

# DISPLAY NONE

FRONTEND DEVELOPMENT LES 4

# VORIGE WEEK

- Selectors deel 2
- Selectors in JavaScript
- Flexbox

Even heel kort samenvatten waar we het ook al weer over hebben gehad vorige week. Niet te lang bij stilstaan want er is heel veel materiaal te behandelen.

# BOX MODEL

# CSS IS GEK

```
aside {  
  width: 300px;  
  padding-left: 25px;  
  padding-right: 25px;  
  border: 1px solid black;  
}
```

Hoe breed is dit ding? Iedereen vindt dat de aside in z'n geheel gewoon 300px breed zou moeten zijn, en dat \*de content\* in de aside  $300 - 50 - 2 = 248$ px breed zou moeten zijn. Maar zo niet de opstellers van de spec. Die vonden dat de content 300px breed is, en dat padding en border er bij op worden geteld.

# HET MEDICIJN TEGEN DEZE GEKTE

```
aside {  
  box-sizing: border-box;  
  width: 300px;  
  padding-left: 25px;  
  padding-right: 25px;  
  border: 1px solid black;  
}
```

Nu doet het wel normaal.

# TIP

```
* {  
  box-sizing: border-box;  
}
```

Nu doet alles normaal. En nee, dit is geen aanslag op de performance.

# FLOAT

## The Greek Time Clock

**W**hen you ask somebody in Greece to wait, y wait two minutes. The interpretation of these minutes in, for instance, The Netherlands. In The l minutes will be exactly two minutes, or a bit less. I anything from five seconds to two hours, or more. say you'll meet at seven, nobody is really surprised much more in Greece than here in The Netherlands

```
h1 + p::first-letter {  
  font-size: 3em;  
  float: left;
```

Iedereen zou moeten kunnen verzinnen wat dit betekent, maar voor de zekerheid: de eerste letter in elke paragraaf die direct na een h1 staat moet heel groot zijn en links floaten.

# FLOAT CLEAREN

## The Greek Time Clock

**W**hen you ask somebody in Greece to wait, you know that two minutes in Greece is different than two minutes in, for example, The Netherlands. In Greece, it's exactly two minutes, or a bit less. In Greece, it's well. That's interesting.

The perception of time in Greece is different. If you say you'll meet me here in The Netherlands. I want to enjoy time here as well.

Soms wil je dat een heading altijd onder een float begint.



# FLOAT CLEAREN

## The Greek Time Clock

**W**hen you ask somebody in Greece how long it takes to get to the airport, the answer is different. In Greece is different than two minutes, or exactly two minutes, or a bit less.

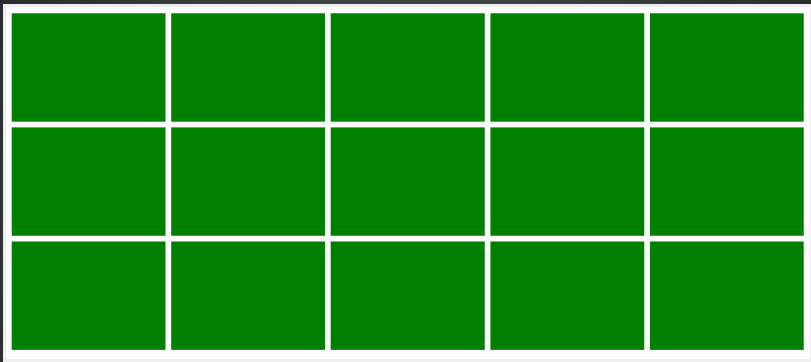
**Well. That's interesting**

*The perception of time in Greece is different. If you say*

```
h2 {  
  clear: left;  
}
```

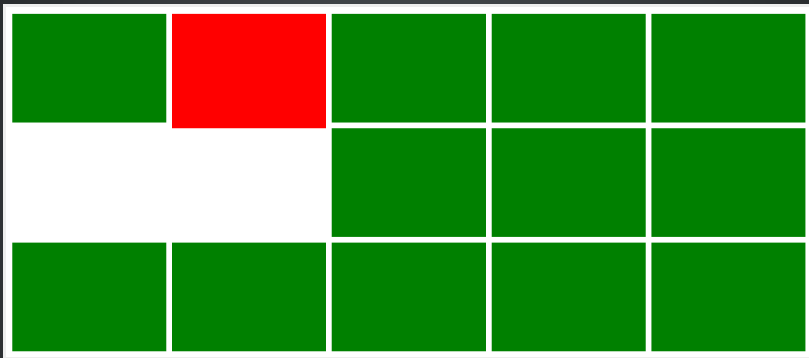
Soms wil je dat een heading altijd onder een float begint.

# FLOAT VOOR LAYOUT



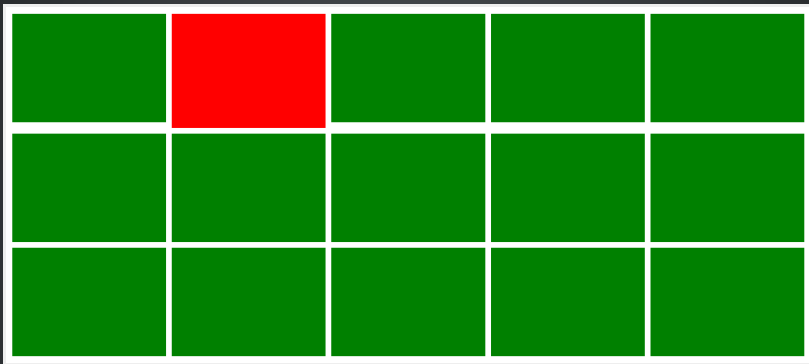
Werkt prima zolang alles even hoog is

# FLOAT VOOR LAYOUT



Wordt een zootje als dat niet zo is

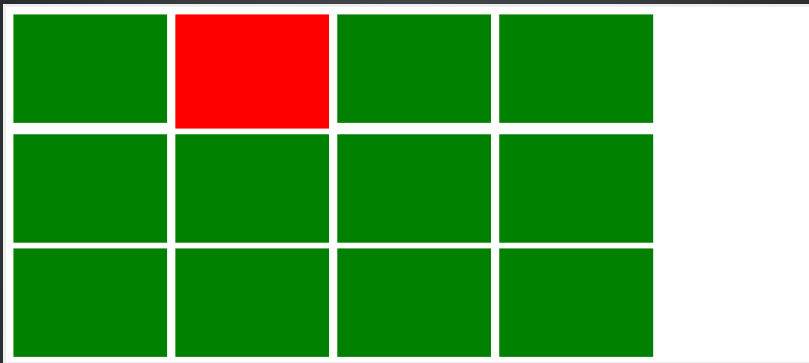
# FLOAT VOOR LAYOUT



```
div:nth-child(5n + 6) {  
  clear: left;  
}
```

Ingewikkeld gedoe om dat op te lossen.

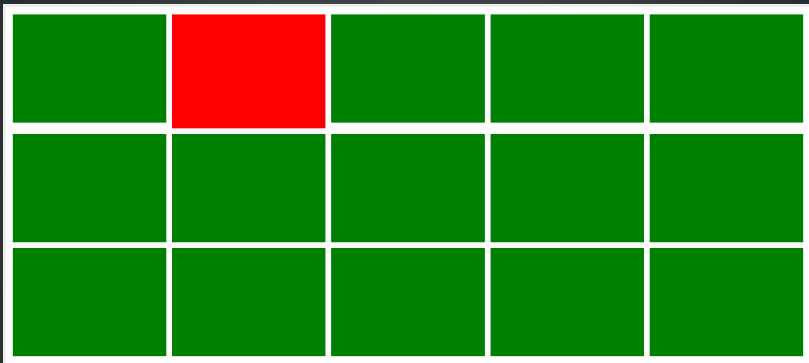
# DISPLAY: INLINE-BLOCK;



```
div {  
  display: inline-block;
```

Theoretisch zou display: inline-block; lekker simpel moeten zijn. Maar helaas laat het de whitespace tussen elementen altijd zien. Wat op zich logisch is, maar ook erg irritant.

# DISPLAY: INLINE-BLOCK;



```
<div></div><!--  
--><div></div>
```

Het is te fixen door geen white-space te gebruiken, of door white-space uit te commenten. Allebei niet echt tof. Fijn dat we dit niet meer hoeven te gebruiken.

# POSITIONING

- Relative
- Absolute
- Fixed
- Static

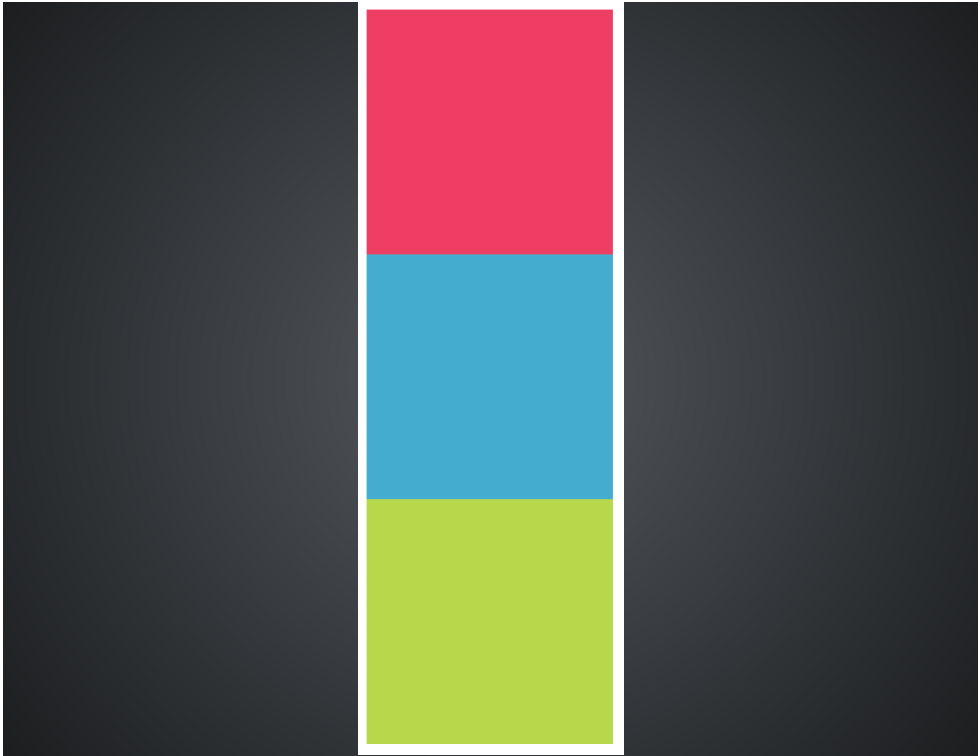
Er zijn vier values voor position.

# POSITION:STATIC

The box is a normal box, laid out according to the normal flow.  
The 'top', 'right', 'bottom', and 'left' properties do not apply.

De standaard waarde van position is static. Het doet niet zo veel en je kan er niet veel mee. Elementen die in een static element staan die bijvoorbeeld absoluut gepositioneerd zijn en gebruik maken van top en left negeren deze parent en positioneren zich tov de eerstvolgende gepositioneerde parent in de DOM tree

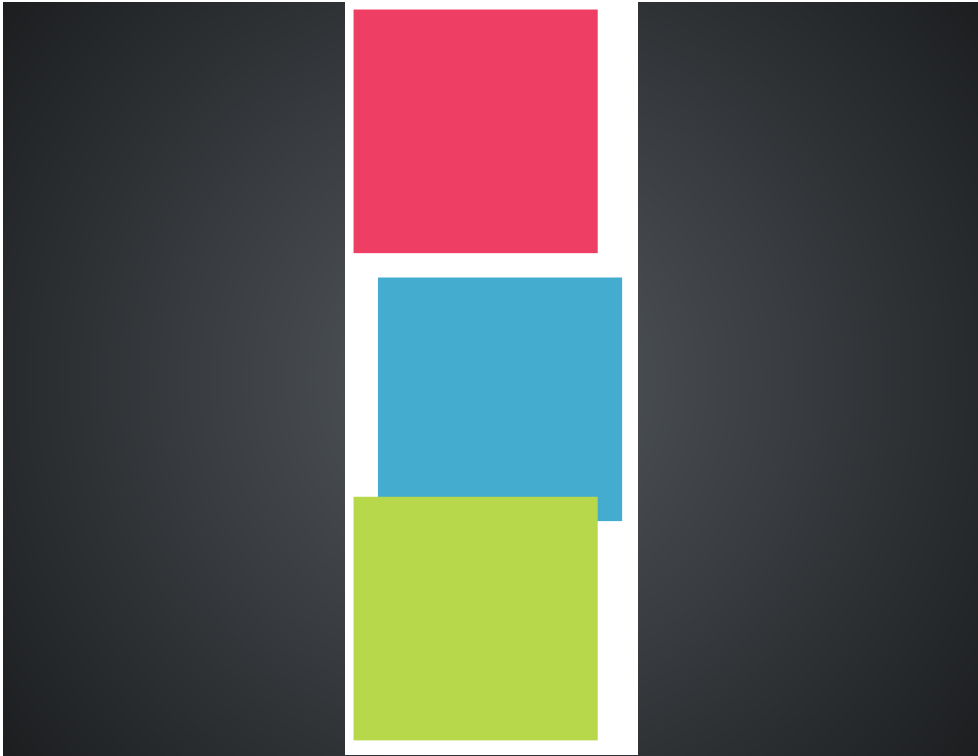




# POSITION:RELATIVE

The box's position is calculated according to the normal flow (this is called the position in normal flow). Then the box is offset relative to its normal position. When a box B is relatively positioned, the position of the following box is calculated as though B were not offset.

Position:relative betekent dat je een element left, right, top en bottom properties mee kan geven. Als je dat doet, dan denken volgende elementen dat je dat niet hebt gedaan. Echt! Zie volgende slide. Z-index werkt nu ook.



# POSITION:ABSOLUTE

The box's position (and possibly size) is specified with the 'top', 'right', 'bottom', and 'left' properties. These properties specify offsets with respect to the box's containing block. Absolutely positioned boxes are taken out of the normal flow. This means they have no impact on the layout of later siblings.

Position:absolute haalt iets uit de flow. Het heeft geen invloed meer op de dingen er omheen. De offset is gerelateerd aan z'n containing block: de eerste parent met position:relative, position:absolute of position:fixed; Een parent met position static wordt genegeerd.

# POSITION:FIXED

The box's position is calculated according to the 'absolute' model, but in addition, the box is fixed with respect to the viewport.



# UNITS VOOR LAYOUT

- em
- rem
- viewport relative units

Er zijn een paar mooie eenheden die je kunt gebruiken voor het definiëren van layout. Hieronder hun usecases

# EM

voor maximale en minimale breedtes

Gebruik deze site om een ideale breedte van een tekstcolom te bepalen:  
<http://nerd.vasilis.nl/code/measure-help/>

# IDEALE MEASURE

*Anything from 45 to 75 characters is widely regarded as a satisfactory length of line...*

Robert Bringhurst

*a column is easy to read if it's wide enough to accommodate an average of 10 words per line*

Müller-Brockmann

Gebruik deze site om een ideale breedte van een tekstcolom te bepalen:  
<http://nerd.vasilis.nl/code/measure-help/>



# REM

perfect voor margins en paddings en border-radii

Altijd gerelateerd aan de grootte van de font-size van het html-element. Voor typografie niet flexibel genoeg, maar dat gebrek aan flexibiliteit is juist weer perfect voor margins en paddings, die gelijk kunnen blijven, ongeacht de font-grootte van het element.

# VIEWPORT RELATIVE UNITS!

FTW!

FTW!

FTW!

Units die gerelateerd zijn aan de viewport. vw = viewport width. vh = viewport height. vmin = de kleinste van de width of de height. vmax = de grootste van die twee. Je kan dit voorbeeld laten zien: <http://nerd.vasilis.nl/code/tschicholds-figures/responsive/> of deze gekke waarbij de units voor font-size worden gebruikt: <http://massivetangent.co.uk/> of deze waarbij je er voor zorgt dat er altijd een stuk van de caption zichtbaar is onder de foto: <http://vasilis.nl/voto/boe/140927132517/> Hier is niks bij te zien als je de viewport niet groter en kleiner maakt.