6 CSS GRID

5 CSS GRID

CSS grid of 'the css grid layout module' is een uitbereiding op bestaande positioning methoden.

Het is bedoeld om complexe pagina layouts op te maken in een 2-dimentionaal vlak, zonder extra markup.



❺ CSS GRID VRAGEN:

*1) Wat is **grid** en wat is het verschil met **Flexbox**?*

*2) Hoe kan ik het in **productie** gebruiken?*

*3) Wat met **oude browsers**?*

*4) Kunnen we oude browsers helpen? (**Polyfills**?)*



CYNICAL BEHIND YOU:



Flexbox & CSS grid zijn naast elkaar ontwikkelt door dezelfde groep (W3C); ze vullen elkaar aan,

HET ZIJN GEEN CONCURRENTEN.

De vragen die je moet stellen;
1) Waarvoor gebruik ik Flexbox?
2) Waarvoor dient CSS Grid?

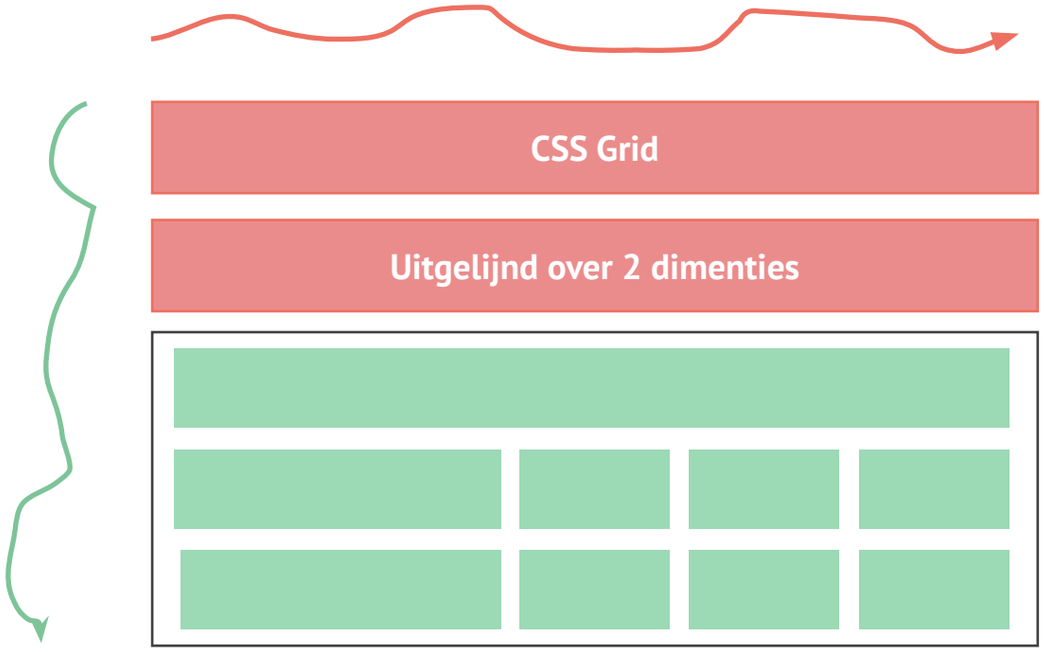
WHICH IS BETTER? – FLEXBOX VS. CSS GRID: HTTPS://WWW.YOUTUBE.COM/WATCH?V=0XRSO4AIJYC

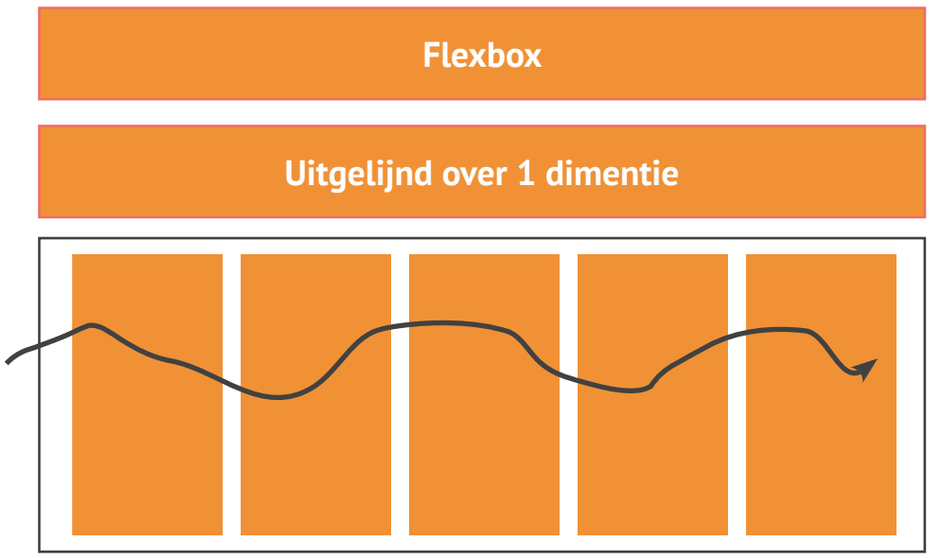
FLEXBOX VS GRID



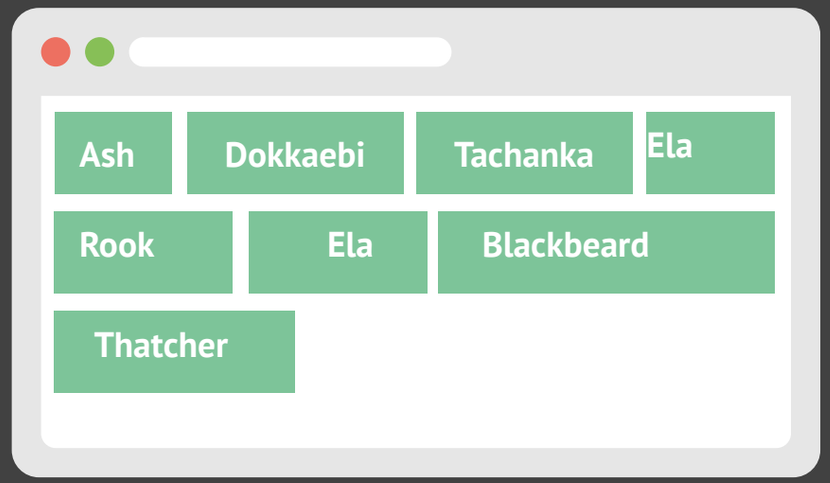
IF YOU ARE ADDING WIDTHS TO ALL YOUR (FLEX) ITEMS, YOU PROBABLY NEED GRID

Als je items wil uitlijnen in een container: flexbox,
Als je een pagina wil verdelen in grote secties met een verschillende vorm: CSS Grid.





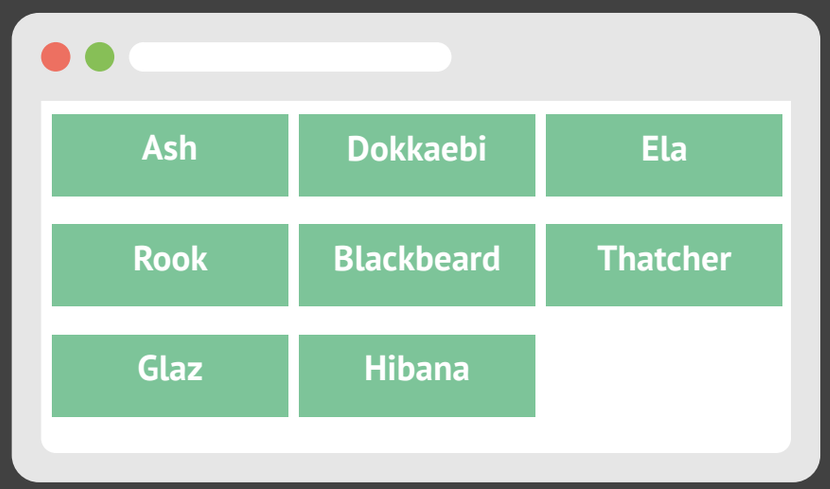
FLEXBOX VS GRID



FLEX vertrekt vanuit de styling van de items

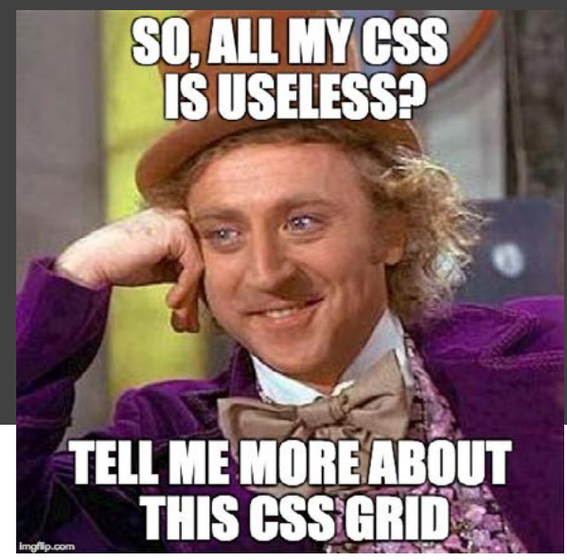
De items respecteren de inhoud van hun cel: groter als er meer inhoud is, kleiner als er minder is.

- **Door Flexbox kunnen we de tussenruimte gelijkmatig verdelen.**
- **Je maakt kolommen door de items een breedte te geven, maar daardoor zijn ze (minder) flexibel.**
- **Flexbox is niet bedoelt om hele pagina-layout te maken; enkel om item in een column of rij uit te lijnen.**



GRID vertrekt vanuit het grid zelf

Je defineert eerst je grid, en je items volgen het grid.



https://codepen.io/petervandenheuvel/pen/qJqpMB

‘OUDE MANIEREN’ OM EEN GRID TE MAKEN:

FLOATS:

```
.BOX {
        FLOAT: LEFT;
        WIDTH: 33.3333%;
}
```

THE FLOAT PROPERTY AND WIDTH ARE ADDED TO THE ITEM…

INLINE-BLOCK:

```
.BOX {
        DISPLAY: INLINE-BLOCK;
        WIDTH: 33.3333%;
}
```

THE DISPLAY PROPERTY AND THE WIDTH ARE ADDED TO THE ITEM…

FLEXBOX:

```
.CONTAINER { DISPLAY: FLEX; }
.BOX {
        FLEX-BASIS: 33.33%;
}
```

THE DISPLAY PROPERTY IS ADDED TO THE PARENT AND WIDTHS ARE ADDED TO THE ITEM…

GRID:

CSS GRID IS ANDERS;
WE MAKEN HET GRID AAN OP DE PARENT-ELEMENT.

WE HOEVEN GEEN EIGENSCHAPPEN OP DE ITEMS TOE TE PASSEN….

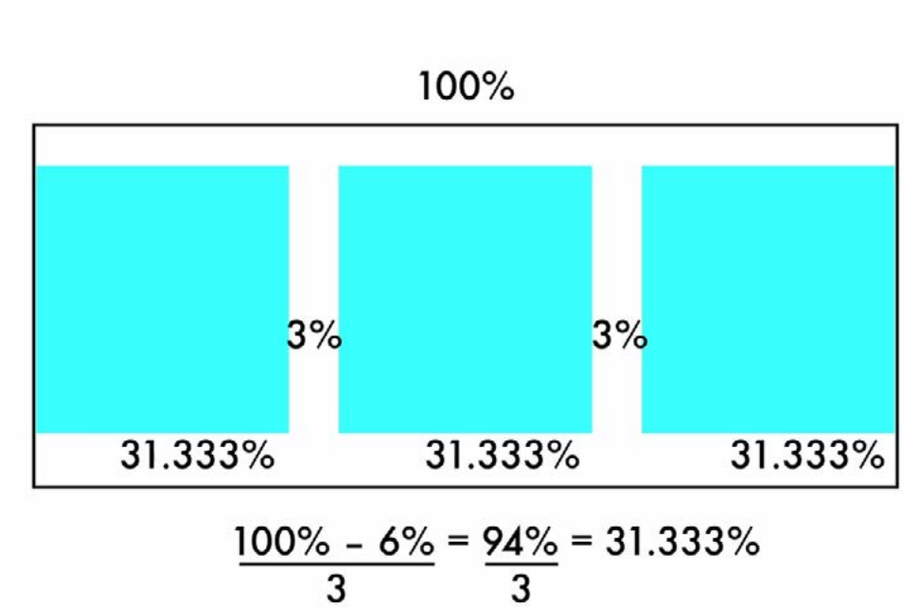
```
1 .grid{
2     display: grid;
3     grid-template-columns: 1f 1fr 1fr;
4     grid-gap: 20px;
5 }
```

GRID: FRACTIONS

fr = fraction.
Het pakt de ruimte die beschikbaar is en deelt deze over alle gedeelten die gelabelled zijn met ‘fr’.

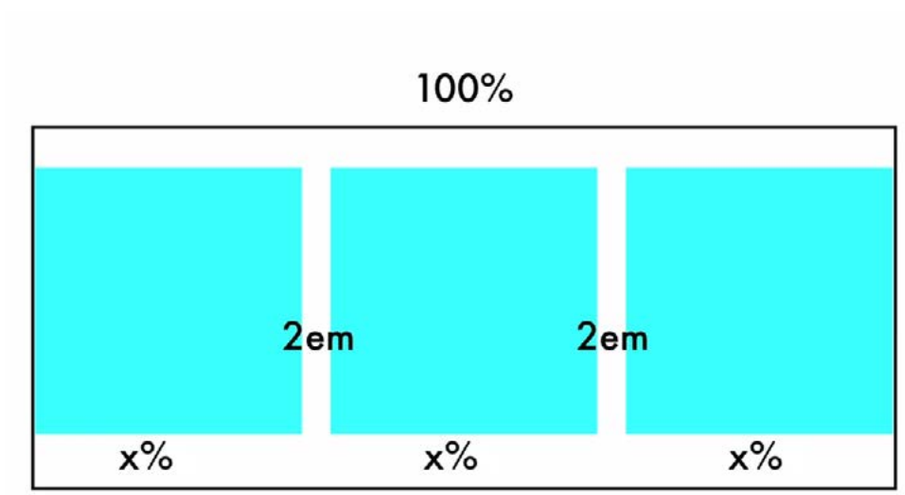
Het is ontwikkelt met als doel ‘het rekenwerk’ uit layouts te nemen.

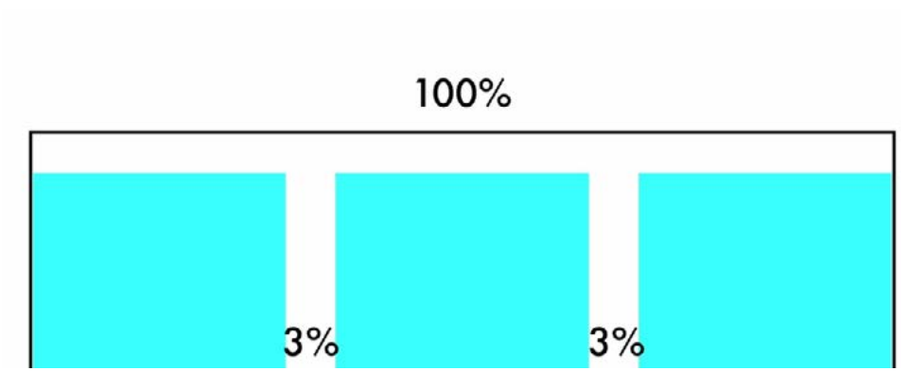
In plaats van rekensommen, of de calc( ) functie met media-query’s te moeten gebruiken, laat je de browser het zelf ‘uitzoeken’.



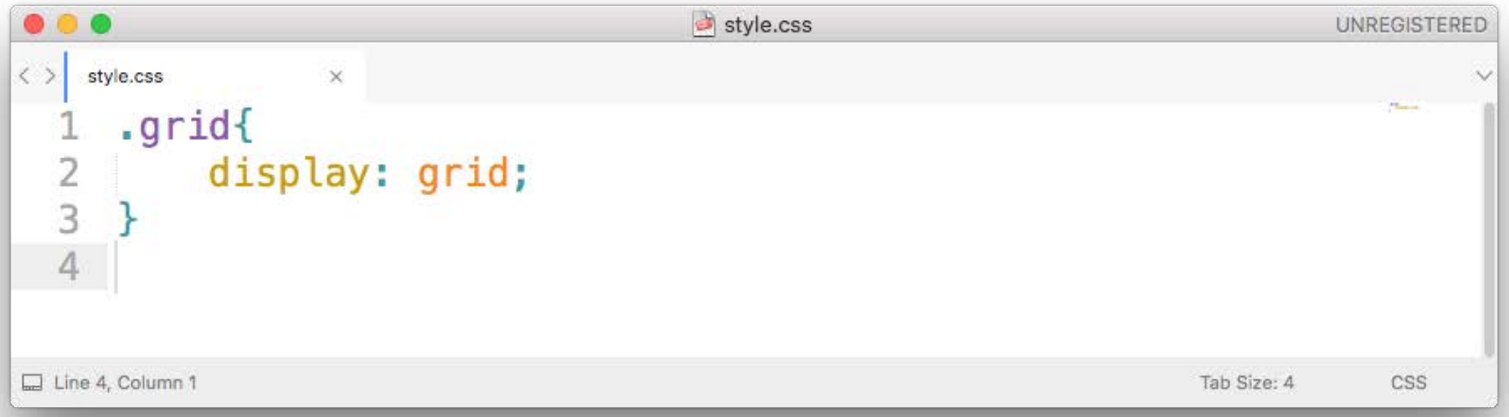
Je mag flexibele kolommen ‘mixen’ met andere afmetingen:

```
.grid{
      grid-template-columns: 100px 1fr 5fr;
}
```





GRID-CONTAINER



Een element dat een grid bevat, geactiveerd door:

```
.container {
    display: grid;
}
```

Je pagina kan meerdere 'grids' bevatten.

GRID-ITEM

- *Alle directe kinderen van het element met 'display:grid'.*
- *Alleen 'first-level-decendants' (directe kinderen)*

index.html — UNREGISTERED

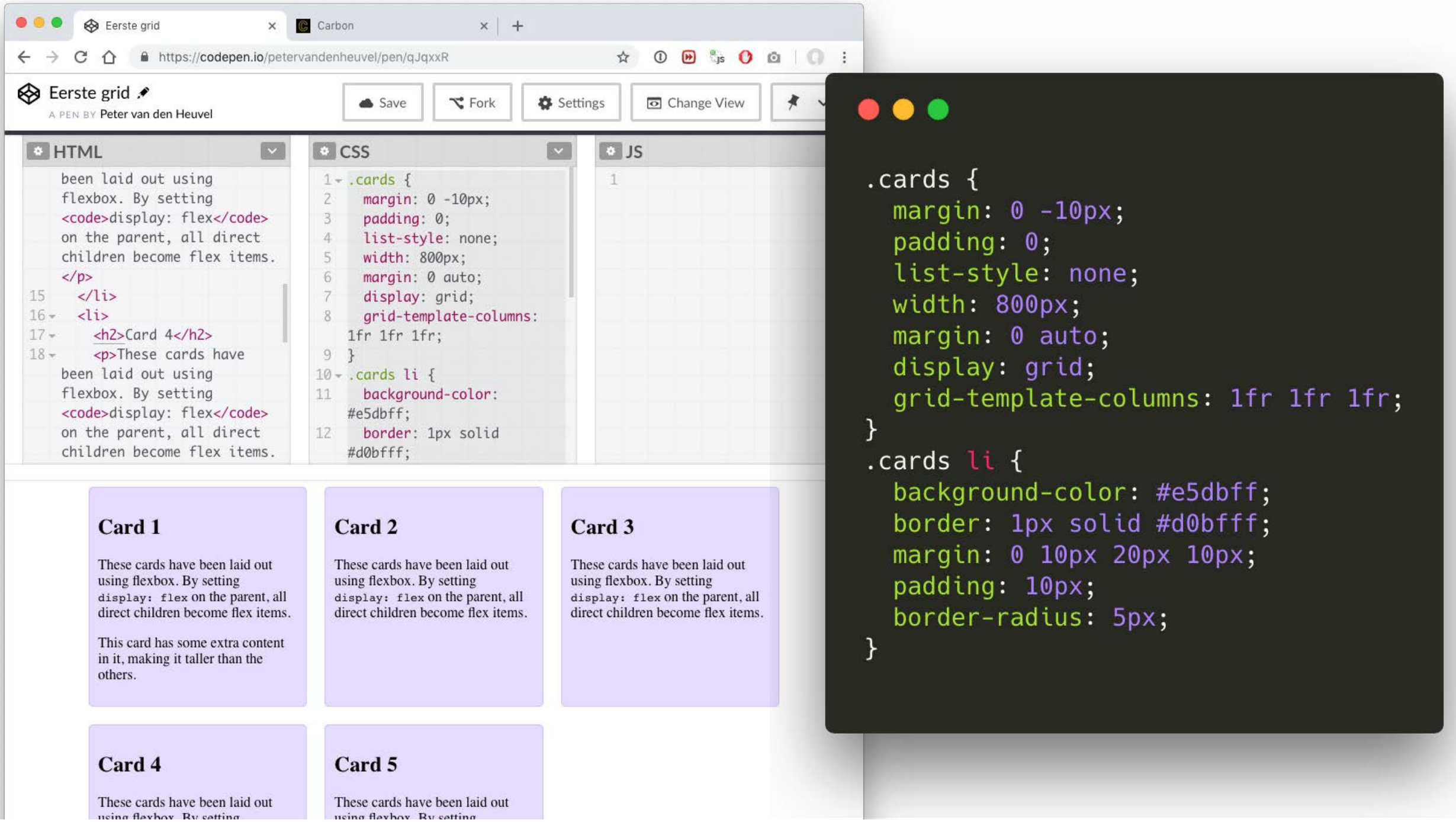
index.html • | style.css

```
1  <div class="container">
2
3      <div class="item">1</div>
4
5      <div class="item">2
6          <p class="sub-item">2a</p>
7      </div>
8
9      <div class="item">3</div>
10
11 </div>
```

Line 10, Column 5 — Tab Size: 4 — HTML

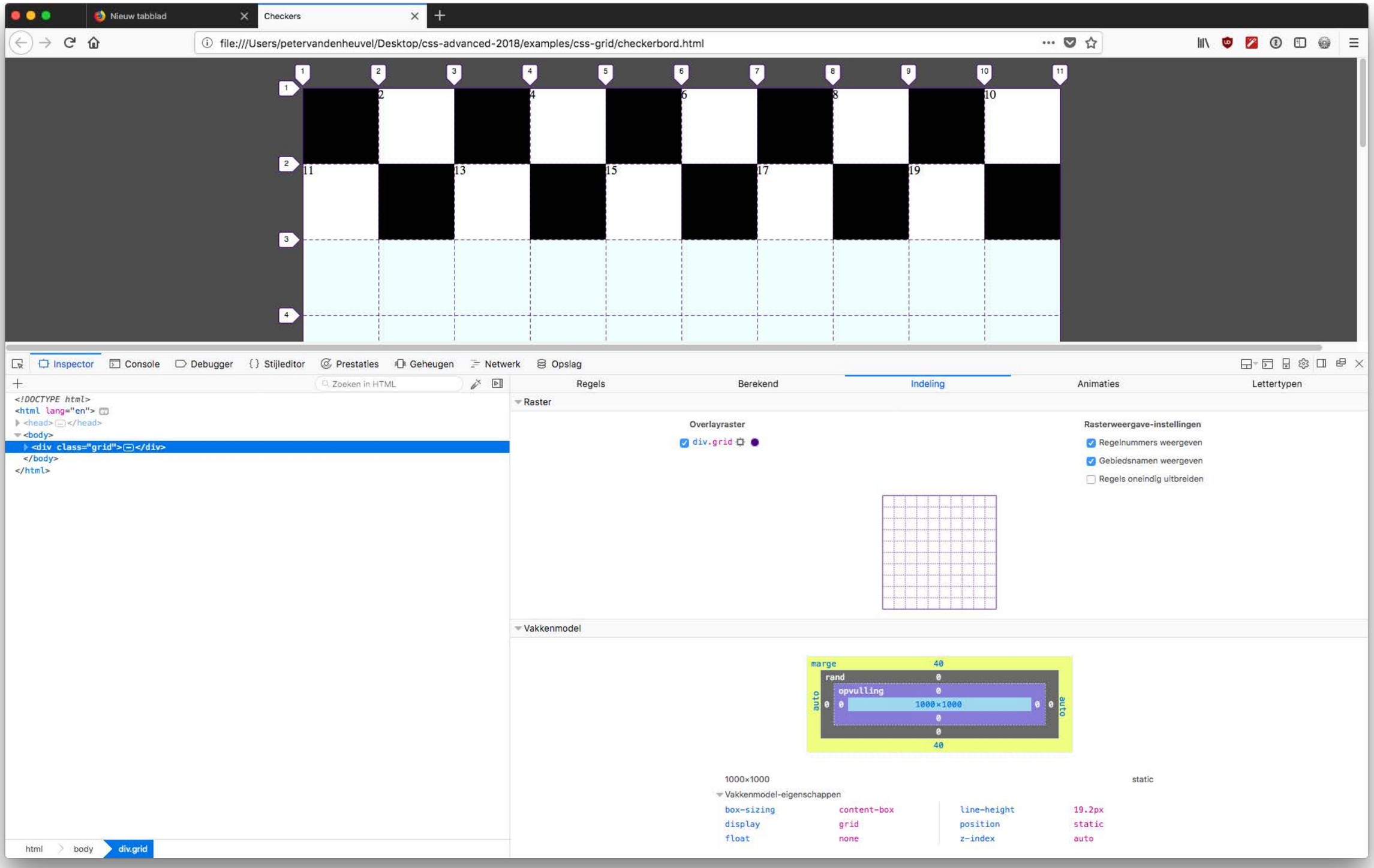
VOORBEELD

https://codepen.io/petervandenheuvel/pen/qJqxxR




```
.cards {
  margin: 0 -10px;
  padding: 0;
  list-style: none;
  width: 800px;
  margin: 0 auto;
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
.cards li {
  background-color: #e5dbff;
  border: 1px solid #d0bfff;
  margin: 0 10px 20px 10px;
  padding: 10px;
  border-radius: 5px;
}
```

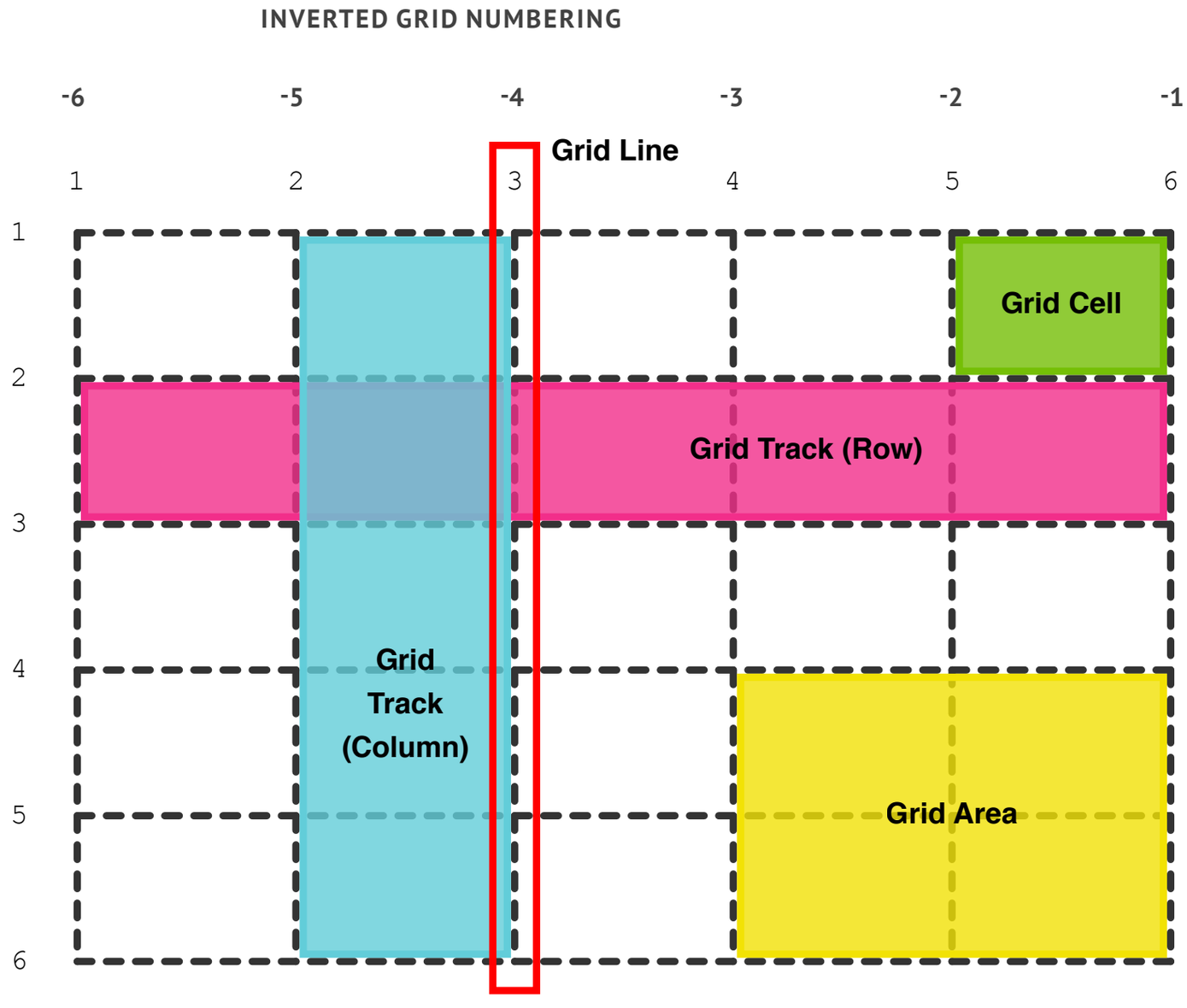
FIREFOX QUANTUM



CARABAS HANDS-ON TRAINING PRINT·WEB·OFFICE

CEVOFORA

GRID-TERMINOLOGIE



NEGATIEF OP DE RIJ-AS; DAN CREEERT HET GRID ER RIJEN, BIJ TOTDAT HIJ WEER POSITIEF KAN VERDER TELLEN.

HTTPS://CODEPEN.IO/PETER-VANDENHEUVEL/PEN/BBQAV-B?EDITORS=1100

GRID-GAP

*De lege ruimte tussen **grid lines***
plaats op container



VOORBEELD LAYOUT MAKEN

EMMET: DIV.SITE>HEADER+H1+MAIN+ASIDE+FOOTER

https://codepen.io/petervandenheuvel/pen/PybamR

```
<div class="site">
    <header>header</header>
    <h1>h1</h1>
    <main>main</main>
    <aside>aside</aside>
    <footer>footer</footer>
</div><!-- /.site -->
```



GRID: TEMPLATE-COLUMNS

other.css

index.html | style.css | other.css

```
1  .site{
2      display: grid;
3      grid-template-columns: 2fr 1fr 1fr;
4  }
5
```

Voorbeeld layout | Carbon

https://codepen.io/petervandenheuvel/pen/PybamR?editors=1...

Voorbeeld layout

A PEN BY Peter van den Heuvel

Save | Fork | Settings | Change View

HTML

```
<div class="site">
  <header>header</header>
  <h1>h1</h1>
  <main>main</main>
  <aside>aside</aside>
  <footer>footer</footer>
</div><!-- /.site -->
```

CSS (SCSS)

```
* {
  margin: 0;
  padding: 0;
}
div.site {
  // our grid here
  display: grid;
  grid-template-columns: 2fr 1fr 1fr;
}
```

header | h1 | main

aside | footer

de items plaatsen zich 1 voor 1 in het grid

Console | Assets | Comments | Last saved less than a minute ago | Delete | Share | Export | Embed | Collections

AANMAKEN VAN GRID-COLUMNS.

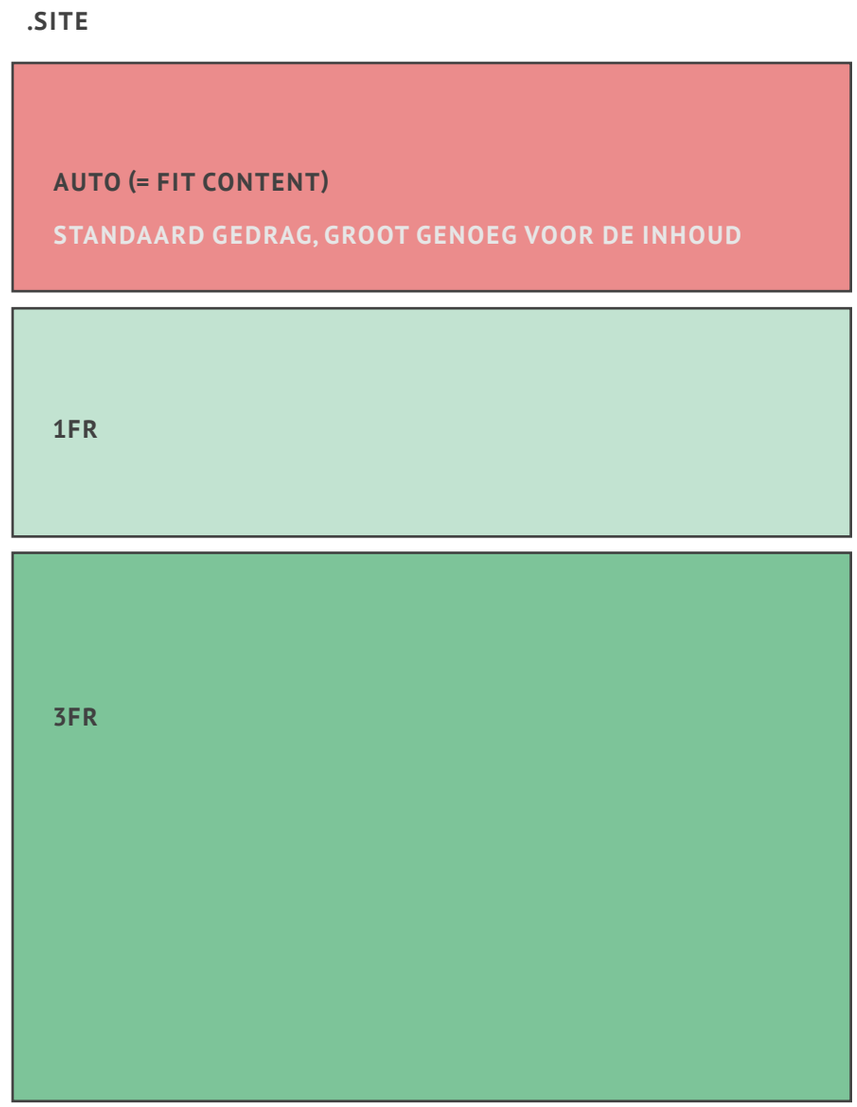
- Je vult de lijst van lengte waarden in die je wil hebben (in volgorde).

- Je kan kiezen uit (em, px, procent %, fr, ..etc.)

- Dat is de afstand tussen elke lijn.

GRID: TEMPLATE-ROWS



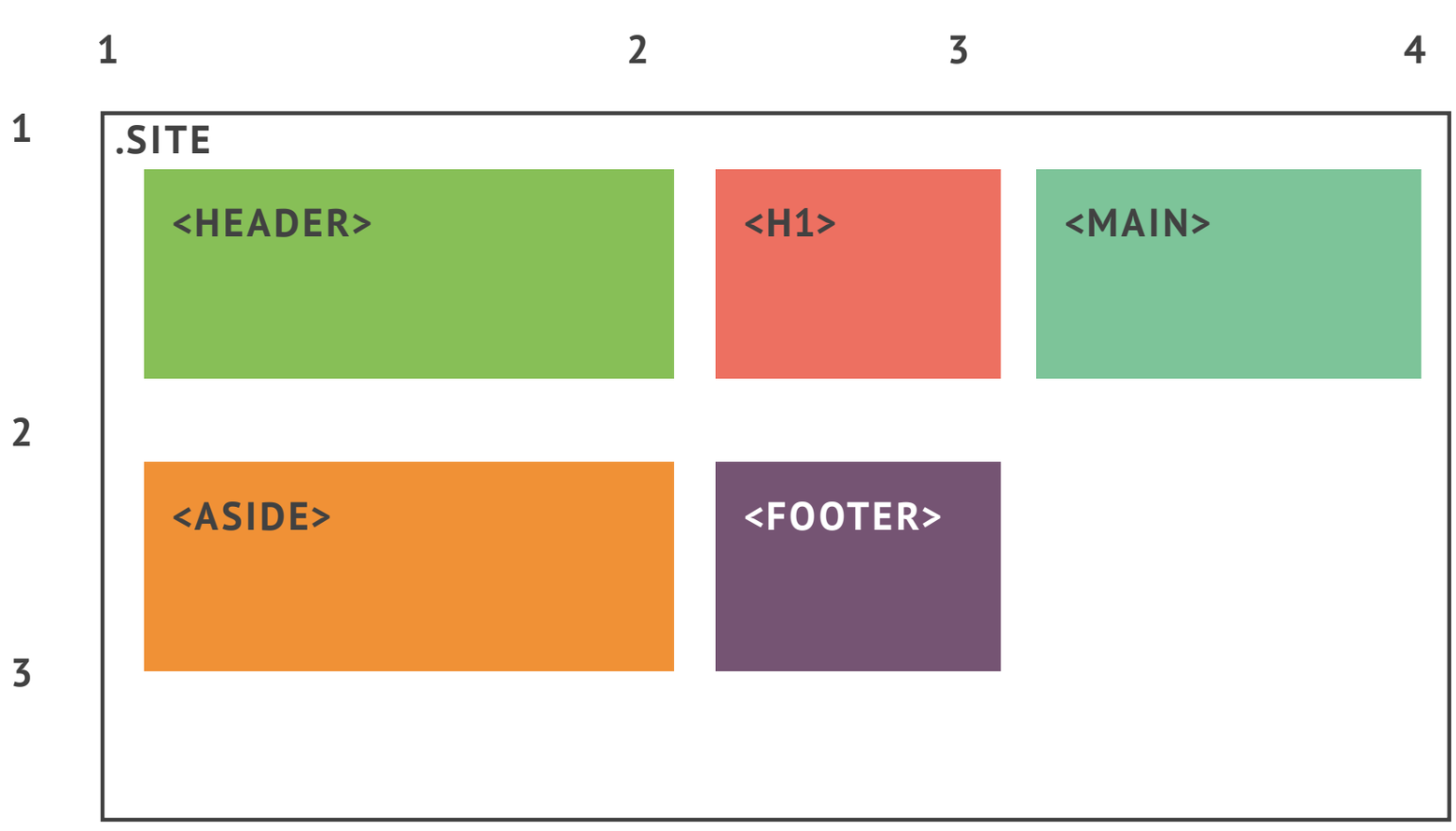


GRID: PLAATSING ITEMS

Grid items plaatsen zich automatisch in het grid (AUTO-placement),

- van Linksboven naar rechtsonder

- volgens de HTML-volgorde in de code



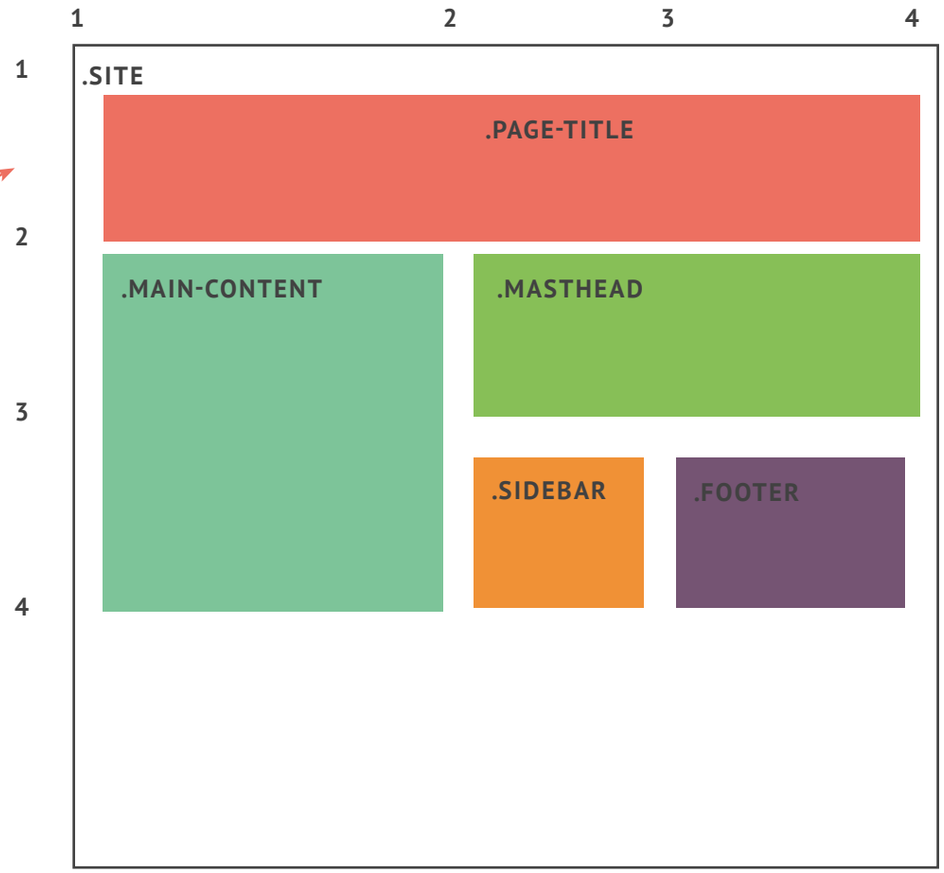
GRID: COLUMN & ROW




```css
h1{
  grid-column: 2 / 4;
  grid-row: 1 / 3;
}
```

GRID: COLUMN & ROW

```css
.site{
    display: grid;
    grid-template-columns: 2fr 1fr 1fr;
    grid template-rows: auto 1fr 3fr;
}

.masthead {
    grid-column: 2 / 4;
    grid-row: 2 / 3;
}

.page-title {
    grid-column: 1 / 4;
    grid-row: 1 / 2;
}

.main-content {
    grid-column: 1 / 2;
    grid-row: 2 / 4;
}
```



etc..

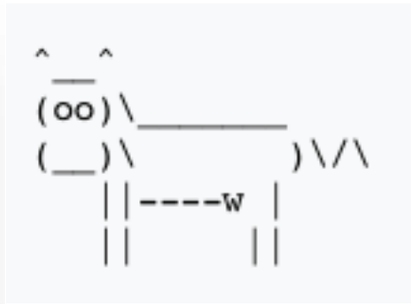
De overige elementen plaatsen zich gewoon weer een voor een in de grid-plekken die vrij zijn. (auto-placement)

GRID: AREAS

RESPONSIVE IS DIT NIET HANDIG:
DE NUMBERS ONTHOUDEN, EN PER MEDIA-QUERY AANPASSEN: VERSCHRIKKELIJK!

GEEFT DE AFZONDERLIJKE CELLEN EEN NAAM OP VOLGORDE
EEN BEETJE ZOALS ASCII-ART...

```
div.site {
  display: grid;
  grid-template-columns: 2fr 1fr 1fr;
  grid-template-rows: auto 1fr 3fr;
  grid-template-areas:
    "header header header"
    "main h1 h1"
    "main aside aside"
    "footer footer footer";
}

header {
  background: #87bf57;
  grid-area: header;
}
h1 {
  background: #ed7061;
  grid-area: h1;
}
main {
  background: #7dc499;
  grid-area: main;
}
aside {
  background: #f09136;
  grid-area: aside;
}
footer {
  background: #755473;
  grid-area: footer;
}
```

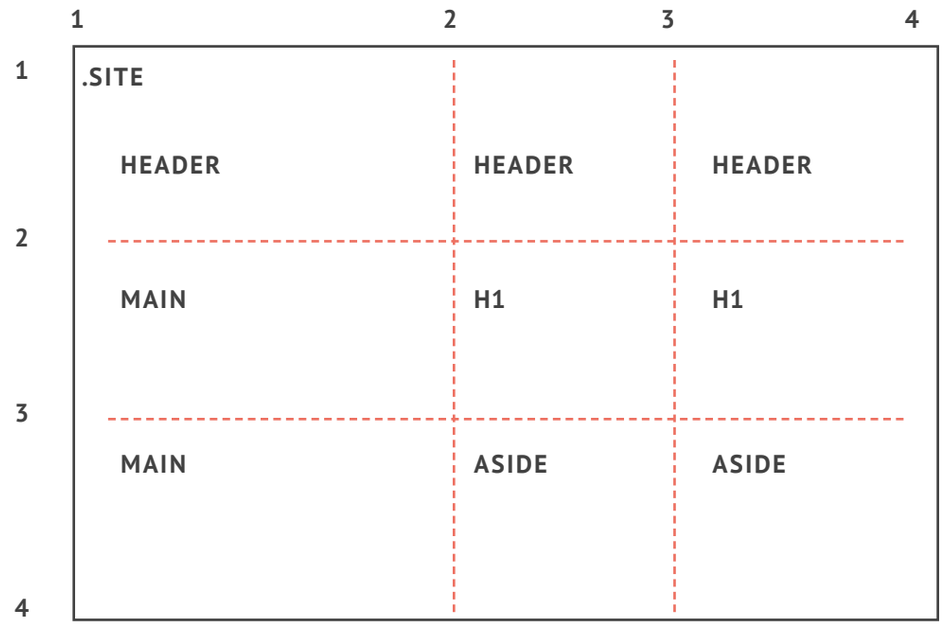


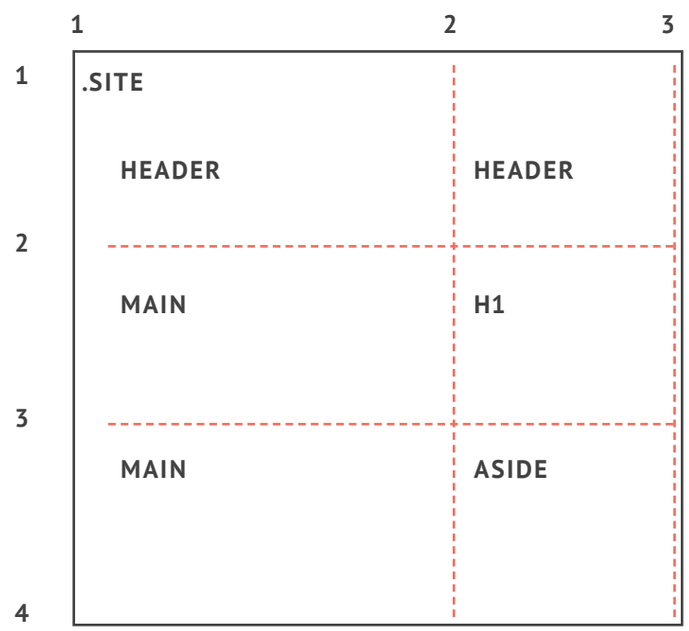
	1	2	3	4
1	.SITE			
	HEADER	HEADER	HEADER	
2	MAIN	H1	H1	
3	MAIN	ASIDE	ASIDE	
4	FOOTER	FOOTER	FOOTER	

GRID: RESPONSIVE

ALS HET SCHERM KLEINER WORDT, KAN JE KOLOMMEN WEGHALEN:

```
.grid{
  display: grid;
  background: lightgrey;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: 1fr;
  grid-gap: 10px;
  max-width: 960px;
  margin: 0 auto;
}

@media screen and (max-width: 800px){
  .grid{
    width: 90vw;
    grid-template-columns: 1fr;
  }
}
```





GRID: MINMAX-UNIT

Met **minmax( )** kan je zowel een **minimale** als een **maximale** waarde geven aan een **kolom**.

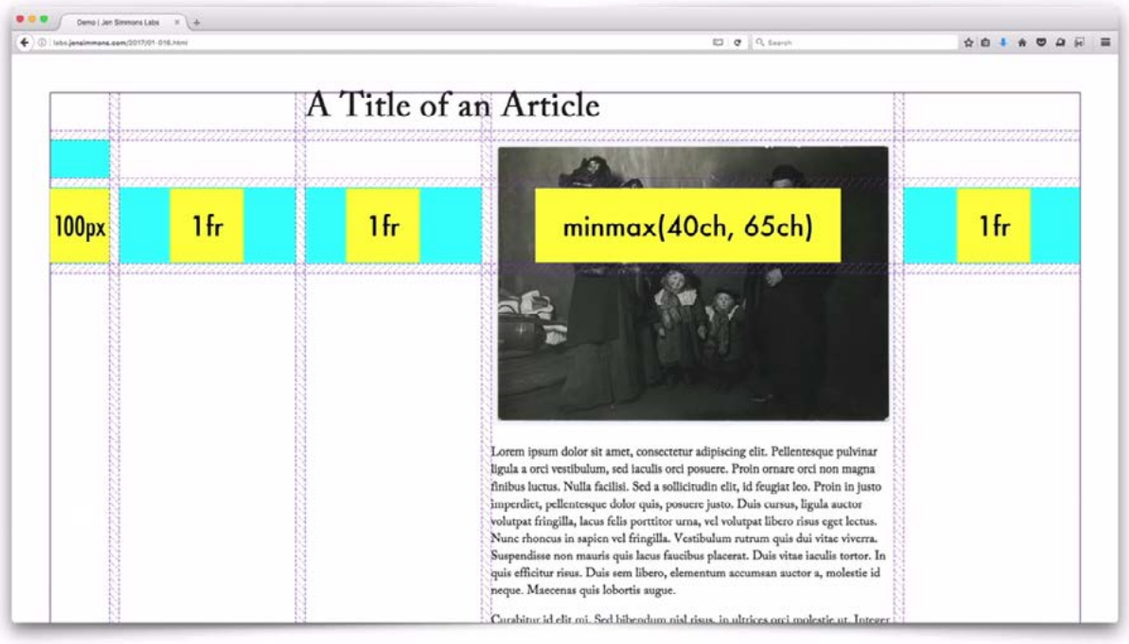
Stel je voor je wil een afbeelding-grid maken met afbeeldingen van 300px breed. Dat werkt prima als de container 600px is, of 900px, maar wat doe met de **tussenliggende waarden***?*

In Grid kan je zeggen, minimaal 300px, en maximaal 1fr.

Jen Simmons maakt een aantal voorbeelden waar ze de kolombreedte instelt op een ***aantal tekens***, minimaal 40 tekens en maximaal 65 tekens.

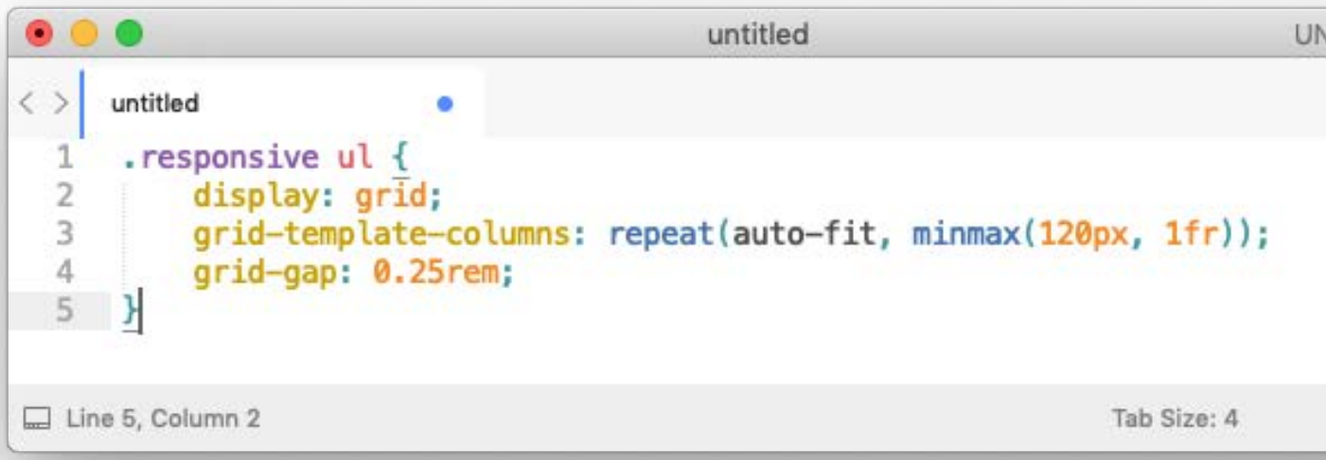
Dan worden je websites pas ***ècht responsive***, dat kan je nooit bekomen met 'oude CSS' of CSS-Frameworks.

VOORBEELD:
https://labs.jensimmons.com/2017/01-003C.html



Voorbeeld op:
https://labs.jensimmons.com/2017/01-016.html

MEER UITLEG:
HTTPS://WWW.YOUTUBE.COM/WATCH?V=MVQINPQXOV8



GRID: REPEAT

Het kan erg vervelend worden als je bij grote grid, bijvoorbeeld een 12 kolommen layout, 12x je breedte van de kolom moet ingeven.

Gelukkig is er de mogelijkheid tot herhaling, met zelfs de meest vreemde patronen:
https://codepen.io/petervandenheuvel/pen/drJvKg?editors=1100

untitled — UNREGISTERED

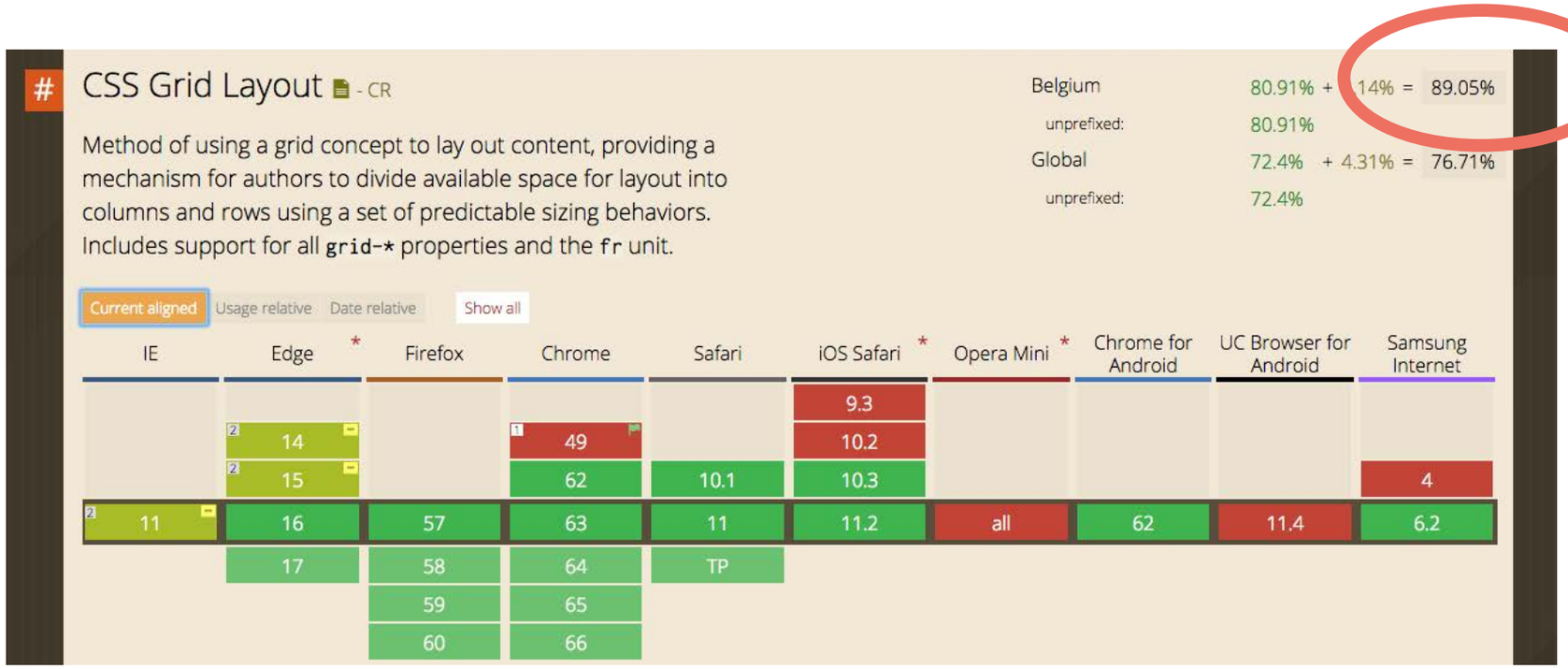
untitled

```
{ 1  .grid{
  2      display: grid;
  3
  4      grid-template-columns: 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr, 1fr;
  5
  6      grid-template-columns: repeat(12, 1fr); // herhaal 12x '1fr'
  7
  8      grid-template-columns: repeat(6, 1fr 2fr); // 12 column met patroon.
  9
 10      grid-template-columns: 50px repeat(6, 1fr 2fr) 100px; // Go crazy...
}11  }
```

Line 2, Column 19 — Tab Size: 4 — CSS

ONDERSTEUNING

WHAT ABOUT OLDER BROWSERS?
ALLE BROWSERS ONDERSTEUNEN HET!



STATS FROM:
HTTP://GS.STATCOUNTER.COM/

NU AL GEBRUIKEN?

IT'S TOO SOON..
THE NEW YORK TIMES GEBRUIKT HET AL (JUN 2017)...



Dit artikel is zeer de moeite waard omdat Natalya 'gedwongen' wordt om Bootstrap 3 te ondersteunen.

Zij gebruikt CSS om de 'bootstrap styles' te unstylen, en dan tussen alle Bootstrap code in, haar eigen CSS Grid code te laden voor browsers die het ondersteunen.

Op die manier kan ze de layout verbeteren voor nieuwe browsers.

HTTPS://OPEN.NYTIMES.COM/BOOTSTRAP-TO-CSS-GRID-87B3F5F830E4

HTTPS://CODEPEN.IO/TALLYS/PEN/BRRLYV

INTERNET EXPLORER?

WHAT ABOUT IE?..

THE  ELEPHANTS IN THE ROOM..

Deze browsers hebben grid bedacht in 2010, alleen de andere browsers hebben de spec aangepast..

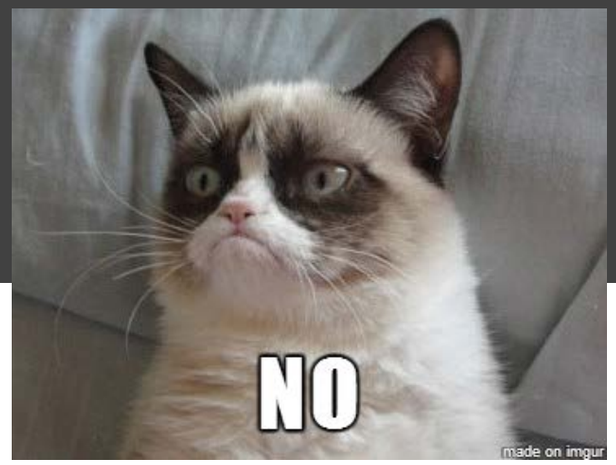
Gebruik de ‘mobile first’-filosofie; bouw de hele site eerst voor mobile, en pas CSS grid alleen toe voor aparaten die het ondersteunen. Good enough for mobile = good enough for desktop.

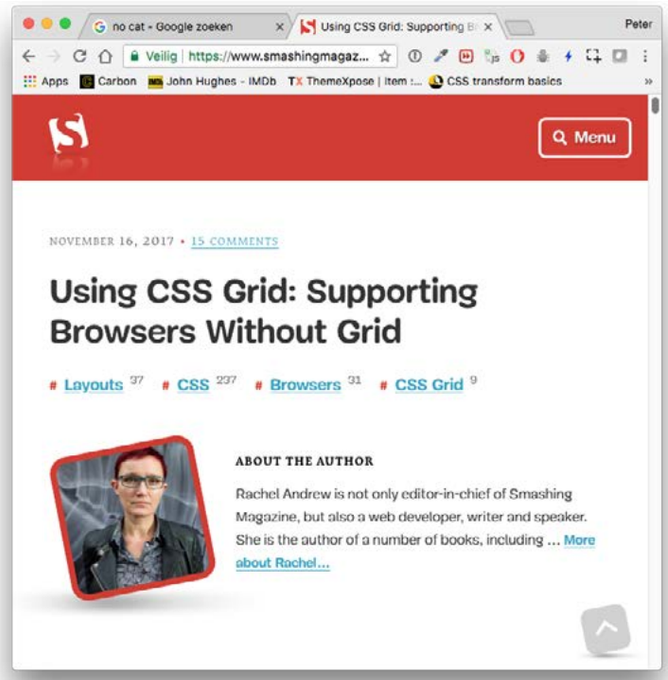
Gebruik Feature-query’s & progressive enhanchment waar nodig.

INTERNET EXPLORER EXISTS. USE CSS GRID?
HTTPS://WWW.YOUTUBE.COM/WATCH?V=7MSERXU7IVG

SHOULD I TRY TO USE THE IE IMPLEMENTATION OF CSS GRID LAYOUT?
HTTPS://WWW.RACHELANDREW.CO.UK/ARCHIVES/2016/11/26/SHOULD-I-TRY-TO-USE-THE-IE-IMPLEMENTATION-OF-CSS-GRID-LAYOUT/

ARE THERE ANY POLYFILLS?





- **CSS Grid is vrij complex,**
  Zelfs al zouden we een JavaScript kunnen ontwikkelen die dit namaakt,
  dan gaat het heel erg traag zijn.

- **Oudere Browsers ondersteunen het meestal niet, meestal zijn dat oude computers en dan zijn Polyfills vaak rampzalig voor performance. Daarom geef deze gewoon de mobile versie van de site.**

- **Wil je het toch met een JavaScript proberen?**
  https://github.com/FremyCompany/css-grid-polyfill

- **Opinie-stuk van de experts:**
  https://www.smashingmagazine.com/2017/11/css-grid-supporting-browsers-without-grid/

FEATURE QUERY

GEBRUIK EEN FEATURE QUERY OM GRID TE ACTIVEREN:

index.html style.css other.css

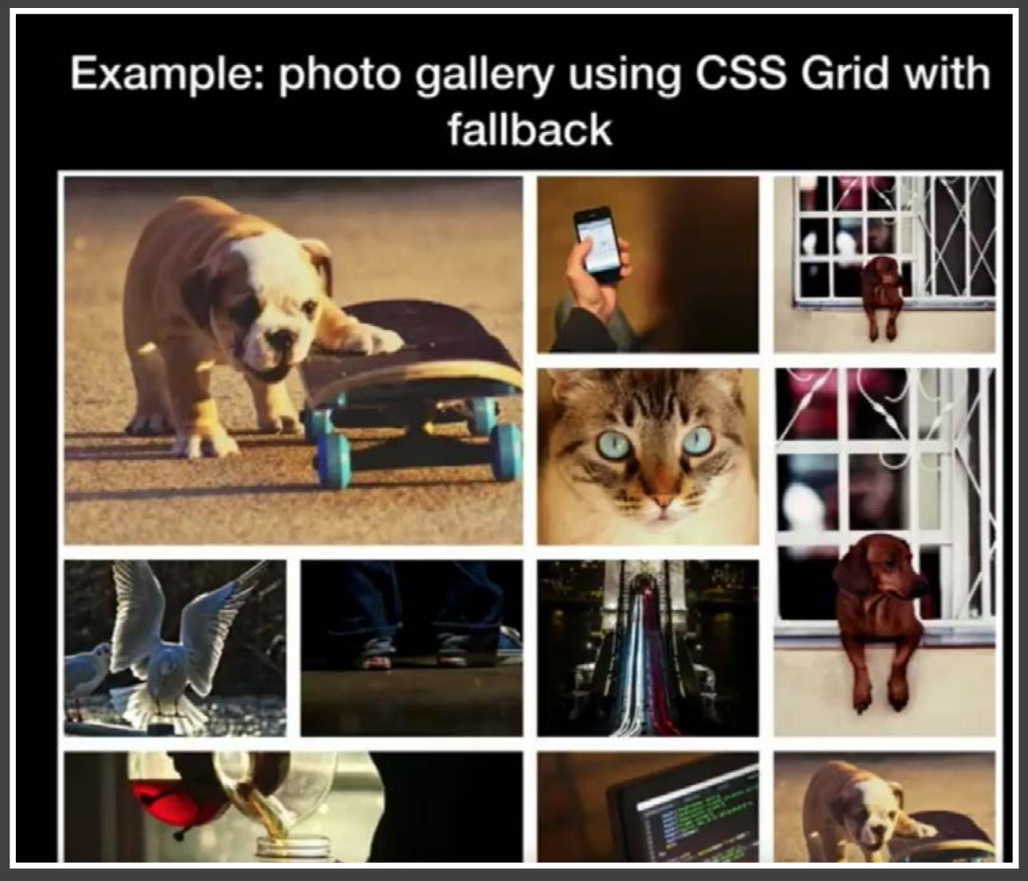
```
/* First Thought: (don't use it!) */
@supports (display: grid){
    .grid-container {
        display: grid;
    }
}

/*  Better: only newer browsers.
    Not 'old IE grid...'
*/

@supports (grid-area: auto){
    /* Your code.. */
}
```

MEER TIPS:
HTTPS://RACHELANDREW.CO.UK/CSS/CHEATSHEETS/GRID-FALLBACKS

Er zijn twee syntaxen van CSS Grid; de oude van Internet explorer en de nieuwe 'officiele'.
Test op een property van de nieuwe syntax.. bijvoorbeeld 'grid-area: auto'.

FALL BACK

Browsers slaan code over die ze niet herkennen.
Maak daar gebruik van.



Dit voorbeeld is vrij lastig in float-based of Flexbox CSS:
Je moet ***heights*** gaan instellen en je zal meerdere '***wrappers***' nodig hebben.

Wij gaan het anders aanpakken; we voorzien een basis, en verbeteren de ervaring waar mogelijk.

Voorbeeld op: https://codepen.io/petervandenheuvel/pen/Pybxrw

FALL BACK

Voorzie code die ‘goed genoeg’ is voor iedereen



VERBETER INDIEN MOGELIJK

“PROGRESSIVE ENHANCEMENT”;
HET GELEIDELIJK VERBETEREN VOOR NIEUWE BROWSERS

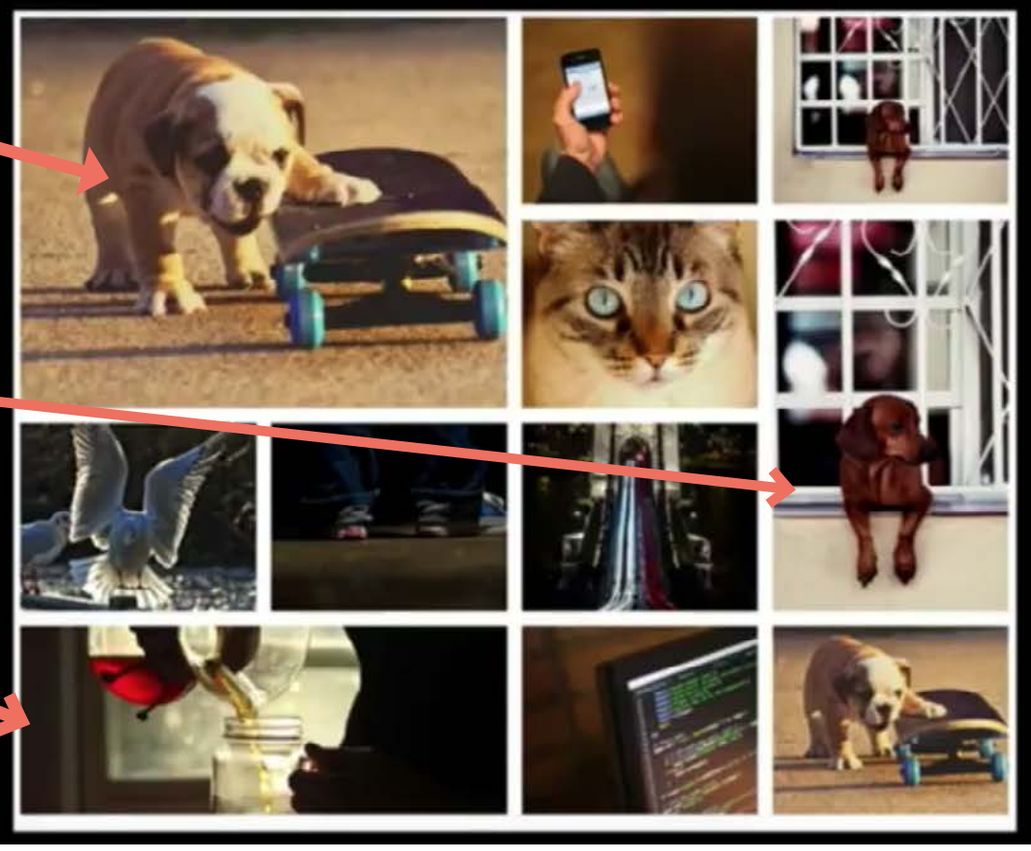
```
/* detect of css-grid werkt: */
@supports (grid-area: auto){
  .grid-gallery{
    display:grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-gap: 10px;
  }
  .grid-gallery-item{
    margin: 0;
    width: 100%;
  }
}
```

FALL BACK

EN ENKELE ITEMS GROTER WEERGEVEN.

Example: photo gallery using CSS Grid with fallback

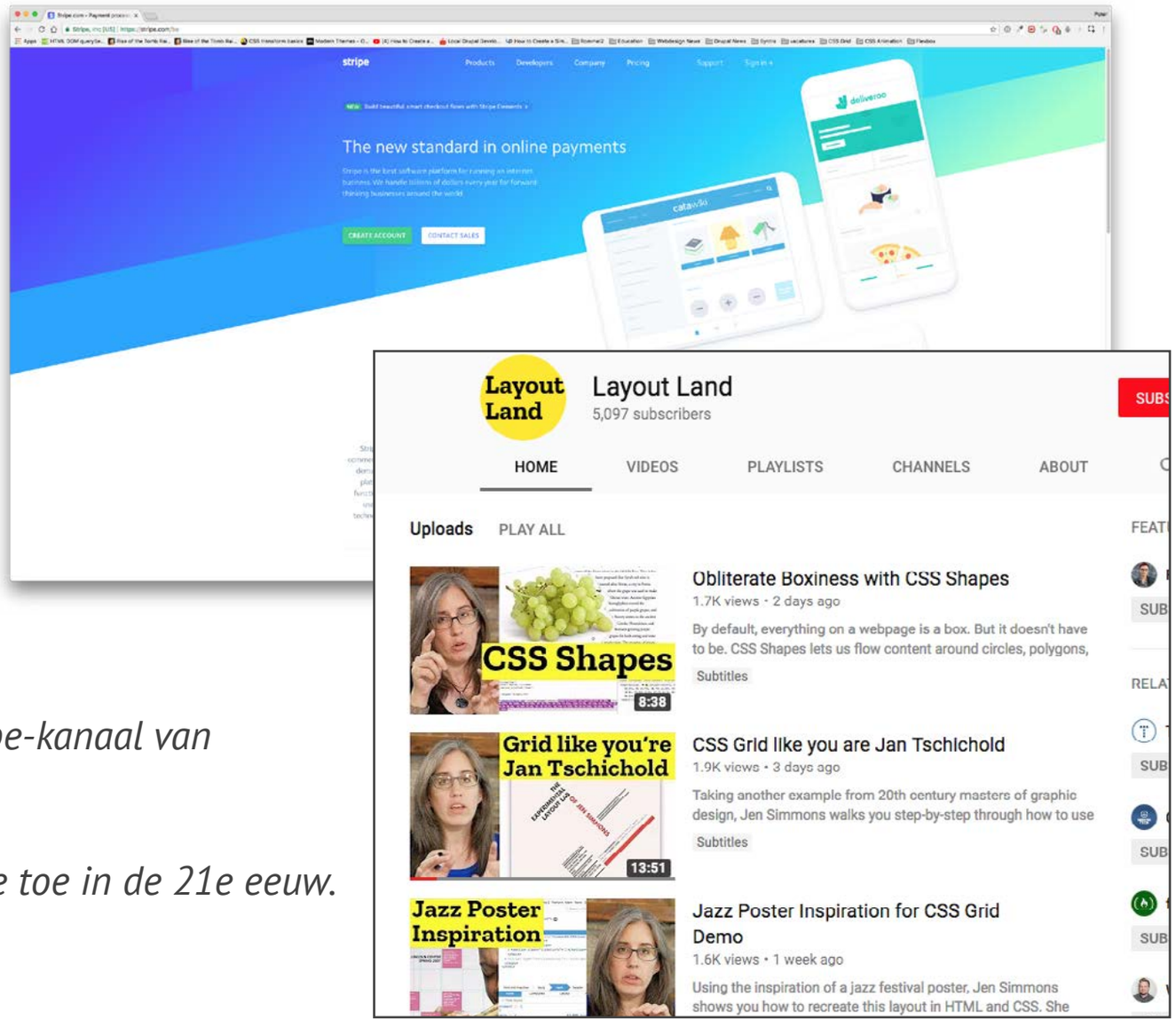
```
.grid-gallery-item:first-child {
  grid-row: span 2;
  grid-column: span 2;
}

.grid-gallery-item:nth-child(5) {
  grid-row: span 2;
}

.grid-gallery-item:nth-child(8) {
  grid-column: span 2;
}
```



FUN-FACT:

JE KAN HET GRID VERVORMEN MET CSS.
IDEE KOMT VAN STRIPE.COM

```
.grid{
    transform: skew(-10deg);
    display: grid;
}
```



Meer experimenten zoals deze kan je vinden op het Youtube-kanaal van 'Layout Land'.

Zij neemt werken van bijv. Piet Zwart uit 1925 en past deze toe in de 21e eeuw.

MIN- & MAX-CONTENT

Text-voorbeeld:
https://codepen.io/petervandenheuvel/pen/YgrMEM

Images-voorbeeld:
https://codepen.io/petervandenheuvel/pen/gEGyEP

Sidebar-voorbeeld:
https://codepen.io/petervandenheuvel/pen/oVGrwe

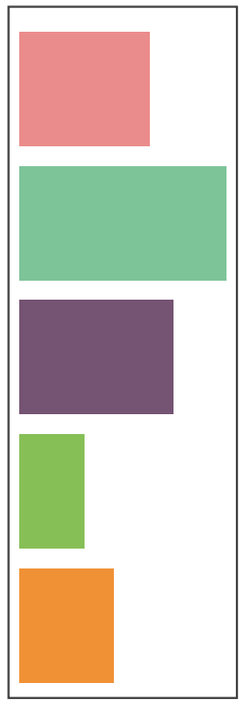
Working example:
https://codepen.io/petervandenheuvel/pen/PLQoEE

This is a phrase with several words.

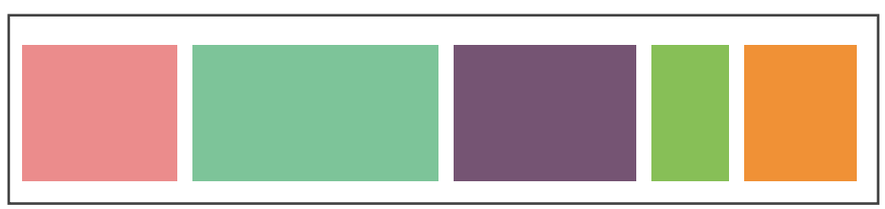
This is a phrase with several words. ← **max-content**

This is a phrase with several words. ← **min-content**

max-content
Bij max-content, zo breed als het grootste 'grid-item' in de kolom.



min-content
Is a keyword representing the largest minimal content contribution of the grid items occupying the grid track.



GRID: AUTO-FILL / AUTO-FIT

Er zijn twee mogelijke ‘keywords’ die je kan toevoegen bij de repeat-syntax, namelijk:
- auto-fill
- auto-fit

Zij zorgen beiden op een iets andere manier hoe er om moet worden gegaan met extra ruimte in de kolom-tracks.

Het verschil is dat bij ‘auto-fill’, de lege ruimte NIET wordt ingenomen door de grid-items, en bij ‘auto-fill’ WEL.

Auto-fill voegt dus ‘lege kolommen’ toe, auto-fit trekt de bestaande kolommen uit, zodat ze passen. Het effect is natuurlijk vooral zichtbaar bij ‘fixed unit’-kolommen.

https://codepen.io/petervandenheuvel/pen/EMoNEr?editors=1100

