

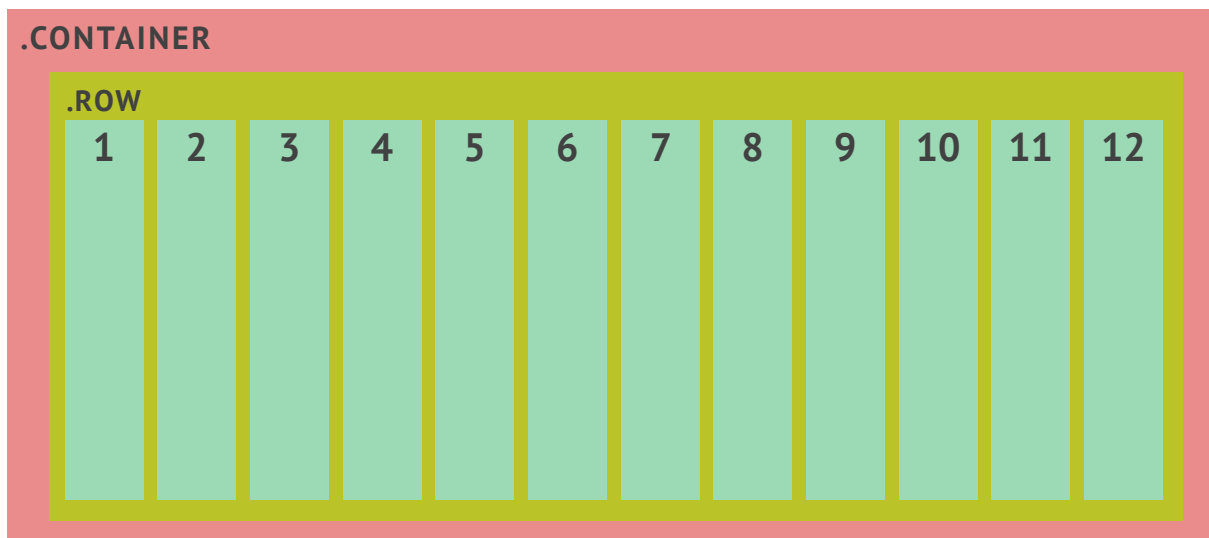
HOOFDSTUK 3

lay-outs maken met het gridsysteem

- *Hoe werkt het grid systeem?*
- *Structuur en opties voor het grid systeem.*
- *Het gridsysteem op verschillende schermformaten.*
 - *offset en nesting van kolommen*

HOE HET GRID WERKT

container, row & columns



Hoe het grid systeem werkt

We hebben al kort kennis gemaakt met het grid-systeem. Het systeem is geschreven met Flexbox, en werkt samen met een aantal slim opgestelde mediaqueries, zodat het perfect schaalbaar is op alle schermen.

Je kan zelf kiezen wanneer de kolommen opelkaar komen te staan of hoe breed ze zijn per schermgrootte.

In de basis bestaat het grid-systeem uit containers, rijen en kolommen. Elke rij kan naar keuze worden verdeeld in maximaal 12 kolommen. Het is niet verplicht om rijen volledig te vullen (tot 12).

De afbeeldingen geven enkele voorbeelden, van wat je kan maken.

.CONTAINER

.ROW

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

.ROW

6	6
---	---

.ROW

4	8
---	---

voorbeelden

.CONTAINER

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

3										3
---	--	--	--	--	--	--	--	--	--	---

.ROW		2								7
------	--	---	--	--	--	--	--	--	--	---

mobile

.CONTAINER

.ROW
KOLOM 1
KOLOM 2

GRID SYSTEMEN

niet het eerste, niet het laatste



Al jarenlang zijn er allerlei CSS-grid bibliotheken (CSS bestanden die je kan koppelen aan je HTML).

Bootstrap heeft dit wiel niet uitgevonden. Het bouwt verder op de ideeën die eerder tot uiting kwamen zoals 'Blueprint.css' en 960.gs en anderen. Deze zijn nu verouderd en worden niet meer gebruikt.

Je gebruikt dit soort systemen niet in combinatie, dus ofwel 960.gs ofwel bootstrap niet de combinatie.

REGELS

regels voor het grid systeem

Het werken met het Bootstrap-gridsysteem is op zich niet zo ingewikkeld. Er zijn echter wel een aantal regels waar je rekening mee moet houden.

Als je deze regels niet volgt, (of per ongeluk een sluit-tag van een div </div>) op de verkeerde plek zet, stort het hele grid systeem als een kaartenhuis in elkaar.

Lees daarom de volgende regels goed door:

- 1) Er zijn drie belangrijke onderdelen in het gridsysteem: containers, rijen en kolommen.
- 2) Containers worden aangegeven met de klasse .container of .container-fluid
- 3) Het grid-systeem werkt alleen als je rijen gebruikt binnen de container. Een rij los op de pagina heeft geen effect.

4) Inhoud moet geplaatst worden in kolommen. Kolommen moeten in rijen, en zijn DIRECTE kinderen.

5) Het grid systeem gaat tot 12.

6) De kolombreedten worden berekent in percentages (flex-basis)

7) Er zijn 5 niveau's voor het grid: de breakpoints (xs, sm, md. etc.)

8) De Grid-niveau's zijn gebaseerd op 'min-width'; van klein naar groot.

Leer deze regels best van buiten...

Layout-problemen? Overloop deze regels dan nog eens...

GRID REGELS

the first rules of bootstrap...



```
<div class="row">
  This is very bad, wrong way, no bueno!!
</div>

<div class="row">
  <p>This is also very bad, the wrong way!!</p>
</div>

<div class="row">
  <h2>No headings either! This is the wrong way!!</h2>
</div>

<div class="row">
  <div class="col">Happy :-) This is the right way.</div>
</div>
```

Regel 1:

Kolommen MOETEN een direct kind zijn van een .row

Regel 2:

Row's worden ALLEEN gebruikt om kolommen te bevatten, niets anders.

Regel 3:

Alle Rows worden in een container geplaatst.

De regels zijn belangrijk, de row's en columns werken samen. (Misschien is de negatieve padding al opgevallen).

Gebruik nooit de ene zonder de andere.

voorbeeld 1: horizontale layout, gestapeld op mobile..

.CONTAINER

.ROW

12

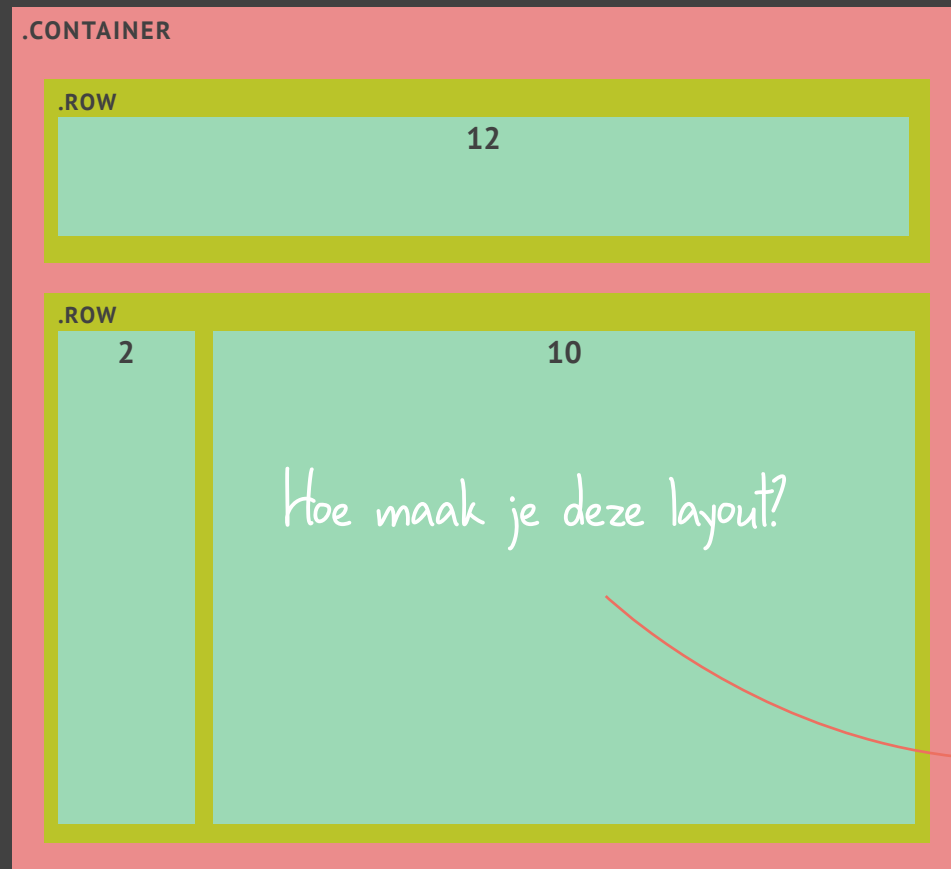
.ROW

2

10

Hoe maak je deze layout?

voorbeeld 1: horizontale layout, gestapeld op mobile..



```
Untitled-1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head> ...
8  </head>
9  <body>
10   <div class="container">
11     <div class="row">
12       <div class="col-md-12">
13         header
14       </div>
15     </div>
16   </div>
17   <div class="row">
18     <aside class="col-md-2">aside</aside>
19     <main class="col-md-10">main</main>
20   </div>
21 </div>
22 </body>
23 </html>
```

Untitled-1

Untitled-1

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head> ...
8  </head>
9  <body>
10   <div class="container">
11   |
12   |   <div class="row">
13   |   |   <div class="col-md-12">
14   |   |   |   header
15   |   |   |   </div>
16   |   |   </div>
17   |   </div>
18   |   <div class="row">
19   |   |   <aside class="col-12 col-md-2">aside</aside>
20   |   |   <main class="col-12 col-md-10">main</main>
21   |   </div>
22   </div>
23   </body>
24   </html>
```

Je mag combineren

Ln 22, Col 1 Spaces: 4 UTF-8 LF HTML Prettier

OFFSETING

een beetje witruimte

Soms wil je in je layout niet alles volproppen met content, het is immers geen krant waar iedere centimeter papier telt.

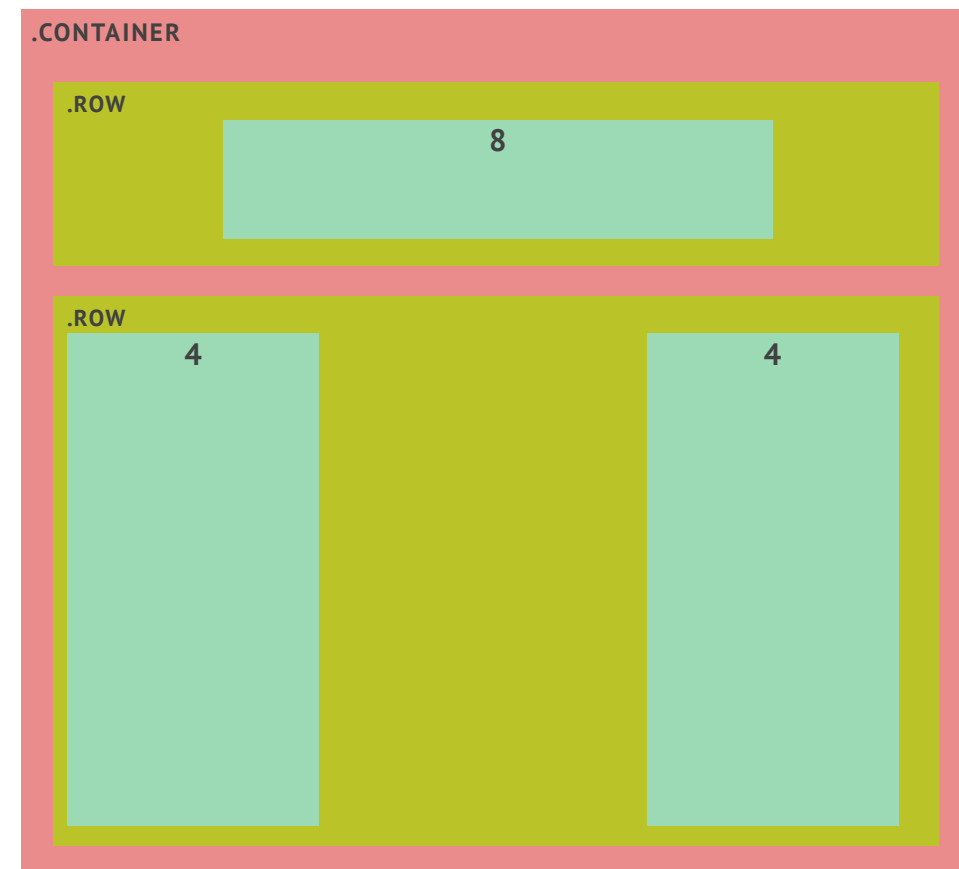
Misschien wil je een kolom centreren op de pagina of een lege ruimte laten in het grid.

Er bestaan klassen waarmee de kolommen naar rechts of links kan verschuiven. Dat doe je door de `.offset-*` te gebruiken. Waar het sterretje staat, vul je een getal in. Een `.offset-3` verschuift een kolom bijvoorbeeld 3 kolommen naar rechts.

Meer informatie:

<https://getbootstrap.com/docs/4.0/layout/grid/#offsetting-columns>

Opdracht



NESTEN

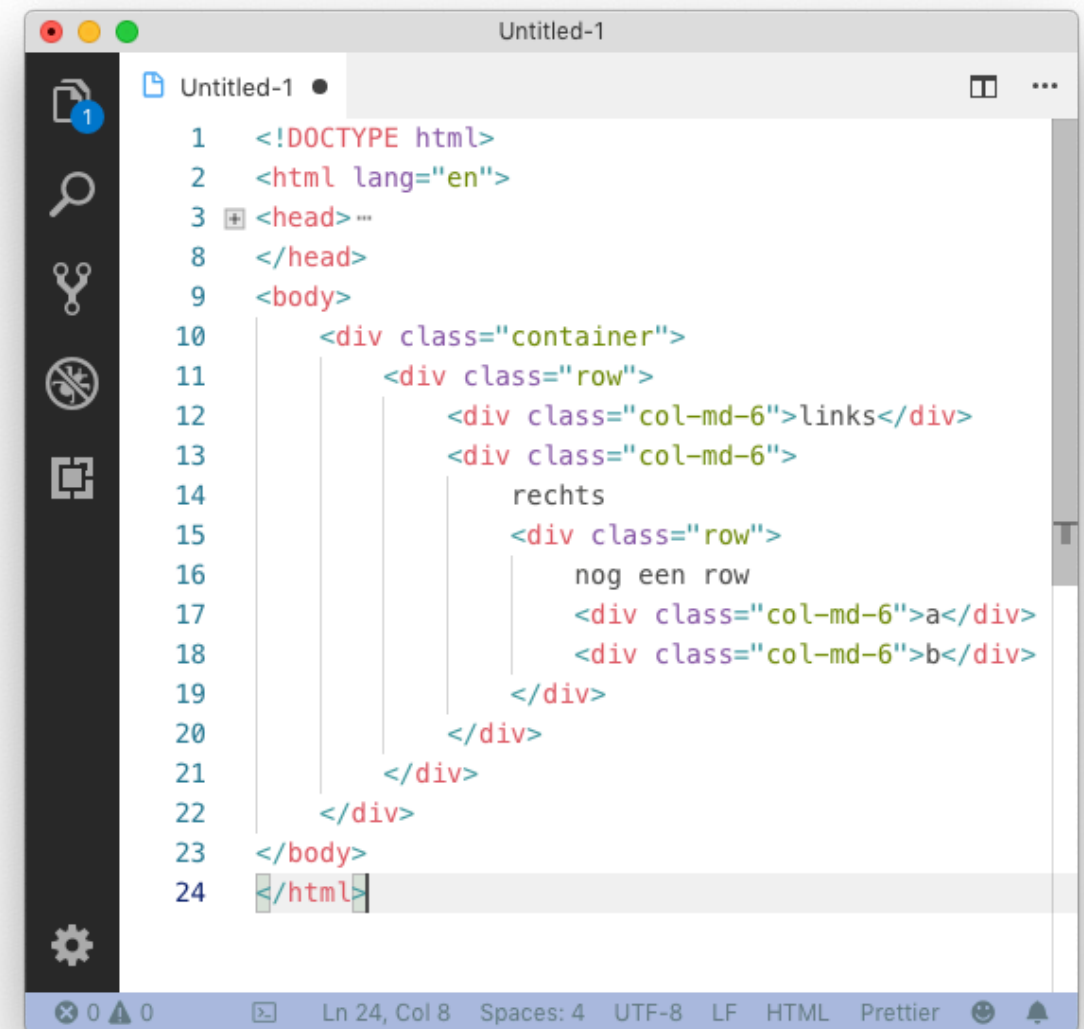
complexe layouts

Hoewel wordt aanbevolen om een layout zo eenvoudig mogelijk te houden, zal het soms toch voorkomen dat je een kolom moet verdelen in kleinere delen. Bijvoorbeeld in subkolommen, in dat geval gebruik je 'nesting'.

In dat geval krijgt de hoofdkolom (die de kleinere kolommen bevat), ook weer een .row (een flex-container), met daarin dus weer nieuwe flex-kinderen.

Binnen zo'n nieuwe rij (.row), werk je WEER met 12 kolommen. Ookal is deze een heel stuk smaller dan de gehele pagina layout. Het is een percentage gebaseerd systeem; remember?

Dat ziet er dan zo uit:



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head> ...
8  </head>
9  <body>
10     <div class="container">
11         <div class="row">
12             <div class="col-md-6">links</div>
13             <div class="col-md-6">
14                 rechts
15                 <div class="row">
16                     nog een row
17                     <div class="col-md-6">a</div>
18                     <div class="col-md-6">b</div>
19                 </div>
20             </div>
21         </div>
22     </div>
23 </body>
24 </html>
```

ORDER

wijzigen van volgorde

Volgorde van de kolommen aanpassen.

Behalve klassen om de offset van de kolommen te regelen, bestaat er ook een mogelijkheid om de volgorde van de kolommen te wijzigen.

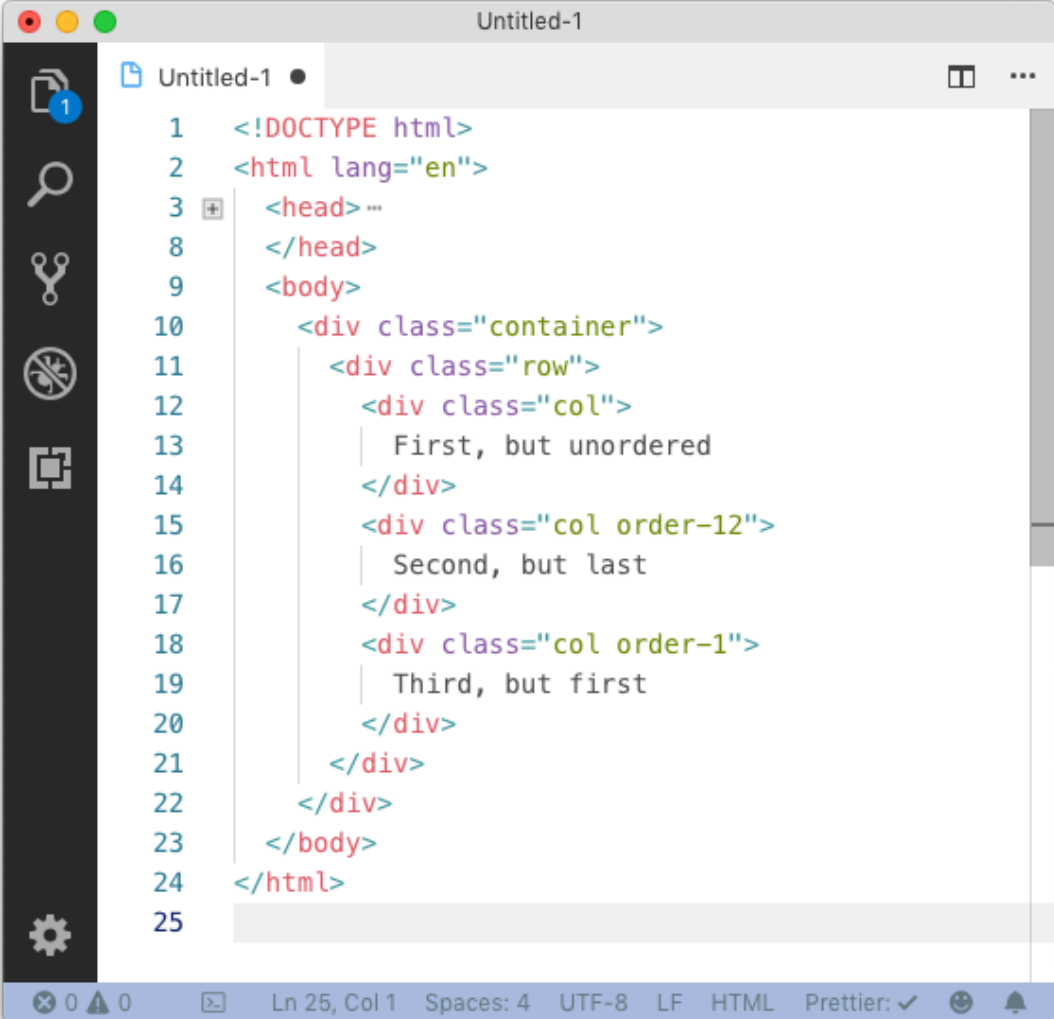
Dat gebruik je met de klassenaam: **.order-***

Uiteraard wordt het sterretje weer gebruikt om de volgorde te sturen.

Het advies is om deze techniek terughoudend toe te passen, het maakt je code erg complex en lastig te onderhouden.

Wil je meer informatie:

<https://getbootstrap.com/docs/4.0/layout/grid/#order-classes>



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head> ...
8  </head>
9  <body>
10  <div class="container">
11  <div class="row">
12  <div class="col">
13  | First, but unordered
14  </div>
15  <div class="col order-12">
16  | Second, but last
17  </div>
18  <div class="col order-1">
19  | Third, but first
20  </div>
21  </div>
22  </div>
23  </body>
24  </html>
25
```

VERTICAAL UITLIJNEN

nu eindelijk mogelijk...!

Verticaal uitlijnen:

Een van de lastigste onderwerpen voor webdesigners was traditioneel het verticaal uitlijnen van elementen. De ruimte boven en onder het element moeten dan precies gelijk zijn.

In Bootstrap 4 is dat een stuk eenvoudiger geworden omdat Flexbox dat standaard ondersteunt.

De klassen die je hiervoor gebruikt zijn `.row` (de flex-container), en `align-items-*`

Meer informatie daarover:

<https://getbootstrap.com/docs/4.0/utilities/flex/#align-items>



```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>...
8    </head>
9    <body>
10     <div class="container">
11       <div class="row">
12         <div class="d-flex align-items-start">...</div>
13         <div class="d-flex align-items-end">...</div>
14         <div class="d-flex align-items-center">...</div>
15         <div class="d-flex align-items-baseline">...</div>
16         <div class="d-flex align-items-stretch">...</div>
17       </div>
18     </div>
19   </body>
20 </html>
```

OPDRACHT

zoek de patronen

Bouw nu eens na:

