -ms-grid-columns: 100px 100px 100px;
5   grid-template-columns: 100px 100px 100px;
6   -ms-grid-rows: 200px 200px 200px;
7   grid-template-rows: 200px 200px 200px;
8   grid-gap: 10px;
9 }
10
11
12
13
```

Autoprefixer with the grid setting set to true.

```
... # main.prefixed.css • ...  
1 .grid {  
2   display: -ms-grid;  
3   display: grid;  
4  
5   -ms-grid-columns: 100px 100px 100px;  
6   grid-template-columns: 100px 100px 100px;  
7  
8   -ms-grid-rows: 200px 200px 200px;  
9   grid-template-rows: 200px 200px 200px;  
10  
11  grid-gap: 10px;  
12 }  
13
```

```
main.css — darksideofthegrid
# main.css x
1 .grid {
2   display: grid;
3   grid-template: 100px 100px 100px / 200px 200px 200px;
4   grid-gap: 10px;
5 }
6

main.prefixed.css —
# main.prefixed.css ●
1 .grid {
2   display: -ms-grid;
3   display: grid;
4
5   -ms-grid-rows: 100px 10px 100px 10px 100px;
6   -ms-grid-columns: 200px 10px 200px 10px 200px;
7
8   grid-template: 100px 100px 100px / 200px 200px 200px;
9   grid-gap: 10px;
10 }
11
```

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.

...

main.prefixed.css •

✖️ □ ⋮

```
1 .grid {  
2   display: -ms-grid;  
3   display: grid;  
4  
5   -ms-grid-rows: 100px 10px 100px 10px 100px;  
6   -ms-grid-columns: 200px 10px 200px 10px 200px;  
7  
8   grid-template: 100px 100px 100px / 200px 200px 200px;  
9   grid-gap: 10px;  
10 }  
11
```

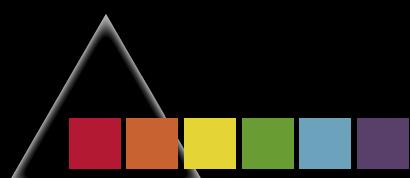
Autoprefixer takes gaps by adding rows and columns in the grid

Autoprefixer gotchas

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

Area names must be unique.

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



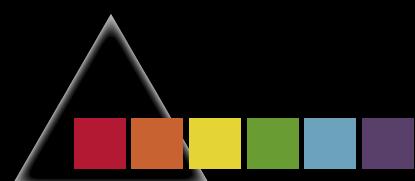
Autoprefixer gotchas

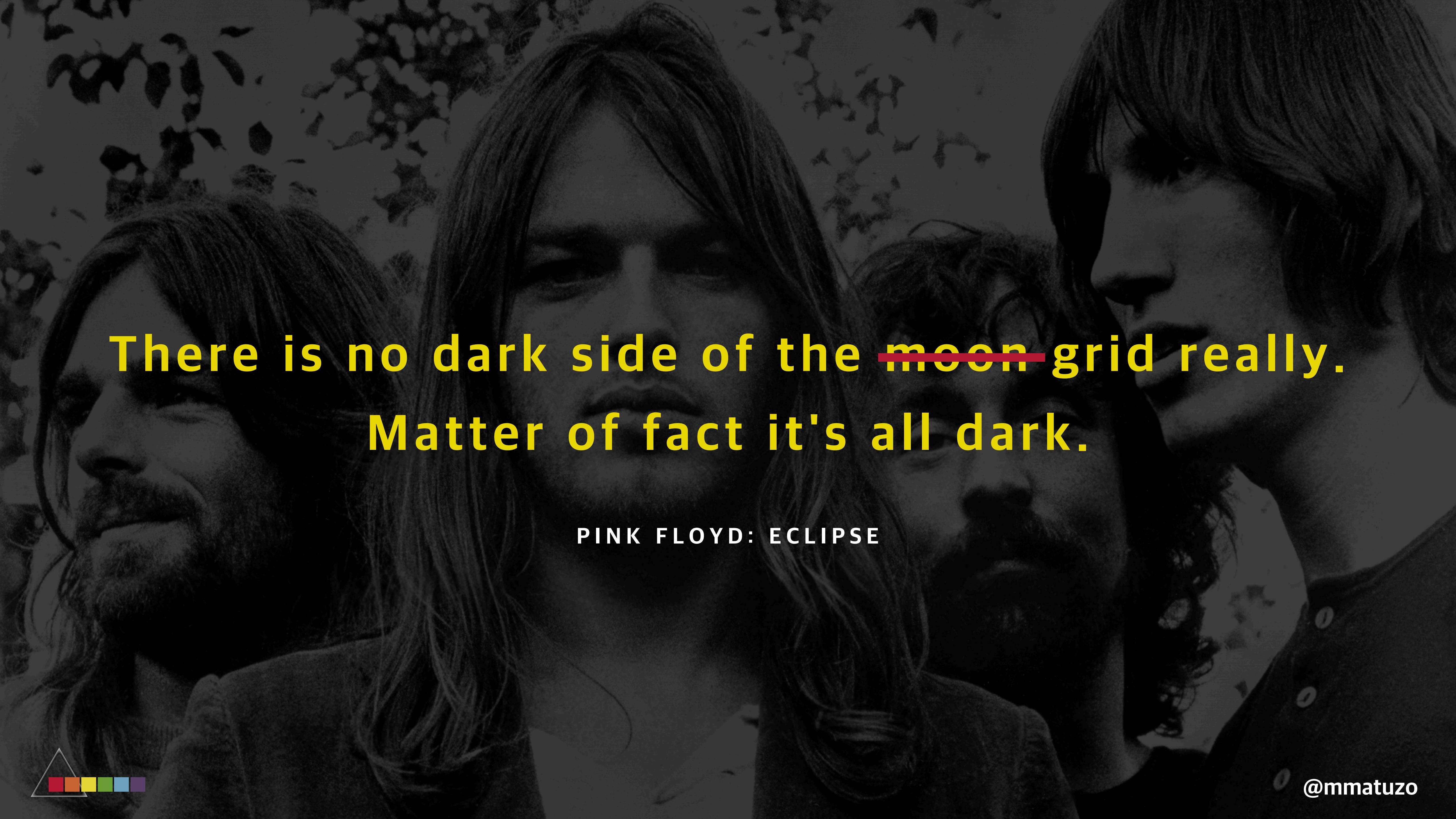
grid shorthand not supported.

There are no named lines at the moment.

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

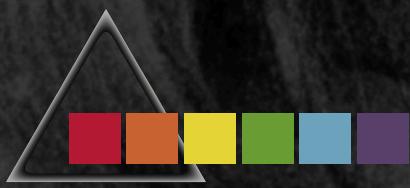
No auto placement.





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

PINK FLOYD: ECLIPSE



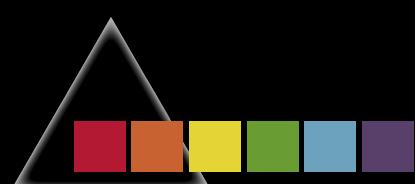
@mmatuzo

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.

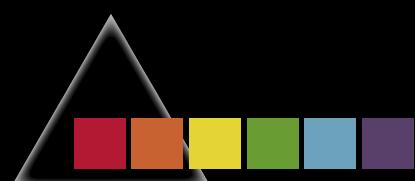


Responsible use of Grid

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.



Thank you ❤

Aaron Gustafson

Adrian Roselli

Heydon Pickering

Léonie Watson

Steve Faulkner

Rob Dodson

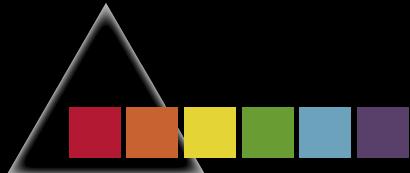
Scott O'Hara

Eric Bailey

Hidde de Vries

Oskar Westin

Billy Gregory



@mmatuzo



manuel@matuzo.at

 @mmatuzo