

Manuel Matuzovic

- Frontend Developer aus Wien (zensions ❤️)



Manuel Matuzovic

- Frontend Developer aus Wien (zensions ❤️)
- Spezialisiert auf CSS, Performance und Accessibility



Manuel Matuzovic

- Frontend Developer aus Wien (zensions ❤️)
- Spezialisiert auf CSS, Performance und Accessibility
- Schreibe Artikel, halte Vorträge und Workshops



Manuel Matuzovic

- Frontend Developer aus Wien (zensions ❤️)
- Spezialisiert auf CSS, Performance und Accessibility
- Schreibe Artikel, halte Vorträge und Workshops
- webclerks und CodePen Meetup in Wien



Manuel Matuzovic

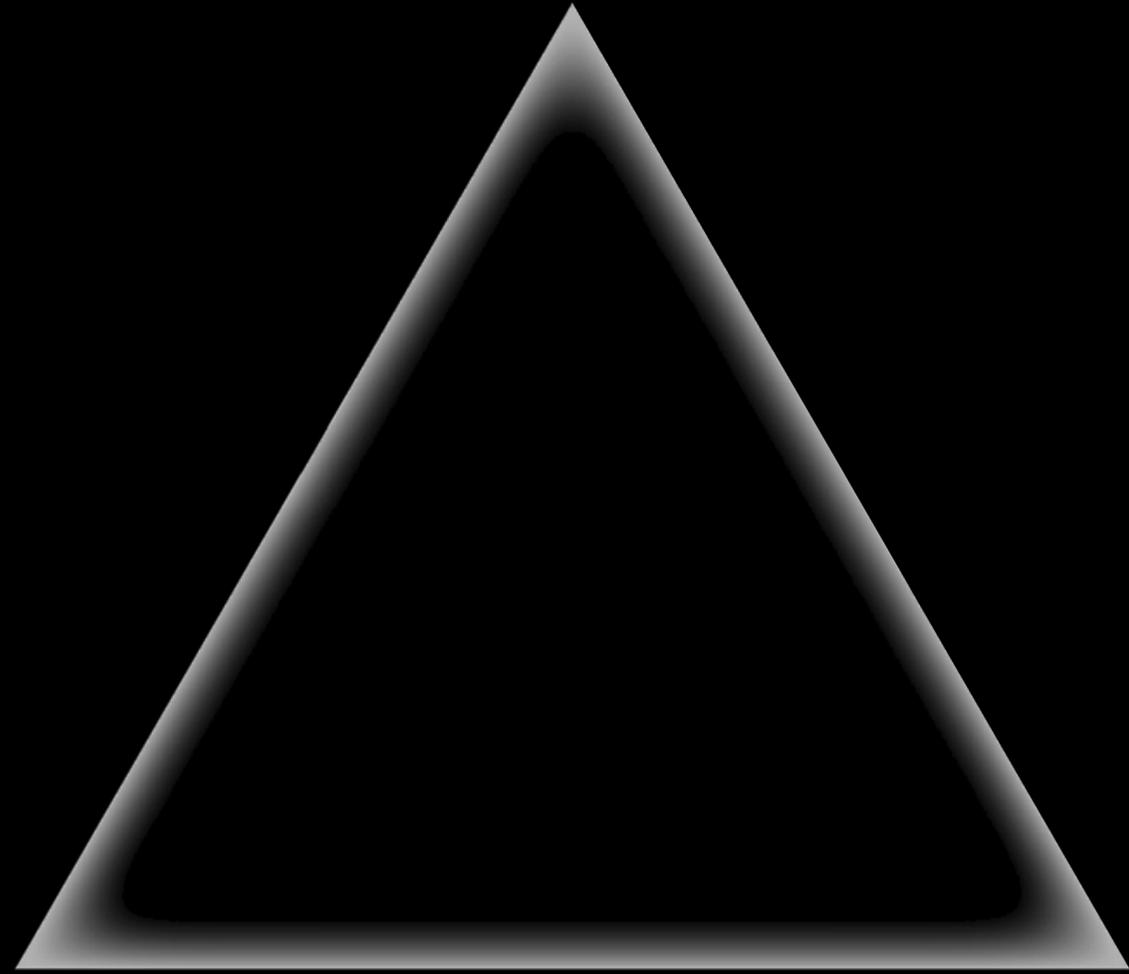
- Frontend Developer aus Wien (zensions ❤️)
- Spezialisiert auf CSS, Performance und Accessibility
- Schreibe Artikel, halte Vorträge und Workshops
- webclerks und CodePen Meetup in Wien
-  @mmatuzo auf Twitter

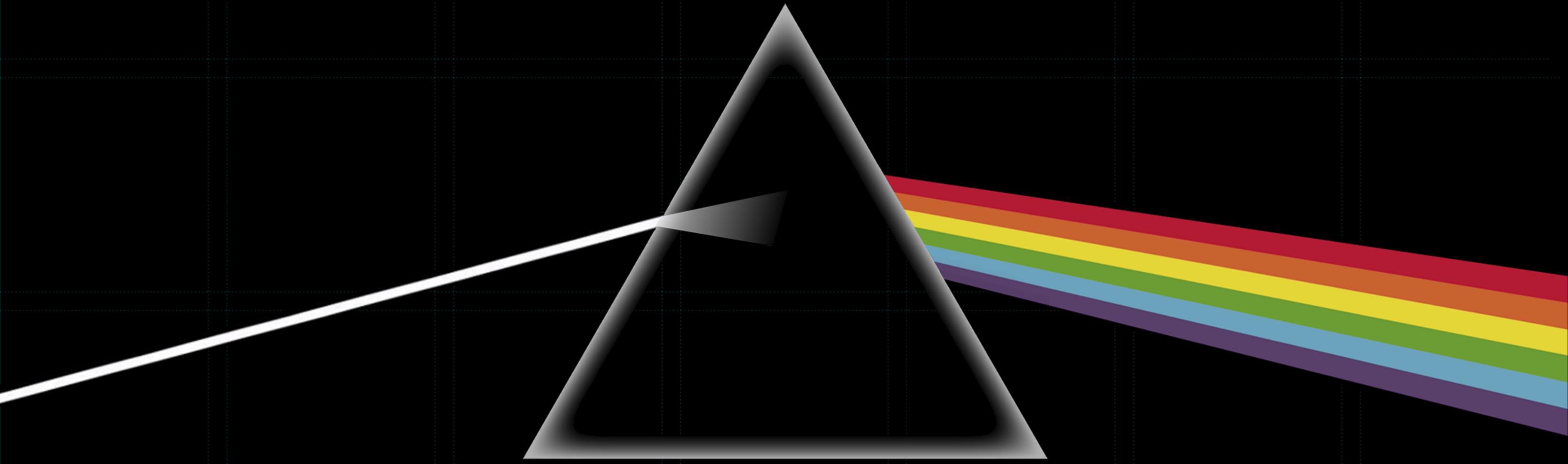




27.09.2018 im stockwerk.co.at
<https://meetup.com/webclerks>







The dark side of the grid

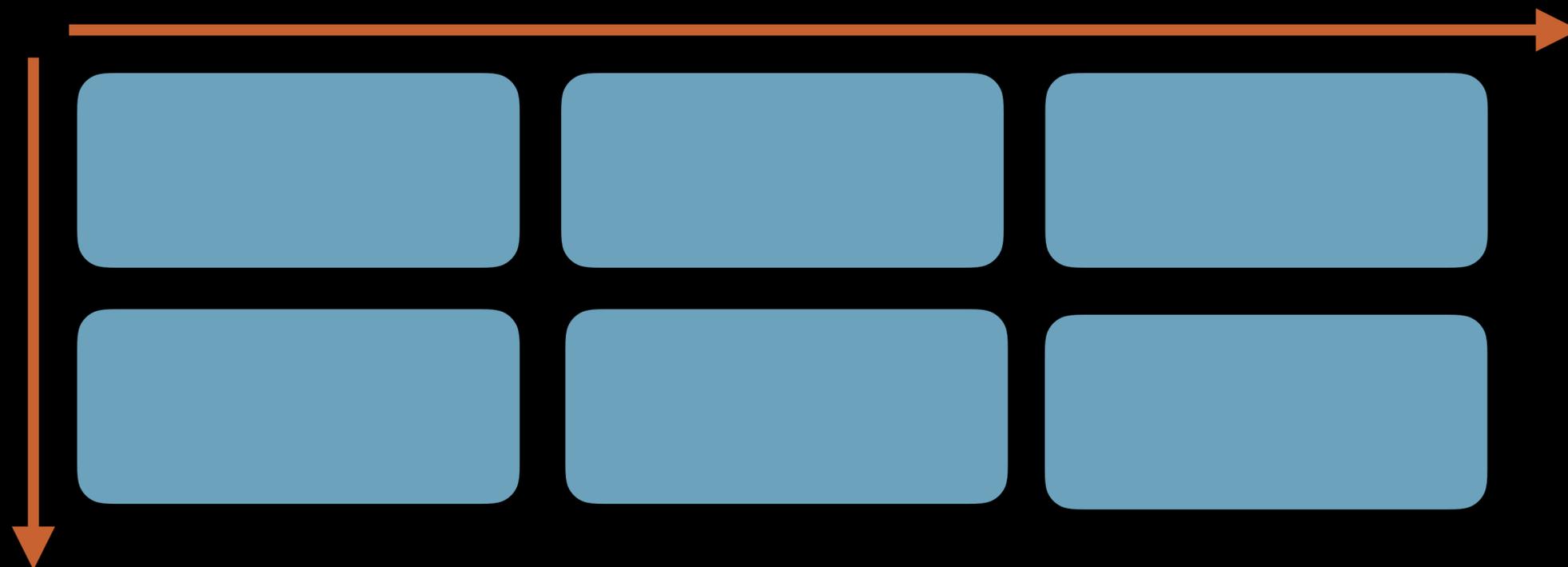
Was ist CSS Grid Layout?

CSS Grid ist ein auf Rastern basierendes Layout-System, das für zweidimensionale Layouts designt wurde.



Was ist CSS Grid Layout?

CSS Grid ist ein auf Rastern basierendes Layout-System, das für zweidimensionale Layouts designt wurde.



Was ist besonders an Grid?



Was ist besonders an Grid?

- Das erste richtige Layout-System in CSS.



Was ist besonders an Grid?

- Das erste richtige Layout-System in CSS.
- **float**, **display: inline-block**, **position**, **display: table** waren ursprünglich nicht für das Bauen von Layouts konzipiert.



Was ist besonders an Grid?

- Das erste richtige Layout-System in CSS.
- **float**, **display: inline-block**, **position**, **display: table** waren ursprünglich nicht für das Bauen von Layouts konzipiert.
- Irgendwie wie table-Layouts, aber responsive, flexibel und in CSS formuliert und nicht in HTML.



Grids Geschichte in Kürze



Grids Geschichte in Kürze

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.



Grids Geschichte in Kürze

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.
- Jänner 2017: CSS Grid in Chromium 56 für Android



Grids Geschichte in Kürze

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.
- Jänner 2017: CSS Grid in Chromium 56 für Android
- Früher März 2017: Chrome und Firefox



Grids Geschichte in Kürze

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.
- Jänner 2017: CSS Grid in Chromium 56 für Android
- Früher März 2017: Chrome und Firefox
- Ende März 2017: Opera und Safari



Grids Geschichte in Kürze

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.
- Jänner 2017: CSS Grid in Chromium 56 für Android
- Früher März 2017: Chrome und Firefox
- Ende März 2017: Opera und Safari
- 17. Oktober 2017: Edge



Grid(-verwandte) Eigenschaften und Werte

- 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



Grid(-verwandte) Eigenschaften und Werte

- 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(-verwandte) Eigenschaften und Werte

- 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 des Albums „The Dark Side of the Moon“, 1973 erschienen.

PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger



PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger

The Dark Side of the Moon ist mit über 45 Millionen verkauften Tonträgern auf Platz 3 der meistverkauften Alben weltweit.



PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger

The Dark Side of the Moon ist mit über 45 Millionen verkauften Tonträgern auf Platz 3 der meistverkauften Alben weltweit.

#1 Thriller von Michael Jackson (66 Millionen)

#2 Back in Black von AC/DC (50 Millionen)





Sven Wolfermann

@maddesigns

Following



@mmatuzo weil wir grad mal wieder nachschlagen wollten wer Platz 2 war... en.wikipedia.org/wiki/List_of_b... Pink Floyd Platz 4 🙌

Translate Tweet

9:09 AM - 20 Sep 2018

2 Likes



1



2



Tweet your reply



Manuel Matuzović @mmatuzo · 7m



Replying to @maddesigns

🙌 Ich halte den Vortrag gleich noch Mal in 5,5 Stunden. Kann ich deinen Tweet in einen Slide packen? 😊

Translate Tweet



PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger



PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger

The Dark Side of the Moon ist mit über 45 Millionen verkauften Tonträgern auf ~~Platz 3~~ Platz 4 der meistverkauften Alben weltweit.



PINK FLOYD FUN FACT #1

45 Millionen verkaufte Tonträger

The Dark Side of the Moon ist mit über 45 Millionen verkauften Tonträgern auf ~~Platz 3~~ Platz 4 der meistverkauften Alben weltweit.

#1 Thriller von Michael Jackson (66 Millionen)

#2 The Greatest Hits (1971-1975) von Eagles (51 Millionen)

#3 Back in Black von AC/DC (50 Millionen)



VISUELLE
REIHENFOLGE



Visuelle Reihenfolge

Sowohl die Tab-Reihenfolge als auch die Reihenfolge in der Screenreader Inhalte lesen folgen der Reihenfolge im DOM.



Wenn die visuelle Reihenfolge nicht der Reihenfolge im DOM entspricht.

1. Keyboard User tun sich schwer, zu erahnen, welches Element als nächstes fokussiert wird.



Wenn die visuelle Reihenfolge nicht der Reihenfolge im DOM entspricht.

2. Benutzer von Bildschirm lupen sind irritiert, wenn der Bildausschnitt ständig hin und her springt.



Wenn die visuelle Reihenfolge nicht der Reihenfolge im DOM entspricht.

3. Wenn ein blinder Screenreader User gemeinsam mit einem sehenden User arbeitet und sie auf Inhalte in unterschiedlicher Reihenfolge stoßen, kann es zu Verwirrungen kommen.



Wer navigiert mit dem Keyboard?



Wer navigiert mit dem Keyboard?

- Menschen, die aus physischen Gründen keine Maus bedienen können.



Wer navigiert mit dem Keyboard?

- Menschen, die aus physischen Gründen keine Maus bedienen können.
- Menschen mit chronischen Erkrankungen, die körperliche Belastung meiden sollten.



Wer navigiert mit dem Keyboard?

- Menschen, die aus physischen Gründen keine Maus bedienen können.
- Menschen mit chronischen Erkrankungen, die körperliche Belastung meiden sollten.
- Jemand, der/die temporär keine Maus verwenden kann (bspw. wegen einer Verletzung)



Wer navigiert mit dem Keyboard?

- Menschen, die aus physischen Gründen keine Maus bedienen können.
- Menschen mit chronischen Erkrankungen, die körperliche Belastung meiden sollten.
- Jemand, der/die temporär keine Maus verwenden kann (bspw. wegen einer Verletzung)
- Poweruser



Wer navigiert mit dem Keyboard?

- Menschen, die aus physischen Gründen keine Maus bedienen können.
- Menschen mit chronischen Erkrankungen, die körperliche Belastung meiden sollten.
- Jemand, der/die temporär keine Maus verwenden kann (bspw. wegen einer Verletzung)
- Poweruser (bspw. DeveloperInnen oder DesignerInnen 🤪)



Wer verwendet Screenreader?



Wer verwendet Screenreader?

- Blinde Menschen.



Wer verwendet Screenreader?

- Blinde Menschen.
- Menschen mit Sehschwächen, die ergänzend einen Screenreader verwenden.



Wer verwendet Screenreader?

- Blinde Menschen.
- Menschen mit Sehschwächen, die ergänzend einen Screenreader verwenden.
- Menschen mit Lernschwächen.



Visuelle Reihenfolge ändern



Visuelle Reihenfolge ändern

- Explizite Platzierung



Visuelle Reihenfolge ändern

- Explizite Platzierung
- Order



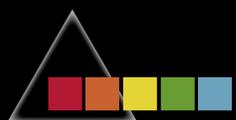
Visuelle Reihenfolge ändern

- Explizite Platzierung
- Order
- Absolute Positionierung



Visuelle Reihenfolge ändern

- Explizite Platzierung
- Order
- Absolute Positionierung
- Auto flow



Visuelle Reihenfolge ändern

- Explizite Platzierung
- Order
- Absolute Positionierung
- Auto flow
- Areas



VISUELLE REIHENFOLGE

AREAS



HEADER

MAIN CONTENT

ADS

NEWSLETTER

COMMENTS

RELATED ARTICLES

FOOTER

Beispiel: Mobiles Layout.

```
$ areas
```

```
body {
```

```
  display: grid;
```

```
  grid-template-columns: minmax(auto, 480px);
```

```
}
```



```
$ areas
```

```
body {
```

```
  display: grid;
```

```
  grid-template-columns: minmax(auto, 480px);
```

```
  grid-template-areas:
```

```
}
```



```
$ areas
```

```
body {  
  display: grid;  
  grid-template-columns: minmax(auto, 480px);  
  grid-template-areas:  
    "header"
```

```
}
```



```
$ areas
```

```
body {  
  display: grid;  
  grid-template-columns: minmax(auto, 480px);  
  grid-template-areas:  
    "header"  
    "content"
```

```
}
```



```
$ areas
```

```
body {  
  display: grid;  
  grid-template-columns: minmax(auto, 480px);  
  grid-template-areas:  
    "header"  
    "content"  
    "ads"  
  
}
```



```
$ areas
```

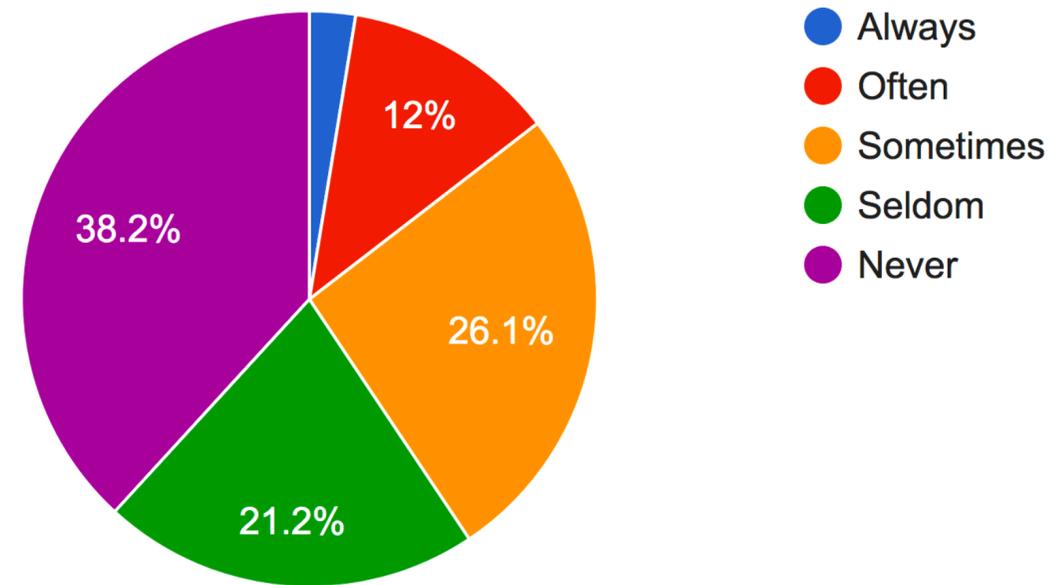
```
body {  
  display: grid;  
  grid-template-columns: minmax(auto, 480px);  
  grid-template-areas:  
    "header"  
    "content"  
    "ads"  
    "newsletter"  
    "comments"  
    "related"  
    "footer";  
}
```



```
$ areas
```

```
body {  
  display: grid;  
  grid-template-columns: minmax(auto, 480px);  
  grid-template-areas:  
    "header"  
    "newsletter"  
    "content"  
    "ads"  
    "comments"  
    "related"  
    "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% der befragten Screenreaderuser verwenden immer oder oft ein Keyboard auf ihrem mobilen Endgerät.

```
$ areas
```

```
body {
```

```
  grid-template-columns: 1fr 400px;
```

```
}
```



```
$ areas
```

```
body {  
  grid-template-columns: 1fr 400px;  
  grid-template-areas:
```

```
}
```



```
$ areas
```

```
body {  
  grid-template-columns: 1fr 400px;  
  grid-template-areas:  
    "header header"
```

```
}
```



```
$ areas
```

```
body {  
  grid-template-columns: 1fr 400px;  
  grid-template-areas:  
    "header header"  
    "newsletter newsletter"  
  
}
```



```
$ areas
```

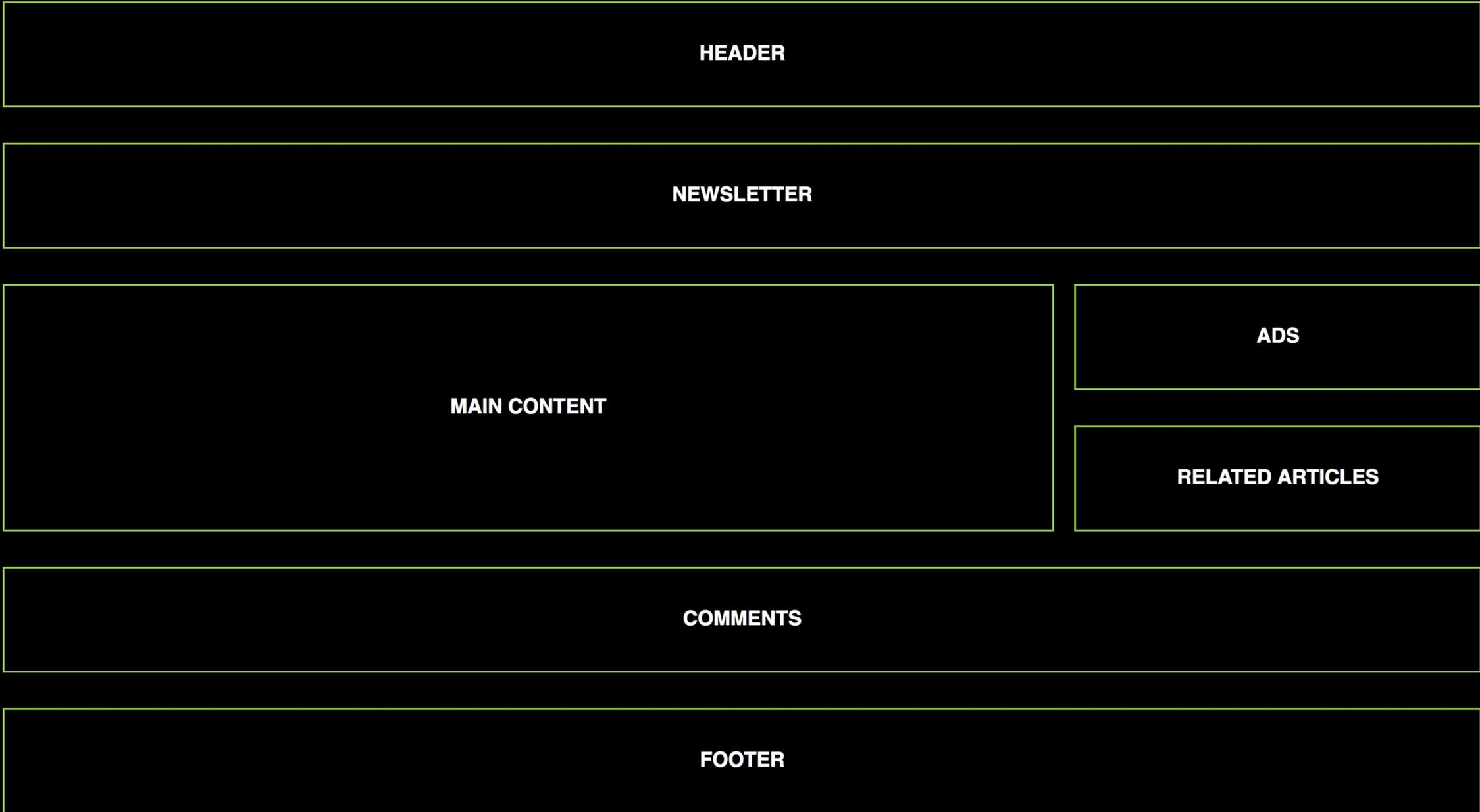
```
body {  
  grid-template-columns: 1fr 400px;  
  grid-template-areas:  
    "header header"  
    "newsletter newsletter"  
    "content partners"  
}
```



```
$ areas
```

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

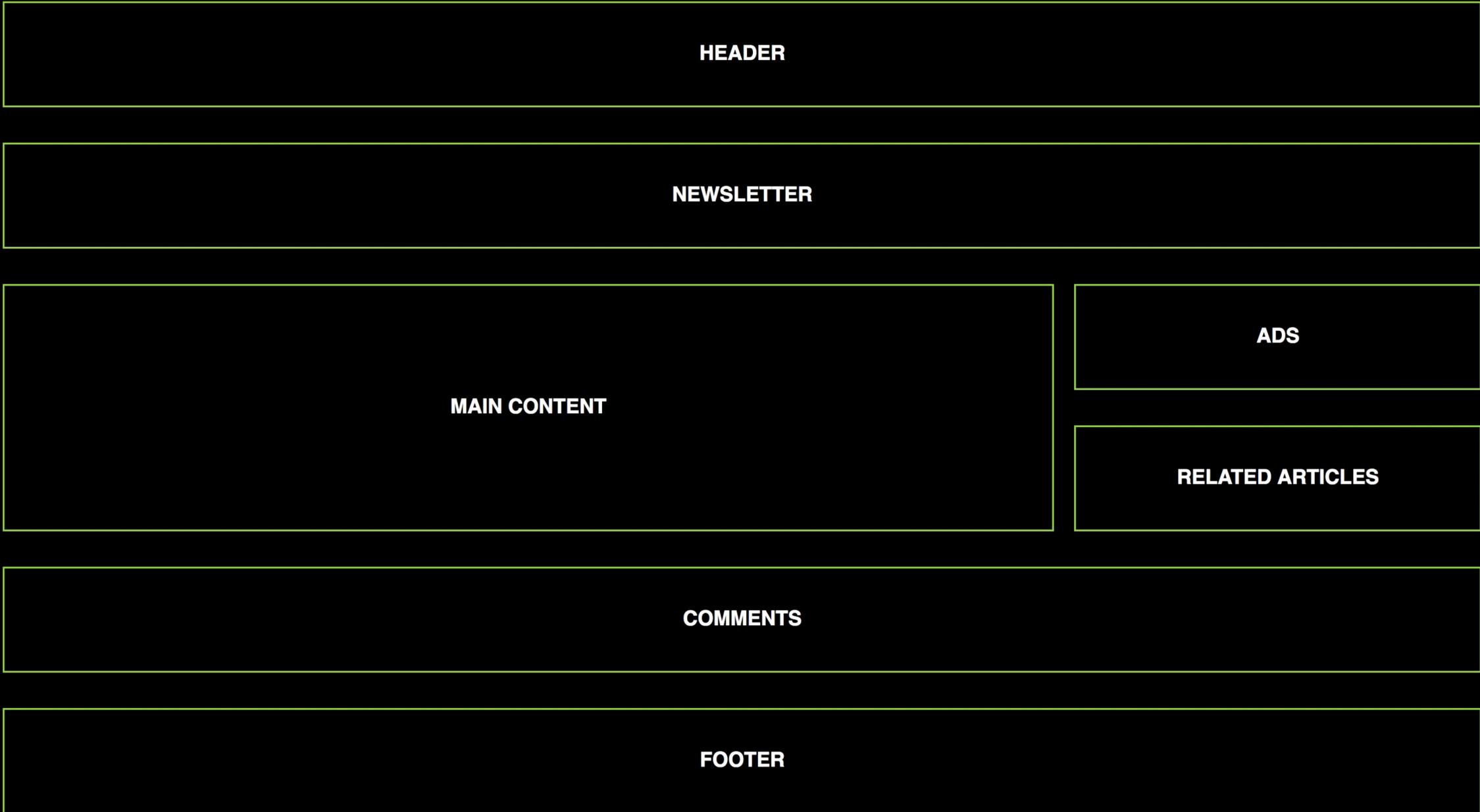




<https://codepen.io/matuzo/pen/QVOXdR>



@mmatuzo



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



@mmatuzo

**B E I W E M L I E G T D I E
V E R A N T W O R T U N G ?**



Unsere Optionen



Unsere Optionen

- Source Order dynamisch bei jedem Breakpoint mit JS anpassen. 🤔



Unsere Optionen

- Source Order dynamisch bei jedem Breakpoint mit JS anpassen. 🙄
- **tabindex** oder **aria-flow-to** allen Elementen zuweisen und entsprechend ordnen. 😭



Unsere Optionen

- Source Order dynamisch bei jedem Breakpoint mit JS anpassen. 🙄
- **tabindex** oder **aria-flow-to** allen Elementen zuweisen und entsprechend ordnen. 😭
- Browsersniffing – Je nach Client/Gerätetyp unterschiedliches HTML ausliefern. 🙄



Unsere Optionen



Unsere Optionen

- Das Problem mindern indem man Skiplinks zur Verfügung stellt. 😐



Unsere Optionen

- Das Problem mindern indem man Skiplinks zur Verfügung stellt. 😐
- Die visuelle Reihenfolge nicht ändern. 🙄



„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>

Was machen wir also?



Was machen wir also?

- Über die Reihenfolge Gedanken machen noch bevor HTML oder CSS geschrieben wird. So früh wie möglich eng mit den DesignerInnen zusammenarbeiten.



Was machen wir also?

- Über die Reihenfolge Gedanken machen noch bevor HTML oder CSS geschrieben wird. So früh wie möglich eng mit den DesignerInnen zusammenarbeiten.
- Mit sauber strukturiertem HTML beginnen.



Was machen wir also?

- Über die Reihenfolge Gedanken machen noch bevor HTML oder CSS geschrieben wird. So früh wie möglich eng mit den DesignerInnen zusammenarbeiten.
- Mit sauber strukturiertem HTML beginnen.
- CSS mobile first schreiben und entsprechend das Layout anpassen.



Was machen wir also?



Was machen wir also?

- Das Layout testen indem man mit der Tab-Taste auf unterschiedlichen Bildschirmgrößen durch die Seiten navigiert.



Was machen wir also?

- Das Layout testen indem man mit der Tab-Taste auf unterschiedlichen Bildschirmgrößen durch die Seiten navigiert.
- Wenn es Diskrepanzen in der Reihenfolge gibt, zurück zum Source Code und das HTML anpassen.





PINK FLOYD FUN FACT #2

736 Wochen in den Charts



@mmatuzo



PINK FLOYD FUN FACT #2

736 Wochen in den Charts

The Dark Side of The Moon war 736 Wochen in den amerikanischen Billboard Charts. Das sind mehr als 14 Jahre!



D O K U M E N T S T R U K T U R

A B F L A C H E N



„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

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

\$ dokumentstruktur abflachen

```
<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>  
  ...
```



\$ dokumentstruktur abflachen

...

```
<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

\$ dokumentstruktur abflachen

```
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

\$ dokumentstruktur abflachen

```
<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>
```



\$ dokumentstruktur abflachen

~~<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>~~



\$ dokumentstruktur abflachen

```
<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>
```



\$ dokumentstruktur abflachen

```
<fieldset>  
  <del>legend strongShirt size</del 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?	<input type="text"/>
E-Mail	<input type="text"/>
Shirt size	<input type="radio"/> S <input type="radio"/> M

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



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

- S
- M

D O K U M E N T S T R U K T U R A B F L A C H E N

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/>

```
$ dokumentstruktur abflachen
```

```
div {  
    display: grid;  
  
}
```



```
$ dokumentstruktur abflachen
```

```
div {  
  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 kommt erst mit Level 2 der Spezifikation

„**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>

```
$ dokumentstruktur abflachen
```

```
form > div,  
fieldset {  
  
}
```



```
$ dokumentstruktur abflachen
```

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



What's your name?	<input type="text"/>
E-Mail	<input type="text"/>
Shirt size	<input type="radio"/> S <input type="radio"/> M

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



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.

Current aligned

Usage relative

Date relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49							
			¹ 63							
			² 67			10.3				
		² 61	² 68		² 54	11.2				4
11	17	62	² 69	² 11.1	² 55	² 11.4	all	² 67	11.8	7.2
	18	63	² 70	² 12		² 12				
		64	² 71	² TP						
			² 72							

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; wird in allen großen Browsern außer Edge unterstützt



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



Wegen eines Bugs entfernt `display: contents;` Elemente aus dem Accessibility-Tree.

\$ dokumentstruktur abflachen

display: block

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



\$ dokumentstruktur abflachen

display: block

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

```
.contents {  
    display: contents;  
}
```



\$ dokumentstruktur abflachen

display: block

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

```
.contents {  
  display: contents;  
}
```

display: contents

Element 1
Element 2
Element 3
Element 4



PINK FLOYD FUN FACT #3

Monty Python and the Holy Grail



PINK FLOYD FUN FACT #3

Monty Python and the Holy Grail

Pink Floyd haben 1975 dabei geholfen, den Monty Python Film „Die Ritter der Kokosnuß“ zu finanzieren.

Andere Investoren waren unter anderem Led Zeppelin und Genesis.



**CROSS BROWSER
SUPPORT**



Kurze Wiederholung der Geschichte



Kurze Wiederholung der Geschichte

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.



Kurze Wiederholung der Geschichte

- 2011: Microsoft liefert die erste Implementierung von Grid hinter dem **-ms-** Browserprefix in IE 10.
- Chrome, Firefox, Opera, Safari und Edge implementieren Grid im Laufe des Jahres 2017.



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns		grid-row	
grid-template-rows		grid-column	
grid-template-areas		grid-area	
grid-template		grid-row-gap	
grid-auto-columns		grid-column-gap	
grid-auto-rows		grid-auto-flow	
grid-auto-flow		grid-gap	
grid		-	
grid-row-start		-	
grid-column-start		align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows		grid-column	
grid-template-areas		grid-area	
grid-template		grid-row-gap	
grid-auto-columns		grid-column-gap	
grid-auto-rows		grid-auto-flow	
grid-auto-flow		grid-gap	
grid		-	
grid-row-start		-	
grid-column-start		align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows	-ms-grid-rows	grid-column	
grid-template-areas		grid-area	
grid-template		grid-row-gap	
grid-auto-columns		grid-column-gap	
grid-auto-rows		grid-auto-flow	
grid-auto-flow		grid-gap	
grid		-	
grid-row-start		-	
grid-column-start		align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows	-ms-grid-rows	grid-column	
grid-template-areas	X	grid-area	
grid-template		grid-row-gap	
grid-auto-columns		grid-column-gap	
grid-auto-rows		grid-auto-flow	
grid-auto-flow		grid-gap	
grid		-	
grid-row-start		-	
grid-column-start		align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows	-ms-grid-rows	grid-column	
grid-template-areas	X	grid-area	
grid-template	X	grid-row-gap	
grid-auto-columns	X	grid-column-gap	
grid-auto-rows	X	grid-auto-flow	
grid-auto-flow	X	grid-gap	
grid	X	-	
grid-row-start		-	
grid-column-start		align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows	-ms-grid-rows	grid-column	
grid-template-areas	X	grid-area	
grid-template	X	grid-row-gap	
grid-auto-columns	X	grid-column-gap	
grid-auto-rows	X	grid-auto-flow	
grid-auto-flow	X	grid-gap	
grid	X	-	
grid-row-start	-ms-grid-row	-	
grid-column-start	-ms-grid-column	align-self	
grid-row-end		justify-self	
grid-column-end			



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	
grid-template-rows	-ms-grid-rows	grid-column	
grid-template-areas	X	grid-area	
grid-template	X	grid-row-gap	
grid-auto-columns	X	grid-column-gap	
grid-auto-rows	X	grid-auto-flow	
grid-auto-flow	X	grid-gap	
grid	X	-	
grid-row-start	-ms-grid-row	-	
grid-column-start	-ms-grid-column	align-self	
grid-row-end	X	justify-self	
grid-column-end	X		



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	X
grid-template-rows	-ms-grid-rows	grid-column	X
grid-template-areas	X	grid-area	X
grid-template	X	grid-row-gap	X
grid-auto-columns	X	grid-column-gap	X
grid-auto-rows	X	grid-auto-flow	X
grid-auto-flow	X	grid-gap	X
grid	X	-	
grid-row-start	-ms-grid-row	-	
grid-column-start	-ms-grid-column	align-self	
grid-row-end	X	justify-self	
grid-column-end	X		



Eigenschaften: Level 1 vs. MS Spezifikation

CR Level 1 Eigenschaft	IE10 Implementation	CR Level 1 Eigenschaft	IE10 Implementation
grid-template-columns	-ms-grid-columns	grid-row	X
grid-template-rows	-ms-grid-rows	grid-column	X
grid-template-areas	X	grid-area	X
grid-template	X	grid-row-gap	X
grid-auto-columns	X	grid-column-gap	X
grid-auto-rows	X	grid-auto-flow	X
grid-auto-flow	X	grid-gap	X
grid	X	-	-ms-grid-column-span
grid-row-start	-ms-grid-row	-	-ms-grid-row-span
grid-column-start	-ms-grid-column	align-self	-ms-grid-column-align
grid-row-end	X	justify-self	-ms-grid-row-align
grid-column-end	X		



```
$ browser support
```

```
.grid {
```

```
  display: grid;
```

```
  grid-template-columns: 200px 200px;
```

```
  grid-template-rows: 130px 130px;
```

```
}
```



```
$ browser support
```

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

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



```
$ browser support
```

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

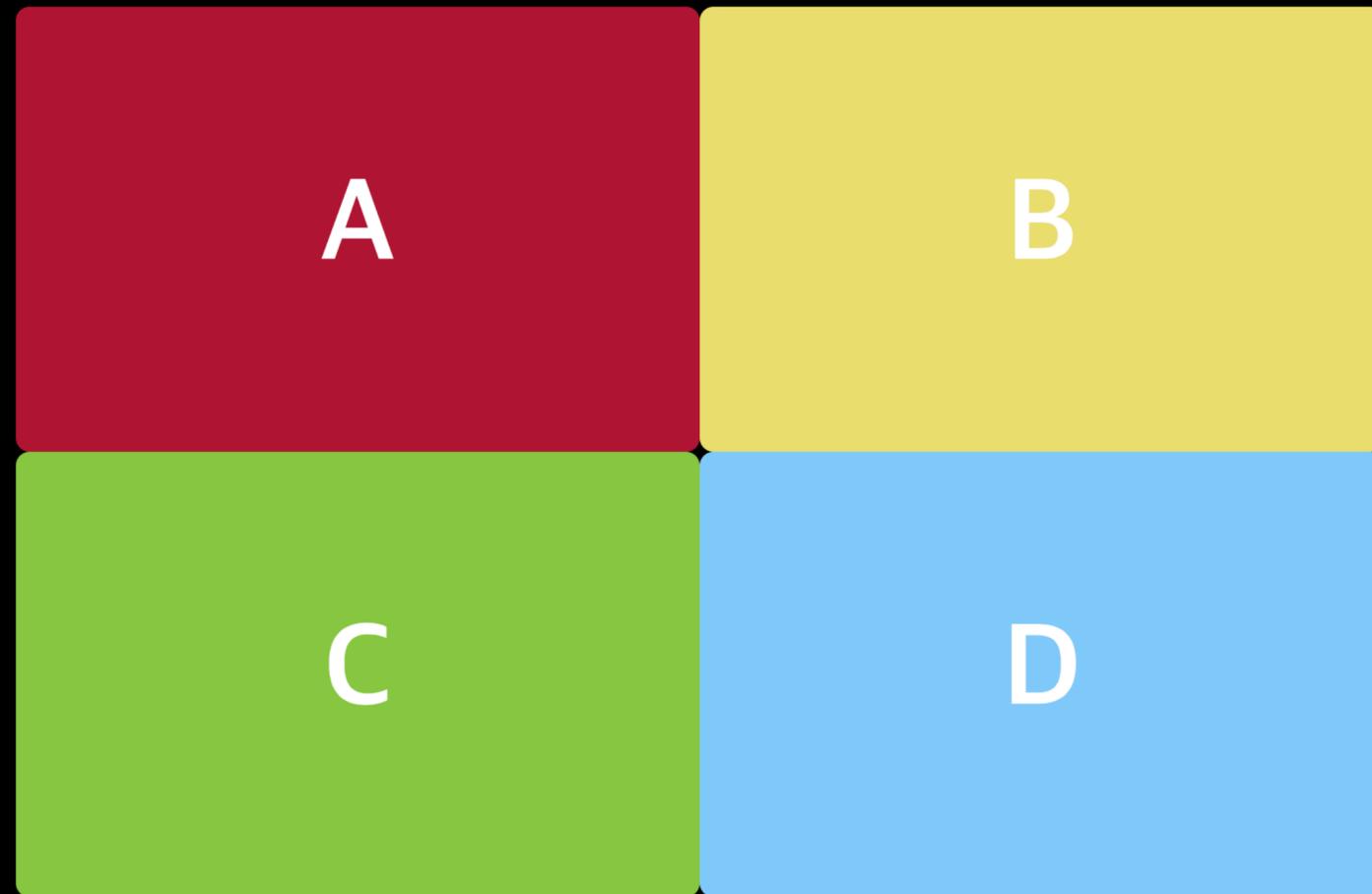


```
$ browser support
```

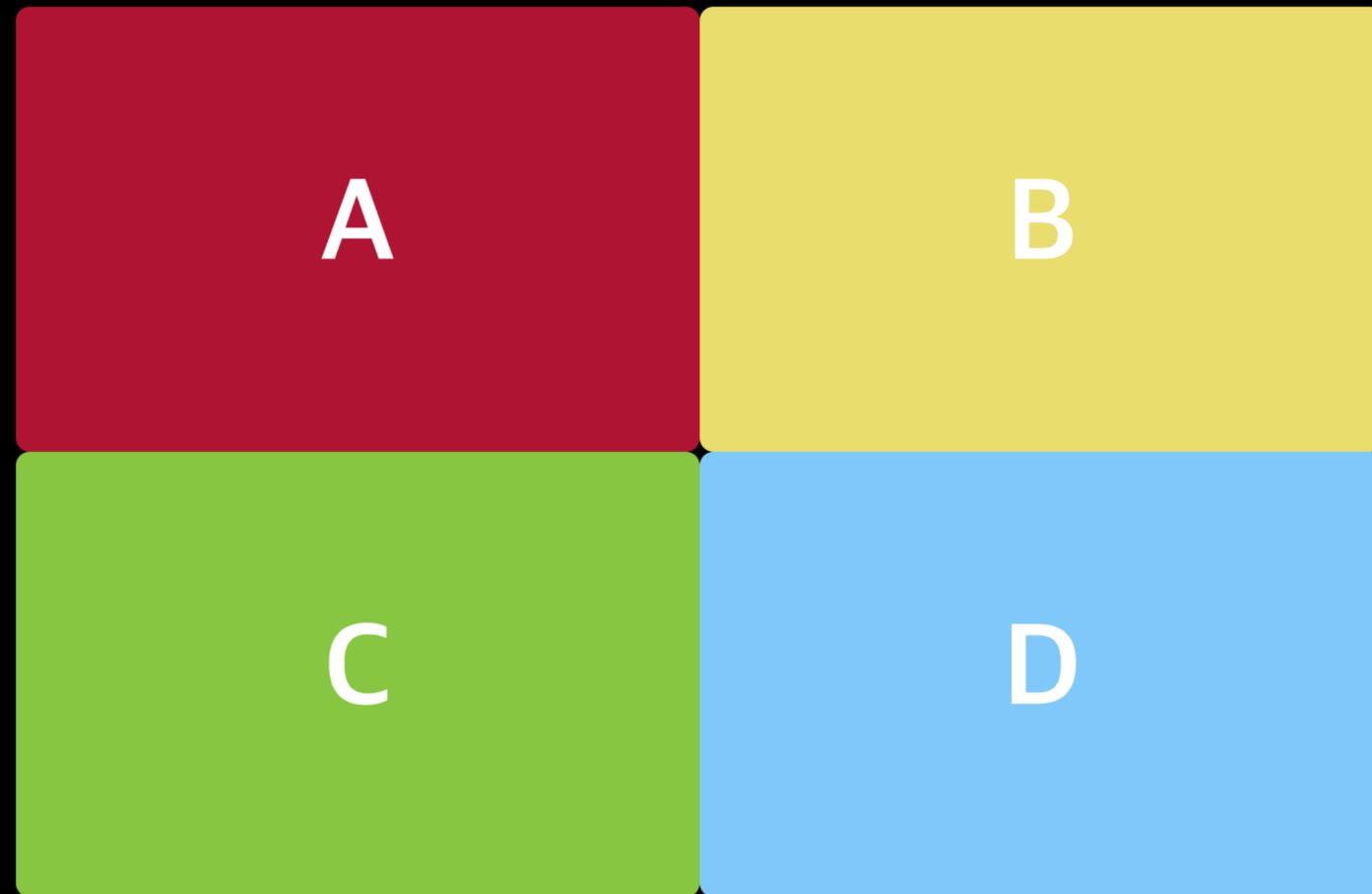
```
.grid {  
  display: -ms-grid;  
  -ms-grid-columns: 200px 200px;  
  -ms-grid-rows: 130px 130px;  
  
  display: grid;  
  grid-template-columns: 200px 200px;  
  grid-template-rows: 130px 130px;  
}
```



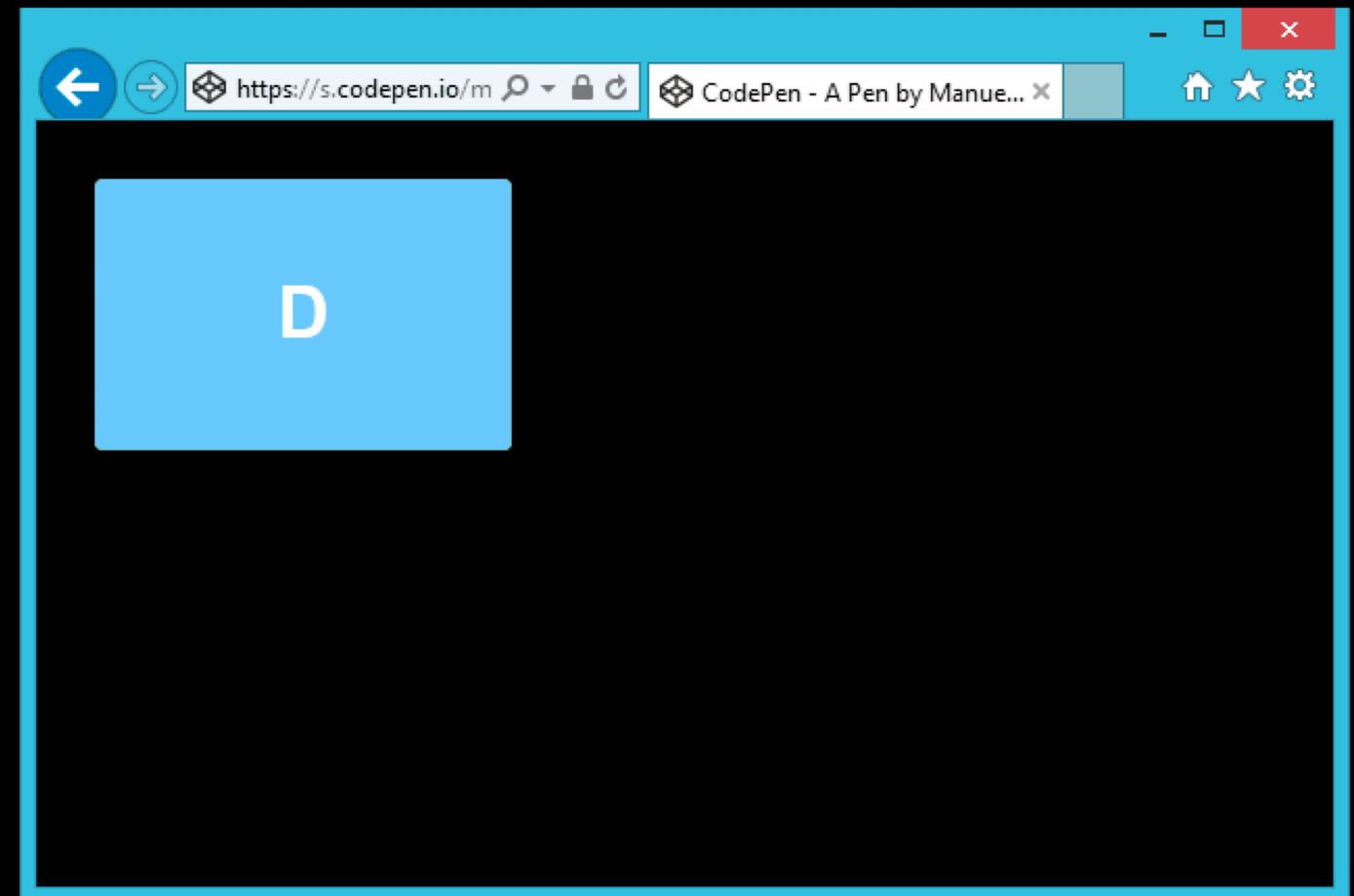
Chrome



Chrome



IE 11



4 Elemente in einem 2 x 2 Grid ohne explizite Platzierung: Darstellung in Chrome und IE11

```
$ browser support
```

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



```
$ 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;  
}
```



```
$ browser support
```

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

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

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



\$ browser support

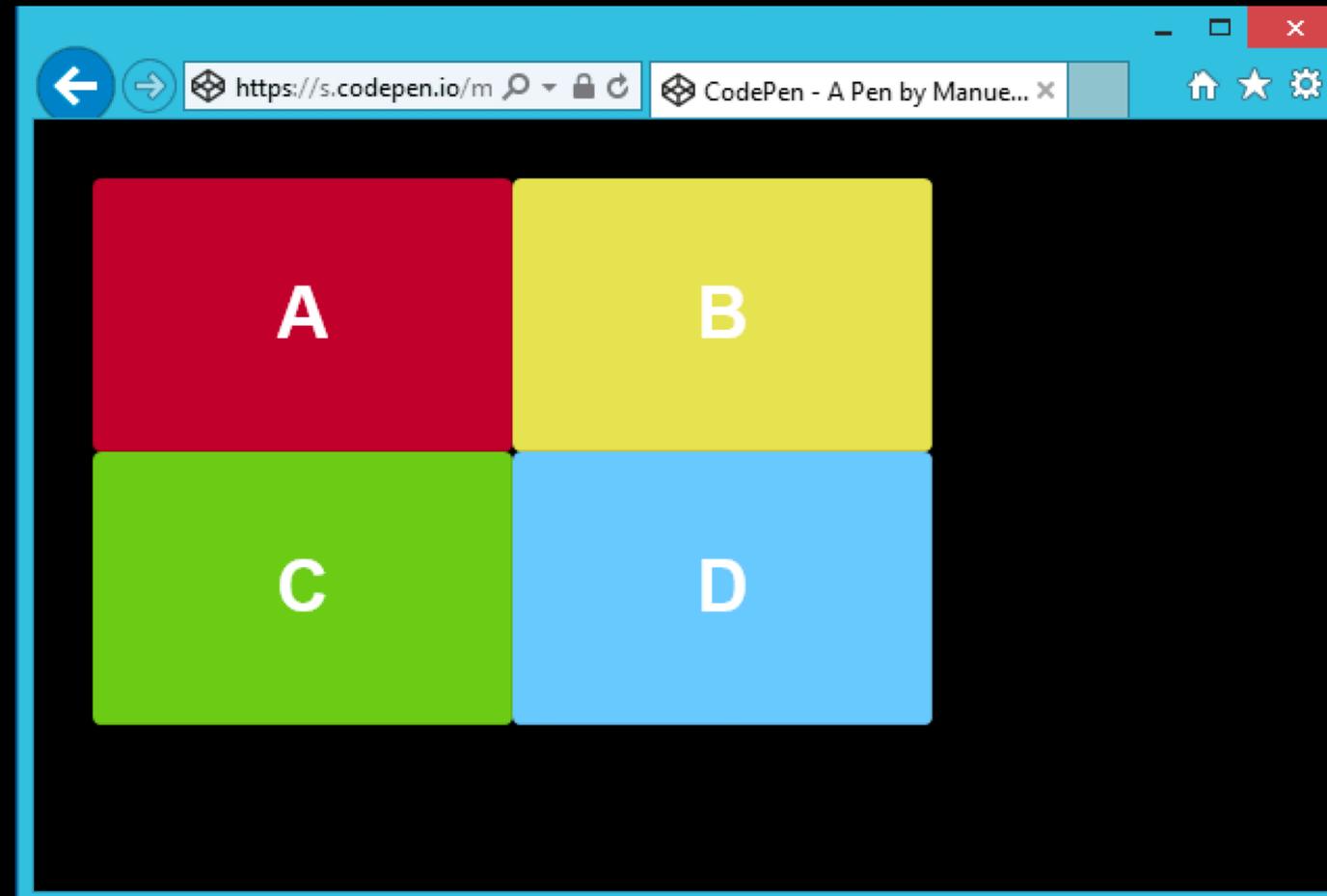
```
.item:nth-child(1) {  
  -ms-grid-column: 1;  
  -ms-grid-row: 1;  
}
```

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

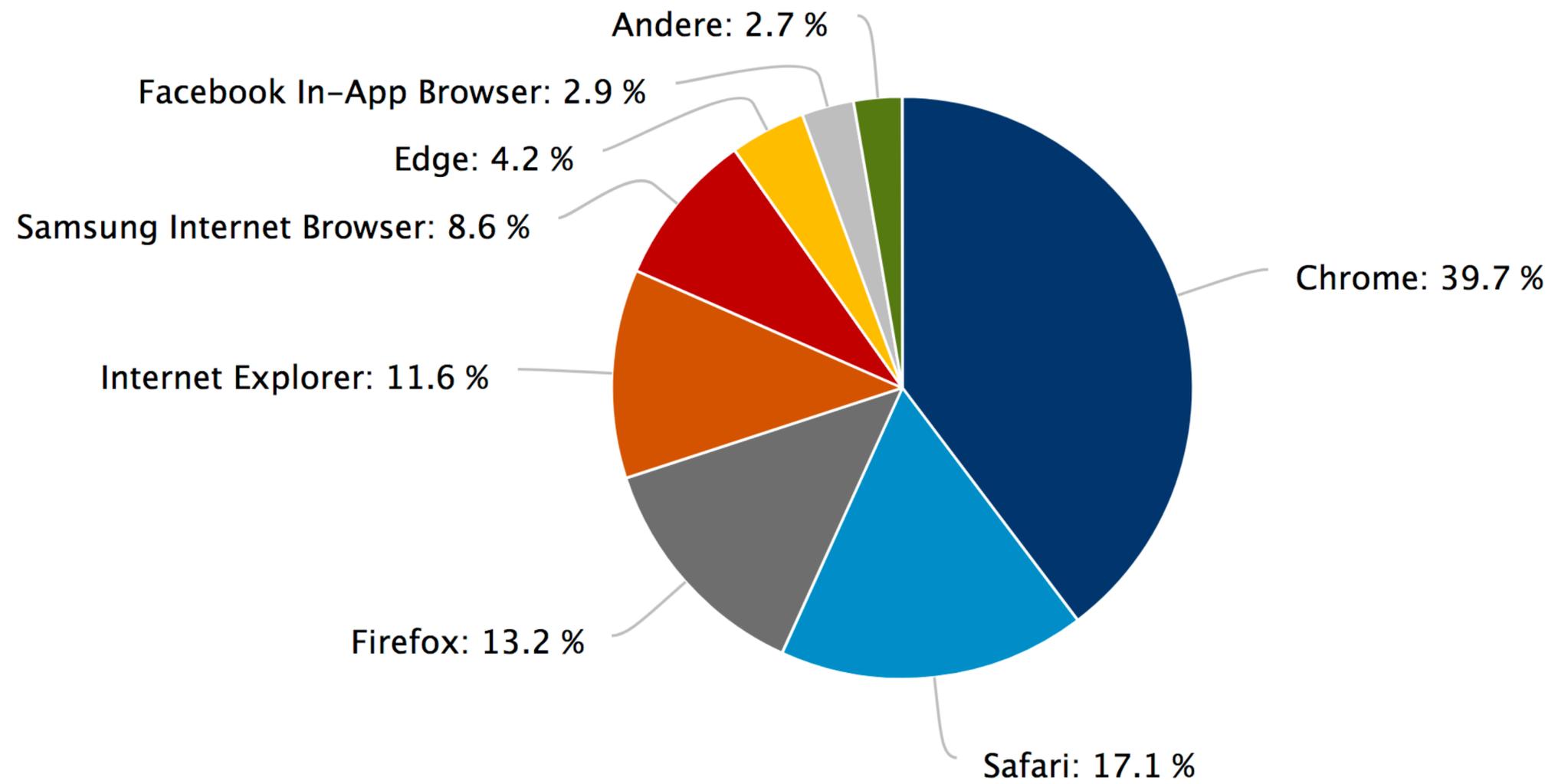
```
.item:nth-child(3) {  
  -ms-grid-column: 1;  
  -ms-grid-row: 2;  
}
```

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





4 Elemente in einem 2 x 2 Grid mit expliziter Platzierung: Darstellung in IE11



11,6% IE Nutzer in Österreich.



Benjamin De Cock

@bdc

Following



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

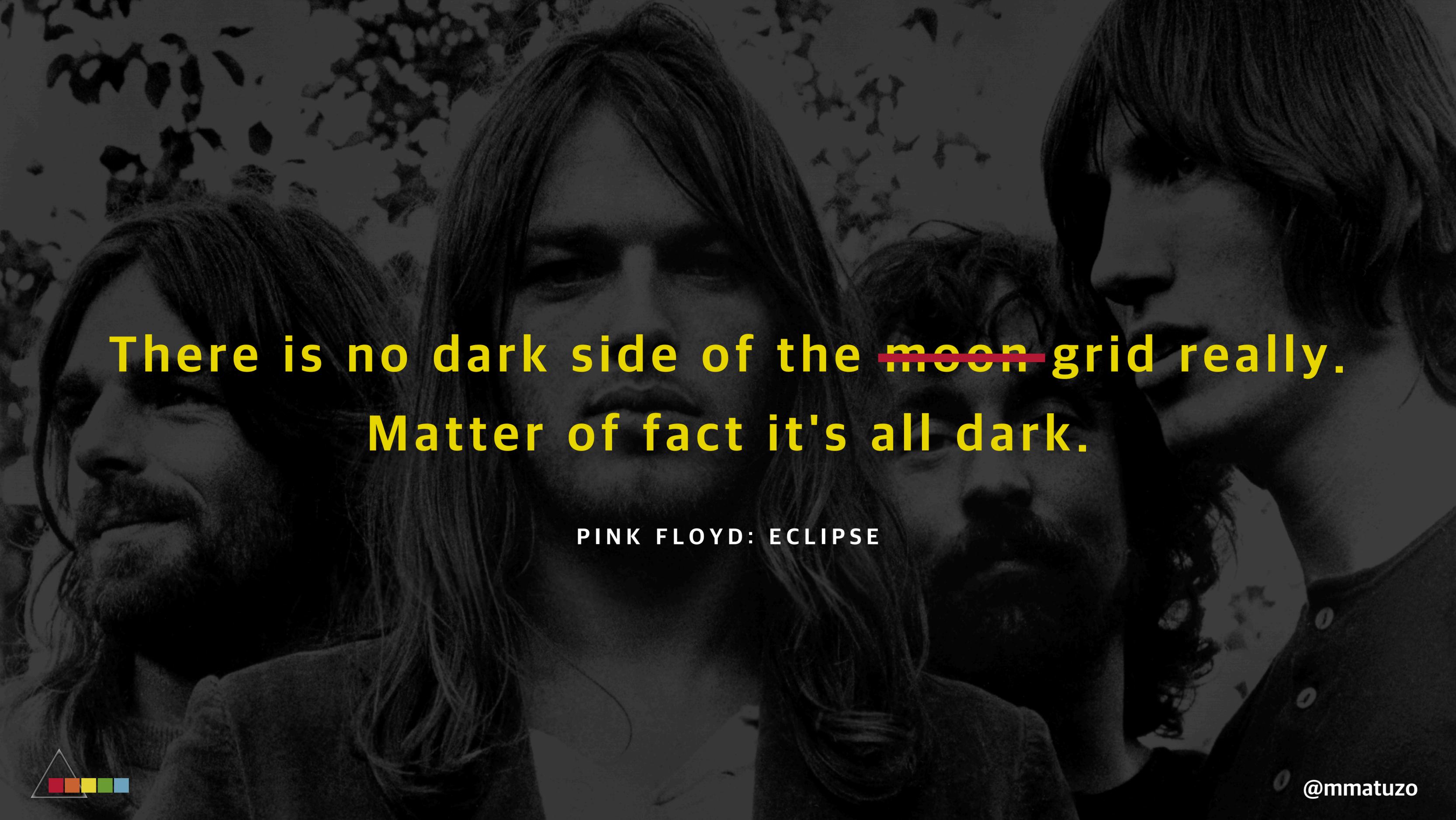
11:49 PM - 5 Jul 2018

Internet Explorer Unterstützung stellt für viele EntwicklerInnen noch eine große Hürde da.

„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

<https://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



@mmatuzo

Verantwortlicher Umgang mit Grid



Verantwortlicher Umgang mit Grid

- Eigenschaften wie **grid-template-areas**; oder andere Techniken, die die visuelle Reihenfolge verändern, bei interaktionsintensiven Komponenten vermeiden.



Verantwortlicher Umgang mit Grid

- Eigenschaften wie **grid-template-areas**; oder andere Techniken, die die visuelle Reihenfolge verändern, bei interaktionsintensiven Komponenten vermeiden.
- Komponenten mit dem Keyboard testen.



Verantwortlicher Umgang mit Grid

- Eigenschaften wie **grid-template-areas**; oder andere Techniken, die die visuelle Reihenfolge verändern, bei interaktionsintensiven Komponenten vermeiden.
- Komponenten mit dem Keyboard testen.
- Nicht die Semantik verschlechtern und die Dokumentstruktur abflachen.



Verantwortlicher Umgang mit Grid



Verantwortlicher Umgang mit Grid

- Wenn man Autoprefixer verwendet, dann so, dass zu jeder Zeit für jede/n im Team klar ist, was warum passiert.



Verantwortlicher Umgang mit Grid

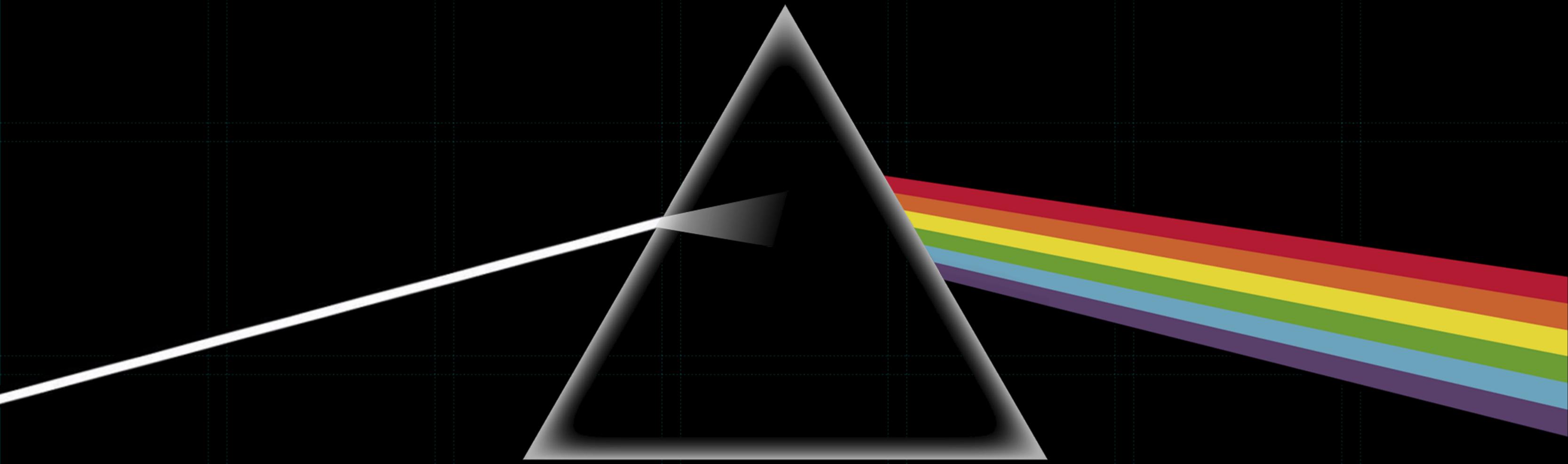
- Wenn man Autoprefixer verwendet, dann so, dass zu jeder Zeit für jede/n im Team klar ist, was warum passiert.
- Sicher gehen, dass die Komponenten auch in älteren / eingeschränkteren Browsern funktionieren.



Verantwortlicher Umgang mit Grid

- Wenn man Autoprefixer verwendet, dann so, dass zu jeder Zeit für jede/n im Team klar ist, was warum passiert.
- Sicher gehen, dass die Komponenten auch in älteren / eingeschränkteren Browsern funktionieren.
- Die Macht von Progressive Enhancement ausnutzen.





Danke

manuel@matuzo.at

 @mmatuzo