

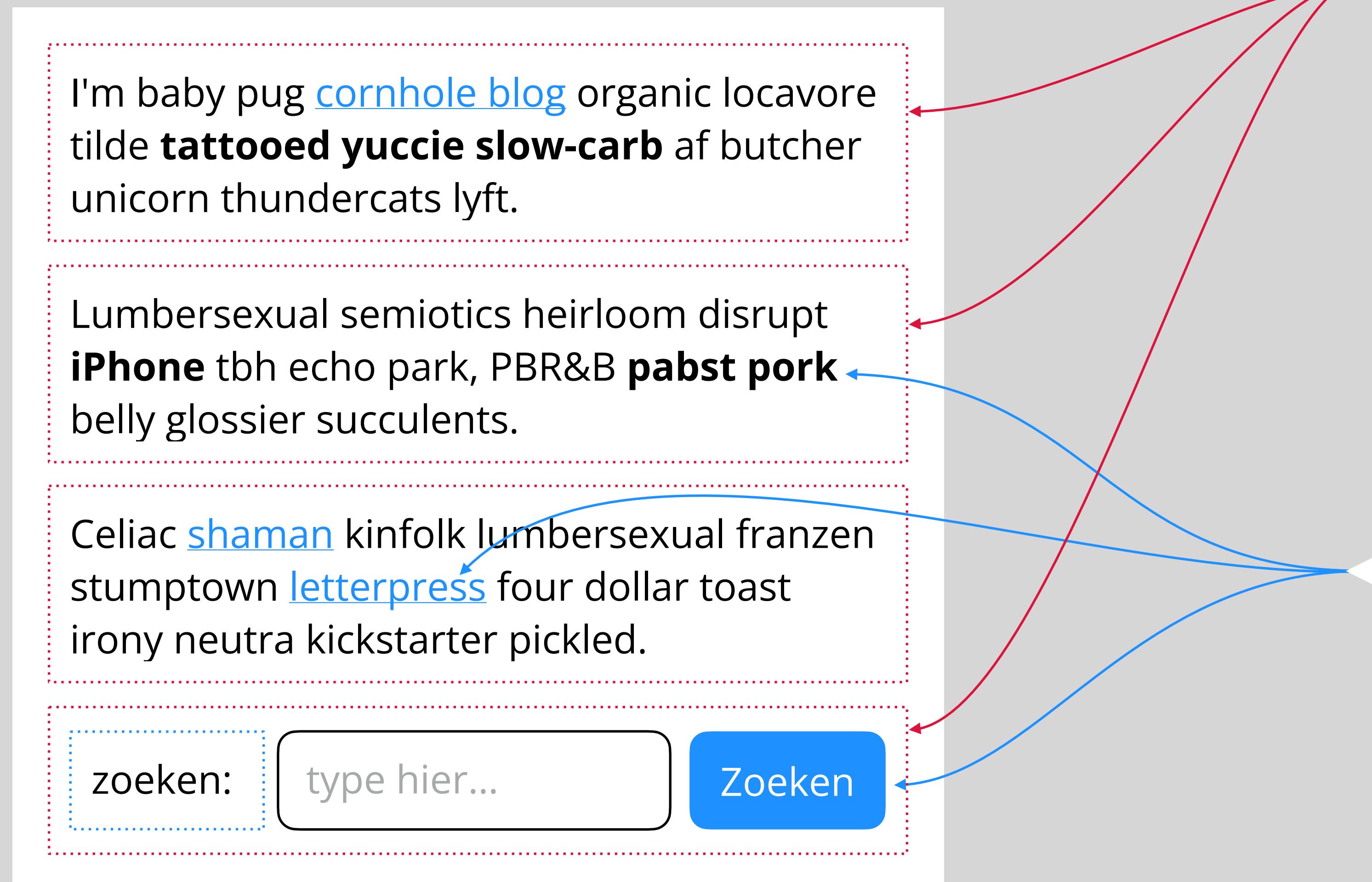
FED 24/25

# 5 Flavours of positioneren



# Standaard: flow layout

[3 paragrafen met links en strongs en een formpje](#)



## Block elements

- Staan **onder elkaar**. Elk op een eigen regel.
- Zo **breed** als de parent.
- Zo **hoog** als de content in het element.

Voorbeelden:

- Elementen voor organisatie: header, main, footer, section, form, fieldset, table, div.
- Content: p, h1, h2, h3, blokquote, ul/ol, li.

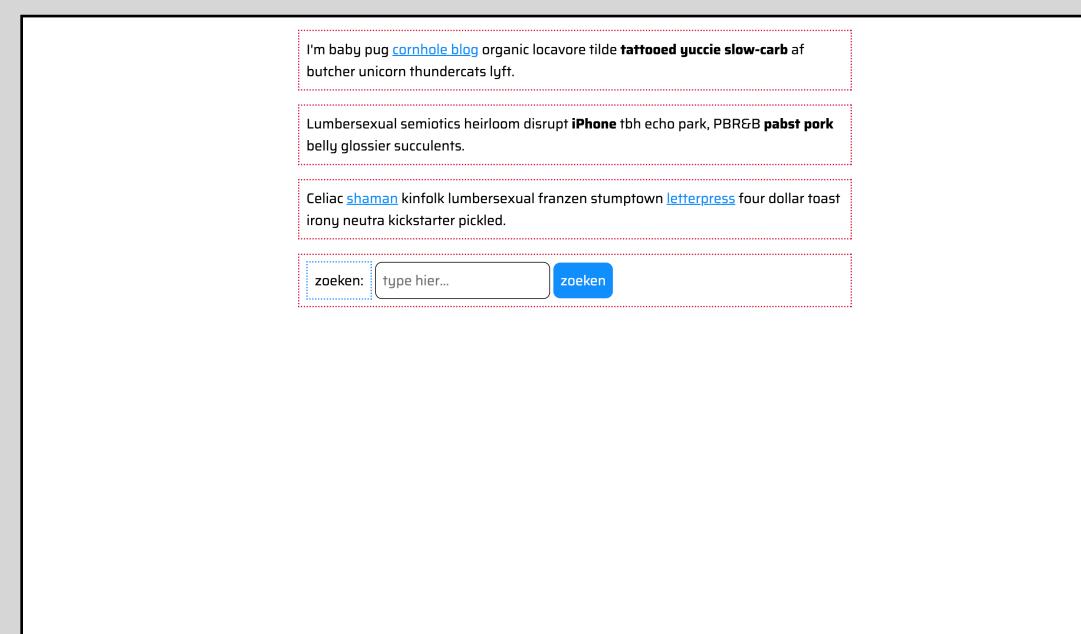
## Inline elements

- Staan **naast elkaar** als dat past (en anders op de volgende regel. 'inline' met de content).
- Zo breed als de content in het element.
- Zo hoog als de content in het element.

Voorbeelden:

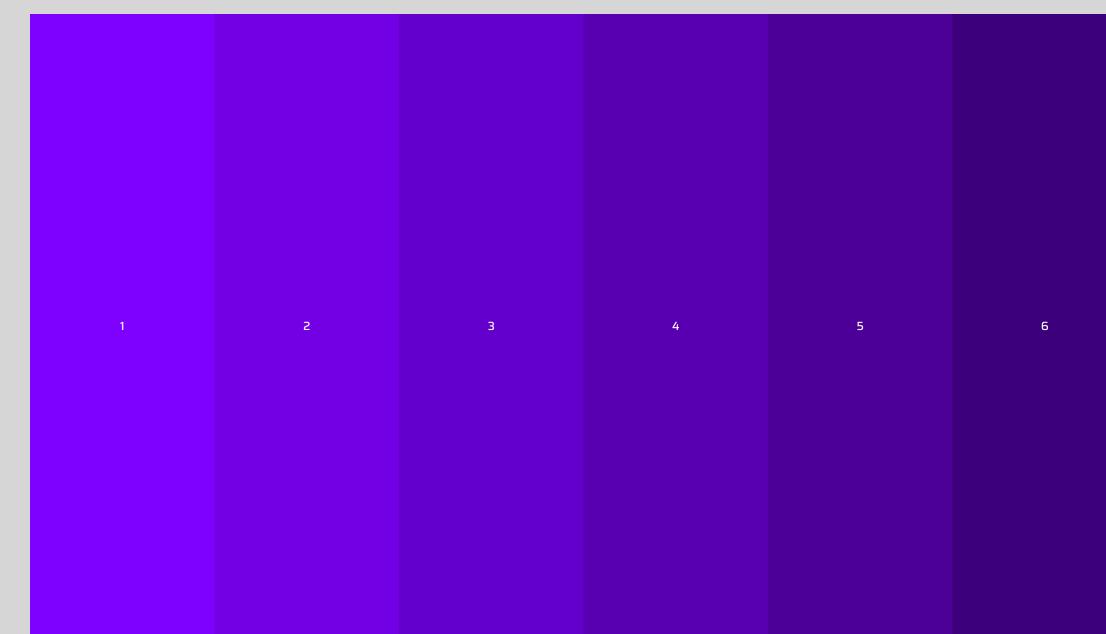
- Text-semantiek: a, strong, em, sup, sub, span.
- Formulier-elementen: input, label, button.
- Afbeeldingen

# Layout overview



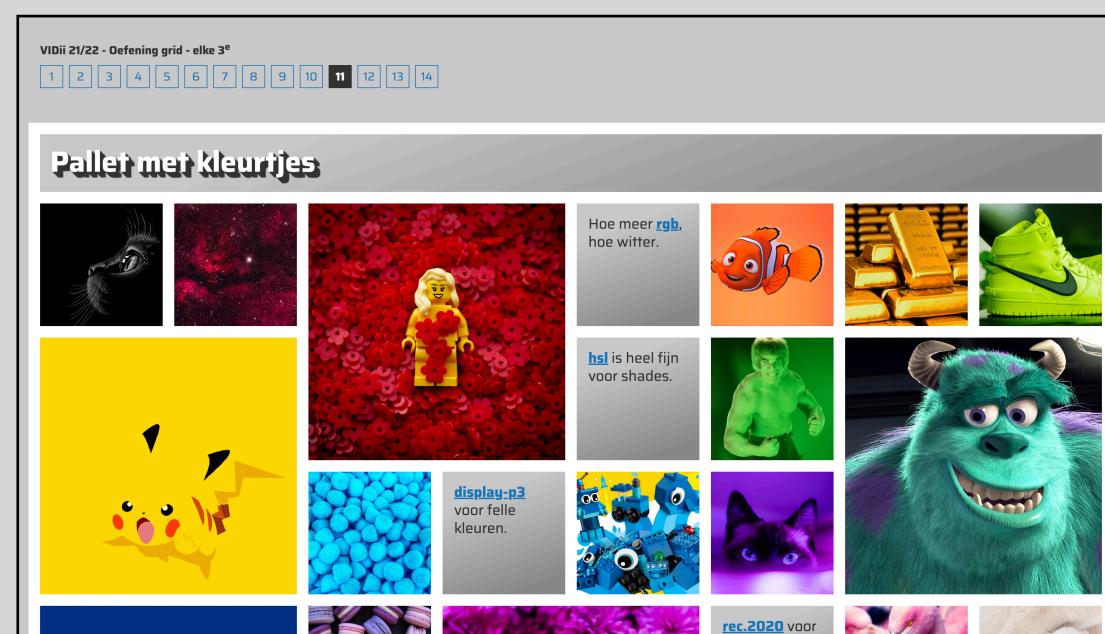
## Standaard: flow layout

Block-elementen onder elkaar  
Inline-elementen naast elkaar



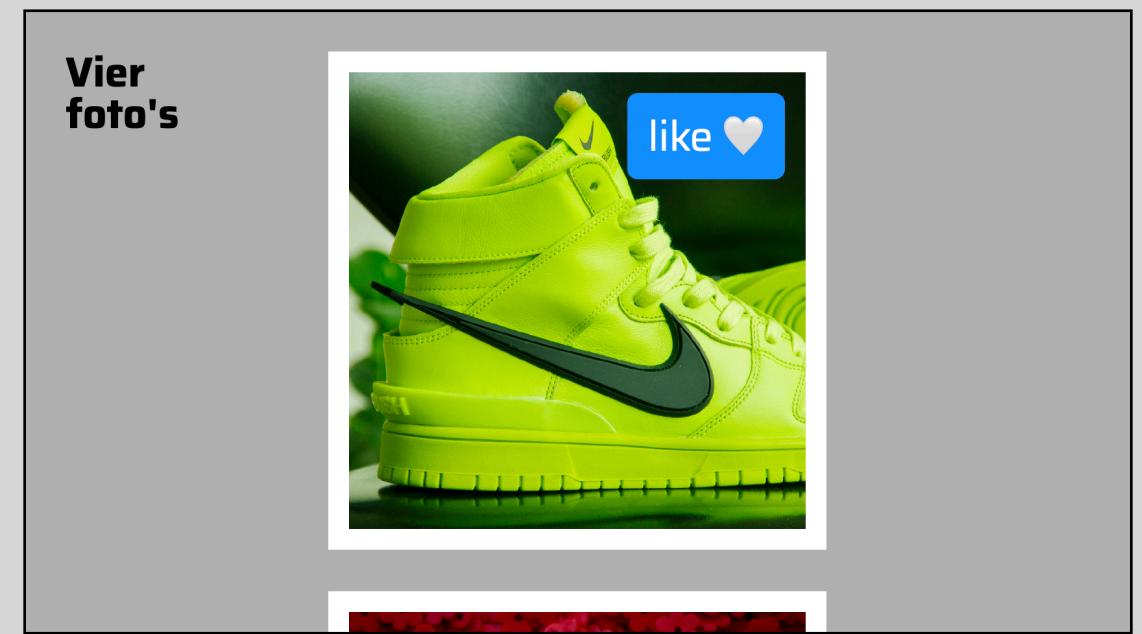
## Display flex

Elementen naast elkaar zetten. De kinderen vullen de parent flexibel (over meer rijen indien nodig).



## Display grid

Voor de parent (in twee dimensies) een grid definiëren. De kinderen vullen de gridcellen.



## Position: absolute en fixed

Elementen op een specifieke plek in een parent of in het scherm zetten.

Op CSS Day 2022 heeft [Michele Barker](#) een [presentatie over CSS layout](#) gegeven. Ze gaat o.a. in op wanneer je het best flexbox, grid of andere CSS technieken kunt gebruiken. Warning - het niveau wordt **zwart**.

# 5 Flavours of position

MDN: The **position CSS property** sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

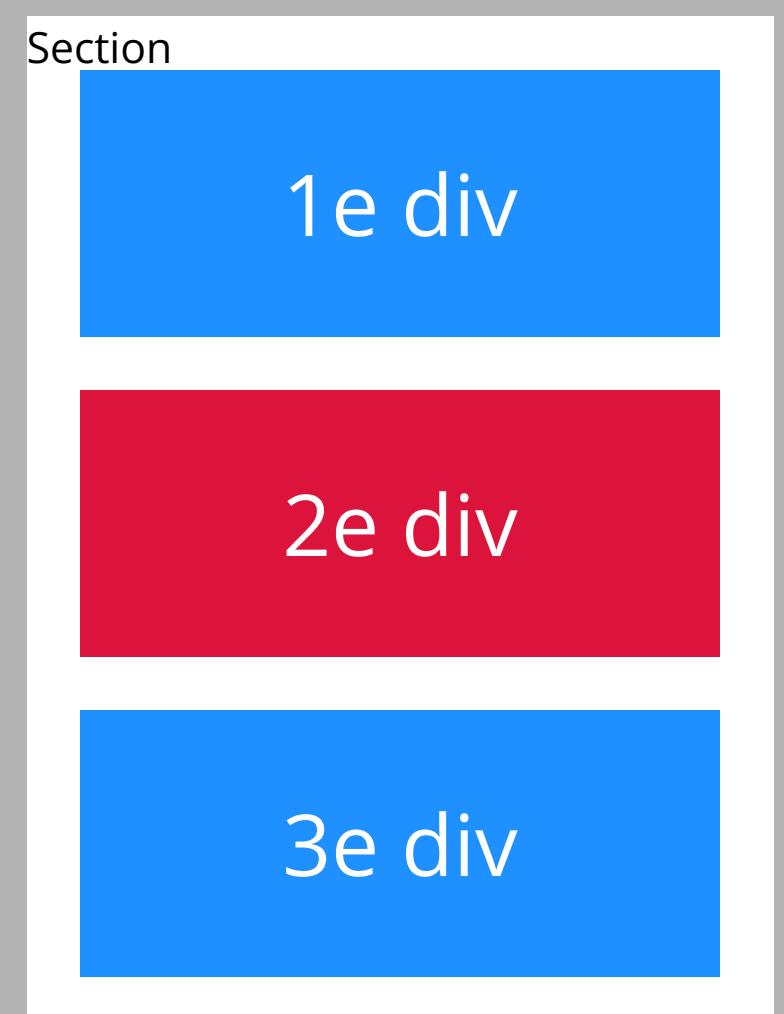
Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jmI0UIXU](#)

## position: static

sample: [codepen.io/shoof/pen/oNrqRmj](#)

De standaard waarde voor **position** is **static**. De positie van een element wordt gebaseerd op de plaats in HTML. Verder niets bijzonders. Dit wordt ook wel flow lay-out genoemd.



```
div:nth-of-type(2) {  
  position: static;  
}
```

Nb. Static is de standaard waarde voor position. Die hoef je dus niet op te geven. Dat is mooi, want de meeste elementen moeten gewoon op de plek staan waar ze staan.

# 5 Flavours of position

MDN: The **position** CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jmI0UIXU](#)



Meestal kun je beter **translate** gebruiken

## position: relative

sample: [codepen.io/shoof/pen/bGPvyza](#)

Met **position relative** behoudt een element zijn plek in de flow. Vanaf die plek kun je het element met top, right, bottom en left verplaatsen.



```
div:nth-of-type(2) {  
  position: relative;  
  top: 1.25em;  
  left: 1.25em;  
}
```

Nb. Daar wordt position relative eigenlijk niet meer voor gebruikt.  
Dat kun je beter met translate doen.

Nb. Position relative wordt nog wel veel gebruikt als basis om kinderen in het element te kunnen positioneren (zie volgende slide).

# 5 Flavours of position

MDN: The **position** CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jmI0UIXU](#)

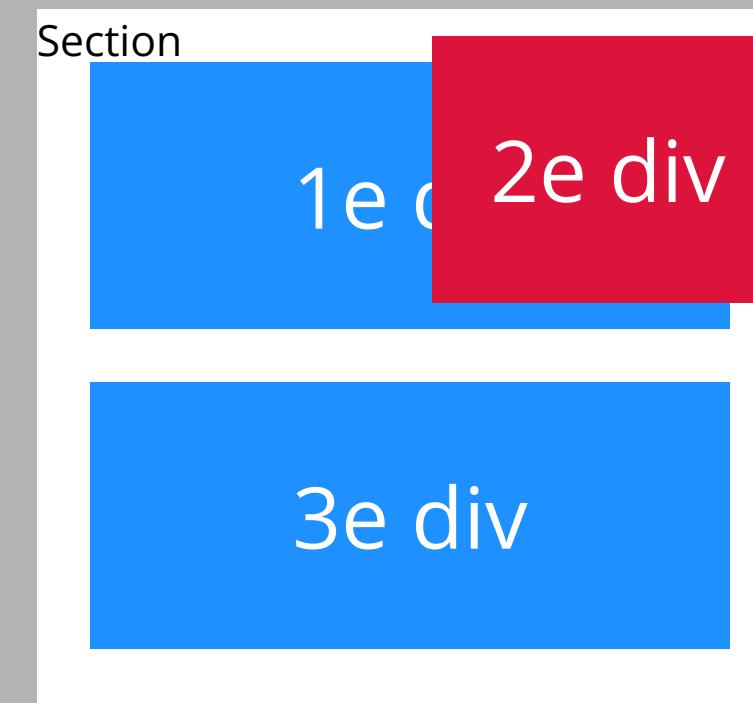


Meestal kun je beter **flex** of **grid** gebruiken om te lay-outen.

## position: absolute

sample: [codepen.io/shoof/pen/gONeNaG](#)

Met **position absolute** verliest een element zijn plek in de flow. Het element wordt met top, right, bottom en left geïnjecteerd in de eerste parent (die zelf geïnjecteerd is).



Nb. Het element krijgt de breedte die de content inneemt (ook block elementen).

```
<!-- parent -->
section {
    <!-- geïnjecteerd,
    childen worden hierdoor
    binnen het element
    geïnjecteerd -->
    position: relative;
}

<!-- child -->
div:nth-of-type(2) {
    position: absolute;
    top: .75em;
    right: 0em;
}
```

# 5 Flavours of position

MDN: The **position** CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDN](#)

Filmpje: [youtu.be/jx5jmI0UIXU](#)

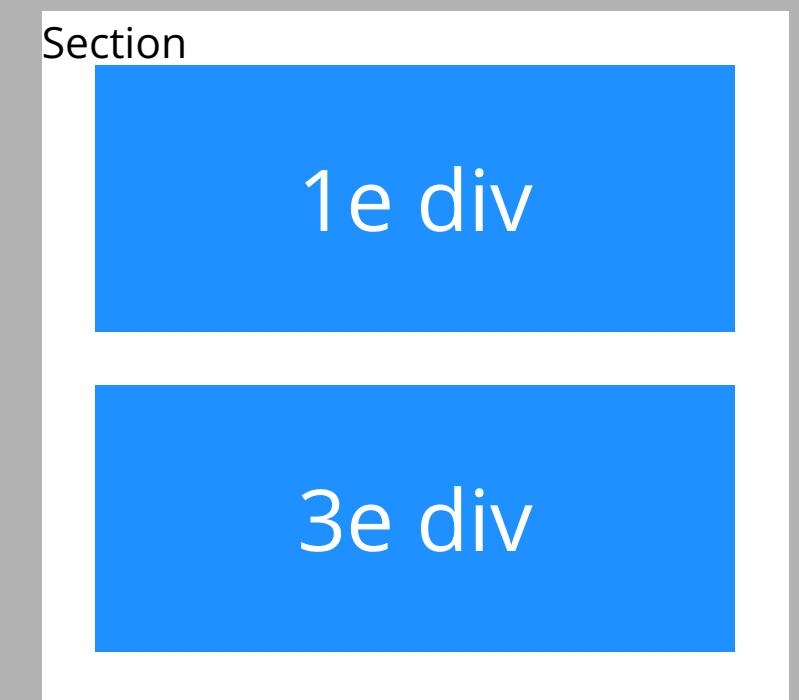


Meestal kun je beter **flex** of **grid** gebruiken om te lay-outen.

## position: fixed

sample: [codepen.io/shoof/pen/eYwMwpa](#)

Met **position fixed** verliest een element zijn plek in de flow. Het element wordt met top, right, bottom en left geïnitialiseerd in de viewport. Het element blijft op die vaste plek staan als de gebruiker scrollt (en neemt dus altijd ruimte in).



```
div:nth-of-type(2) {  
  position: fixed;  
  bottom: .75em;  
  right: .75em;  
}
```

Nb. Het element krijgt de breedte die de content inneemt (ook block elementen).

2e div

# 5 Flavours of position

MDN: The **position** CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

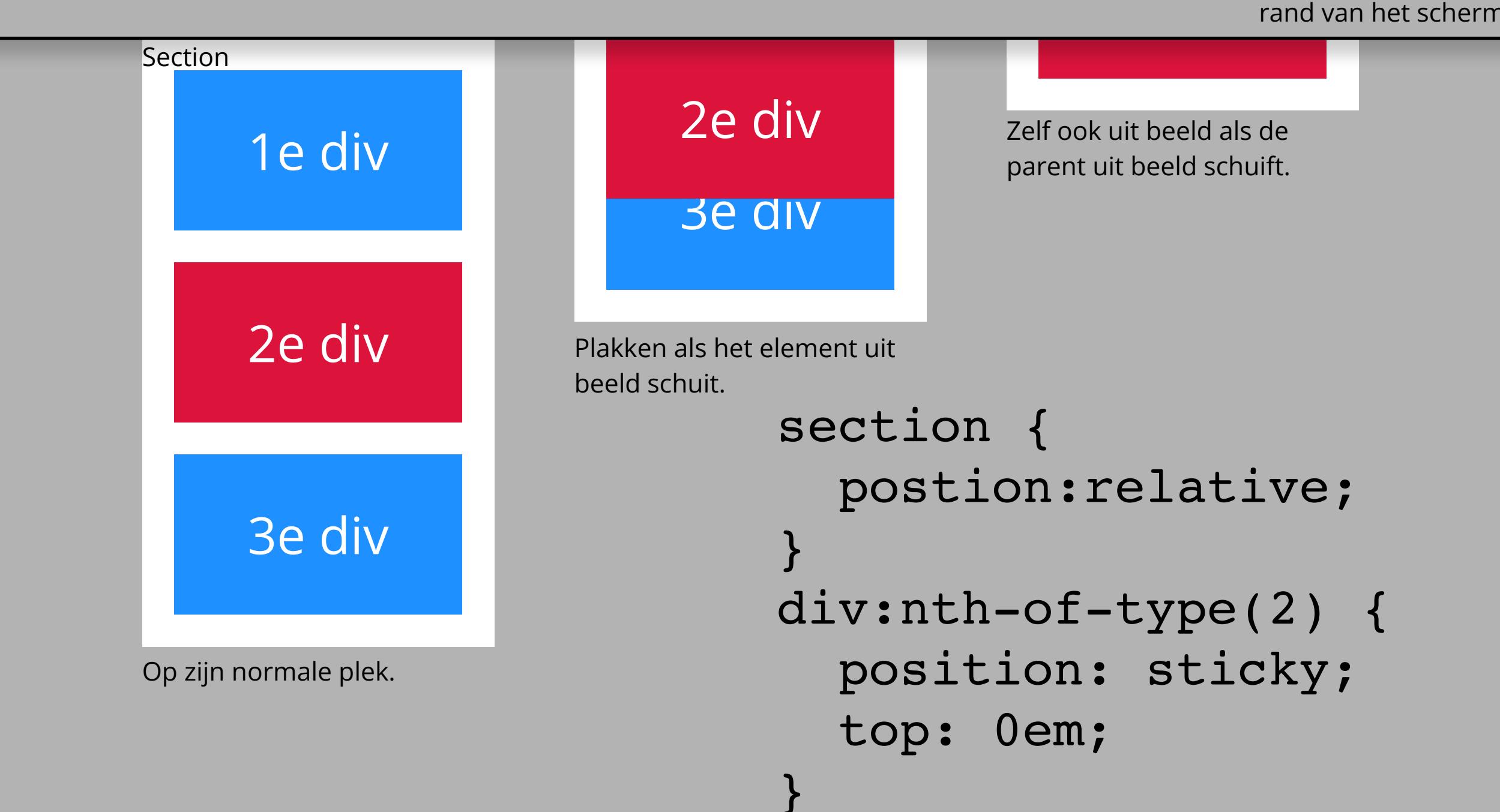
Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jmI0UIXU](#)

## position: sticky

sample: [codepen.io/shoof/pen/vYqRqLx](#)

Met **position sticky** behoudt een element zijn plek in de flow. Maar op het moment dat het element uit beeld schuift blijft het element *plakken* aan de rand (tot de parent helemaal uit beeld is verdwenen).



**POSITIONS  
2 ΕΙΓΛΟΥΣ ΟΤ  
ΕΕΔ ΣΑΙΣΡ**

