

Frontend leerlijn

# HTML & CSS

---

## Les 2

# Planning

14:00 – 14:50

- CSS selectoren en regels
- Google developer tools
- Layouts maken met flexbox

15:00 – 15:30

- Flexbox continued
- Huiswerk volgende les

15:30 – 17:00

- *Lesopdracht: Flexbox*

# Aanvullende workshops Git

Ingeroosterd in het Studenten Team > SD Software Development

**Deel 1**

**18  
mei**

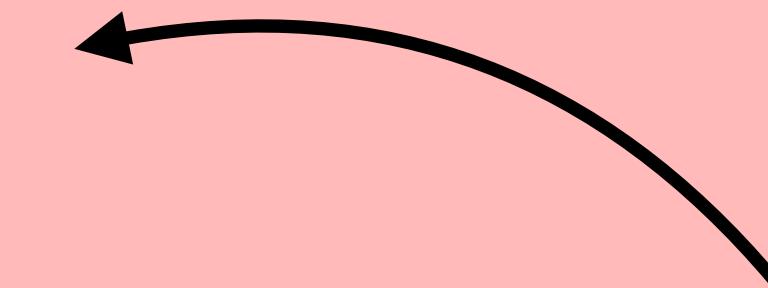
Repositories maken, commit's en pushen

**Deel 2**

**25  
mei**

Branches en Pull Requests

Je hebt deze kennis nodig om vanaf volgende week je huiswerk in te kunnen leveren!!!



Deel 2 vereist dat je de kennis van deel 1 beheerst!

# css

De basisbeginselen



# Hoe CSS werkt

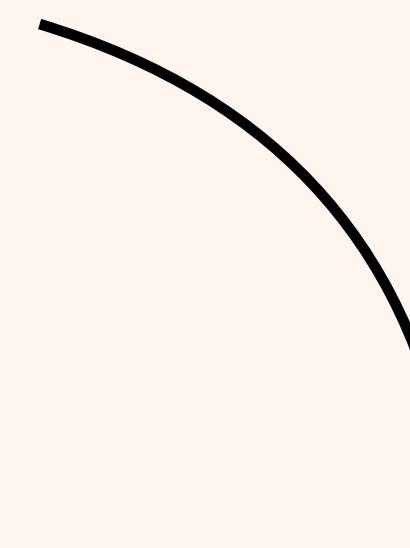
CSS maakt associaties tussen stijl regels en HTML elementen

<p>Lorem ipsum dolor</p>

# Hoe CSS werkt

CSS maakt associaties tussen stijl regels en HTML elementen

“Maak de tekstkleur blauw!”



<p>Lorem ipsum dolor</p>

SELECTOR



p {  
}

color: blue;



DECLARATION

```
h1, h2 {  
    font-family: Arial;  
    color: blue;  
}
```

PROPERTY

VALUE

# Meest voorkomende selectors

Universal selector	<code>* { ... }</code>	ALLES (alleen gebruiken voor CSS reset)
Type selector	<code>h1, h2 { ... }</code>	<code>&lt;h1&gt; &lt;/h1&gt;</code> <code>&lt;h2&gt; &lt;/h2&gt;</code>
Class selector	<code>.note { ... }</code>  <code>p.note { ... }</code>	<code>&lt;h3 class="note"&gt;</code>  <code>&lt;p class="note"&gt;</code>
ID selector	<code>#introduction { ... }</code>	<code>&lt;span id="introduction"&gt;</code>

# Vraag

Typ je antwoord in de chat (1 min)

“Wanneer is het handig  
om een **type selector**  
te gebruiken?”



Direct child selector	<code>li &gt; a { ... }</code>	<code>&lt;li&gt;   &lt;a href="/"&gt;link&lt;/a&gt; &lt;/li&gt;</code>  <code>&lt;a href="/"&gt;link&lt;/a&gt;</code>
Descendant selector	<code>p a { ... }</code>	<code>&lt;p&gt;   &lt;span&gt;     &lt;a href="/"&gt;link&lt;/a&gt;   &lt;/span&gt; &lt;/p&gt;</code>
Adjacent sibling selector	<code>button + button { ... }</code>	<code>&lt;button&gt;Versturen&lt;/button&gt; &lt;button&gt;Cancel&lt;/button&gt;</code>
General sibling selector	<code>h1 ~ p { ... }</code>	<code>&lt;h1&gt; &lt;/h1&gt; &lt;p&gt; &lt;/p&gt; &lt;p&gt; &lt;/p&gt; &lt;p&gt; &lt;/p&gt;</code>

# Tijd voor een demo!

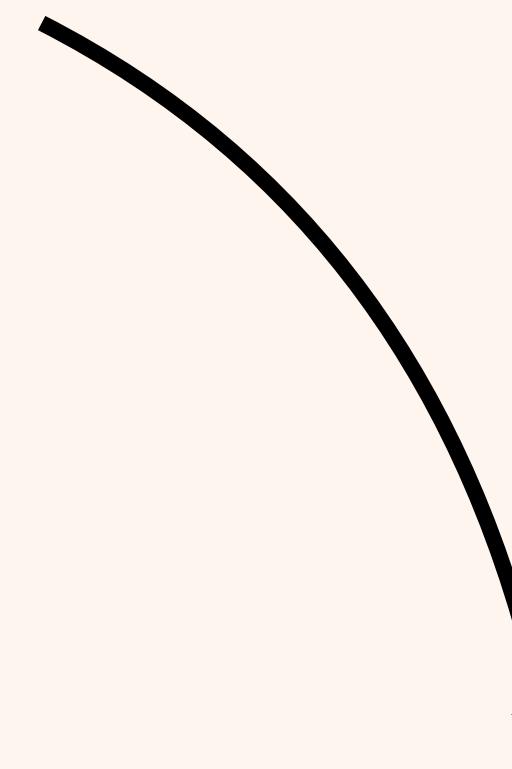
- CSS Reset
- Selectors groeperen
- Padding, margin en The Box Model
- Google Developer Tools

# Battle of the selectors

<p>Lorem ipsum dolor</p>

# Battle of the selectors

“Maak de tekstkleur blauw!”

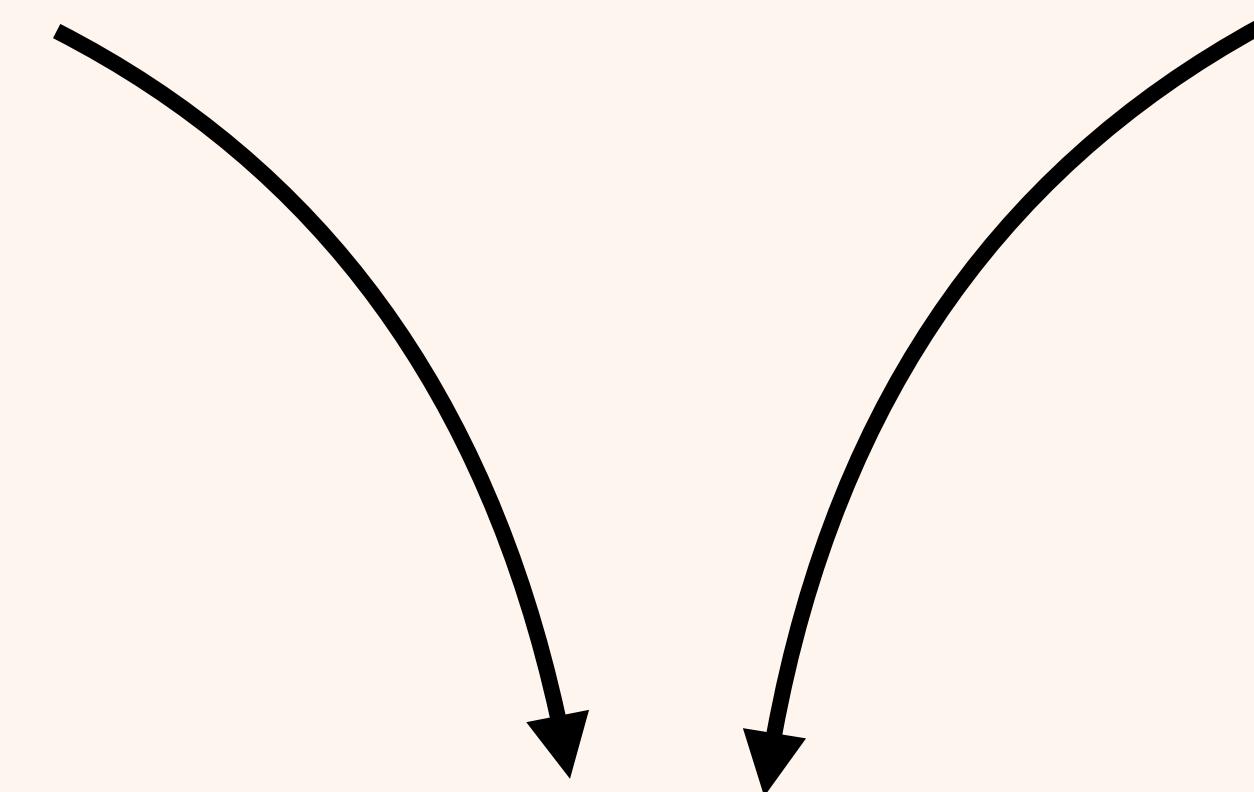


<p>Lorem ipsum dolor</p>

# Battle of the selectors

“Maak de tekstkleur blauw!”

“Maak de tekstkleur groen!”



<p>Lorem ipsum dolor</p>

# Battle of the selectors

“Maak de tekstkleur blauw!”

“Maak de tekstkleur groen!”

<p>Lorem ipsum dolor</p>



Wie wint 🤔

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}
```

```
i {  
    color: green;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}
```

```
i {  
    color: green;   
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}
```

```
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;  
}  
  
* {  
    color: green;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}
```

```
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;   
}  
  
* {  
    color: green;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}  
  
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;   
}  
  
* {  
    color: green;  
}  
  
p#intro {  
    color: blue;  
}  
p {  
    color: green;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}  
  
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;   
}  
  
* {  
    color: green;  
}  
  
p#intro {  
    color: blue;   
}  
p {  
    color: green;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}  
  
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;  
}  
  
* {  
    color: green;  
}  
  
p#intro {  
    color: blue;   
}  
  
p {  
    color: green;  
}
```

!important

```
p#intro {  
    color: blue;  
}  
  
p {  
    color: green !important;  
}
```

# Battle of the selectors

De laatste regel

```
i {  
    color: blue;  
}  
  
i {  
    color: green;   
}
```

De meest specifieke regel

```
i {  
    color: blue;  
}  
  
* {  
    color: green;  
}  
  
p#intro {  
    color: blue;   
}  
  
p {  
    color: green;  
}
```

!important

```
p#intro {  
    color: blue;  
}  
  
p {  
    color: green !important;  
} 
```

# Layouts

Met CSS Flexbox

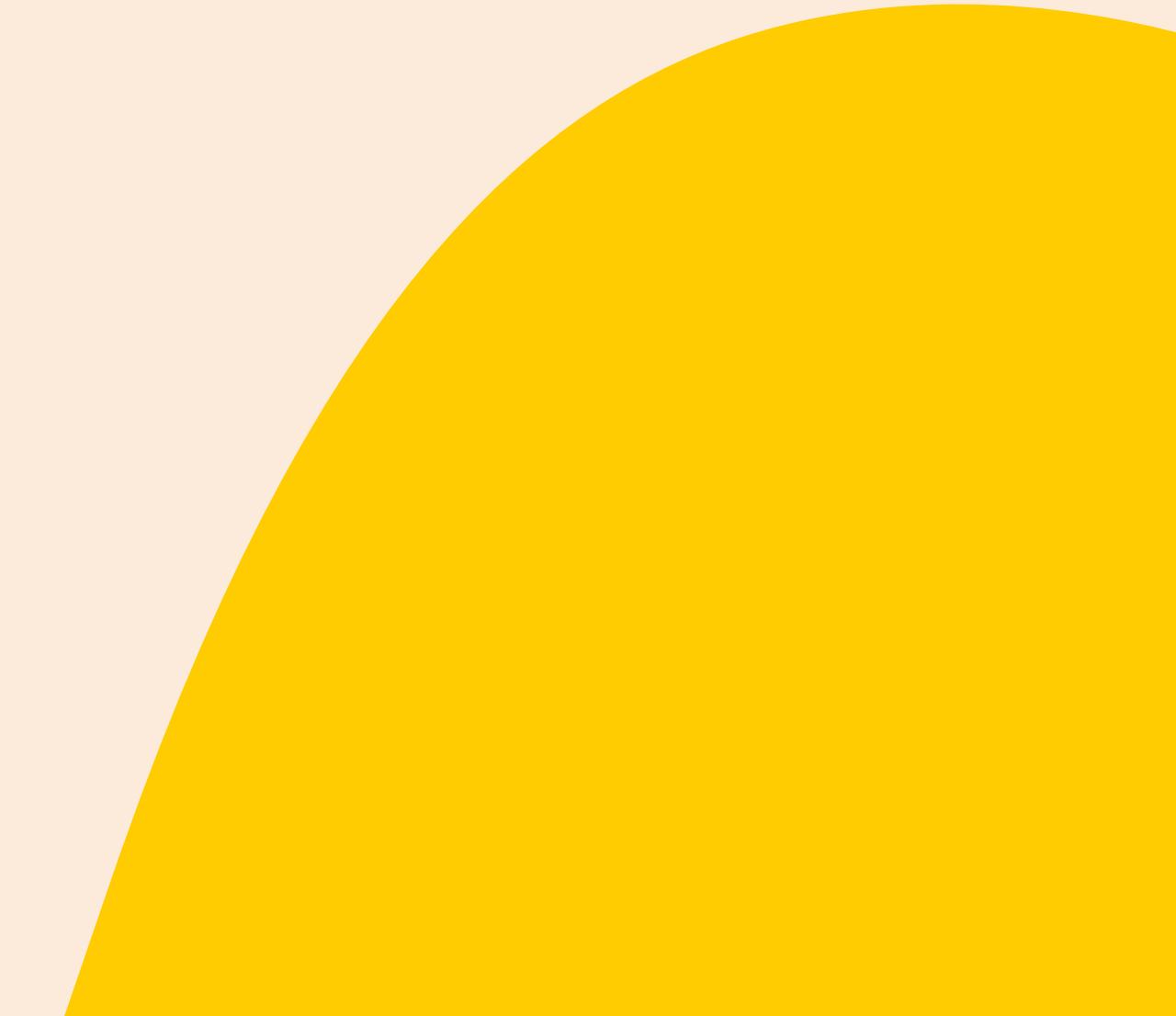


# Vroeger vs. Nu



# Vroeger vs. Nu

float: left;



# Vroeger vs. Nu



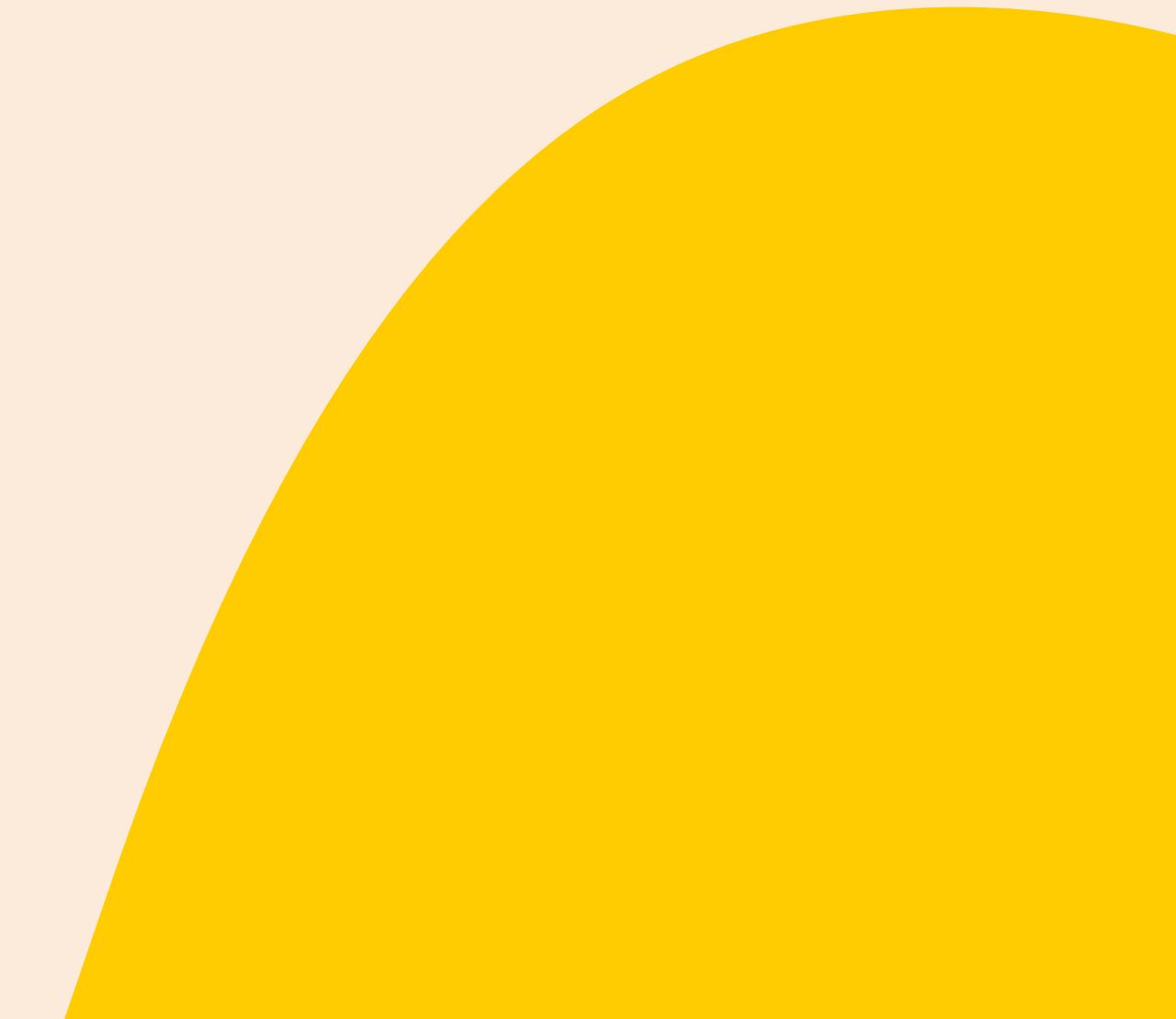
float: left;



# Vroeger vs. Nu



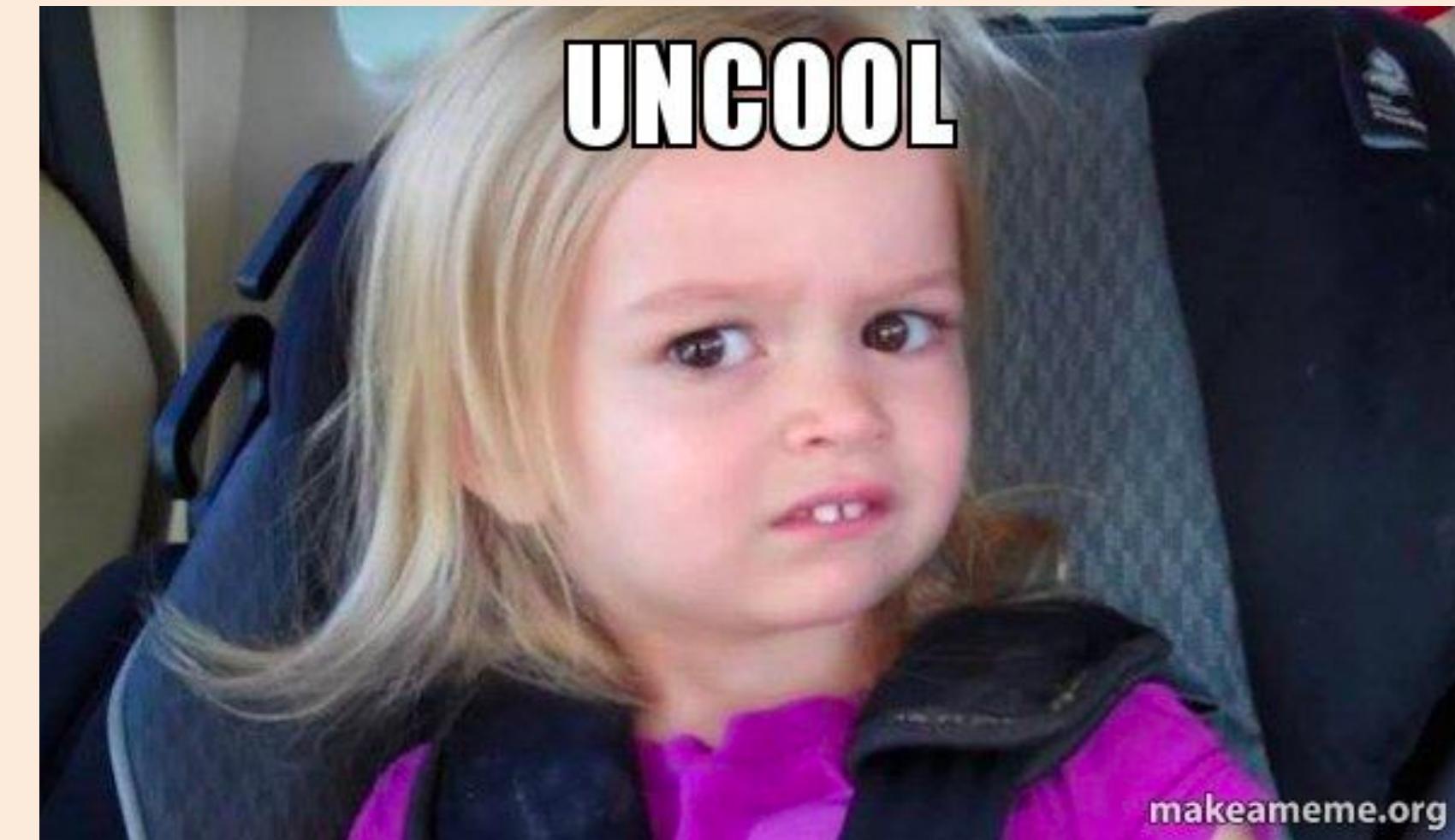
float: left;



# Vroeger vs. Nu



float: left;



# Vroeger vs. Nu



Flexbox

# Flexbox is... awesome!

- Flexbox elementen passen hun afmeting **automatisch** aan op basis van de beschikbare ruimte op de pagina

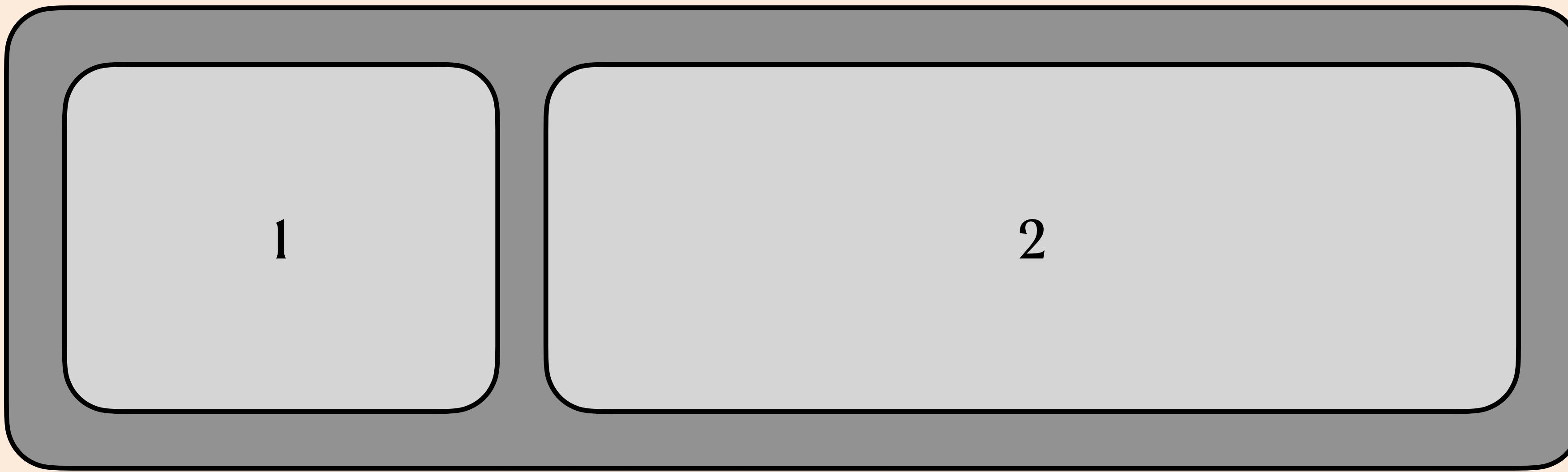
# Flexbox is... awesome!

- Flexbox elementen passen hun afmeting **automatisch** aan op basis van de beschikbare ruimte op de pagina
- Het is nu veel makkelijker om elementen te positioneren dan het vroeger was met  float 

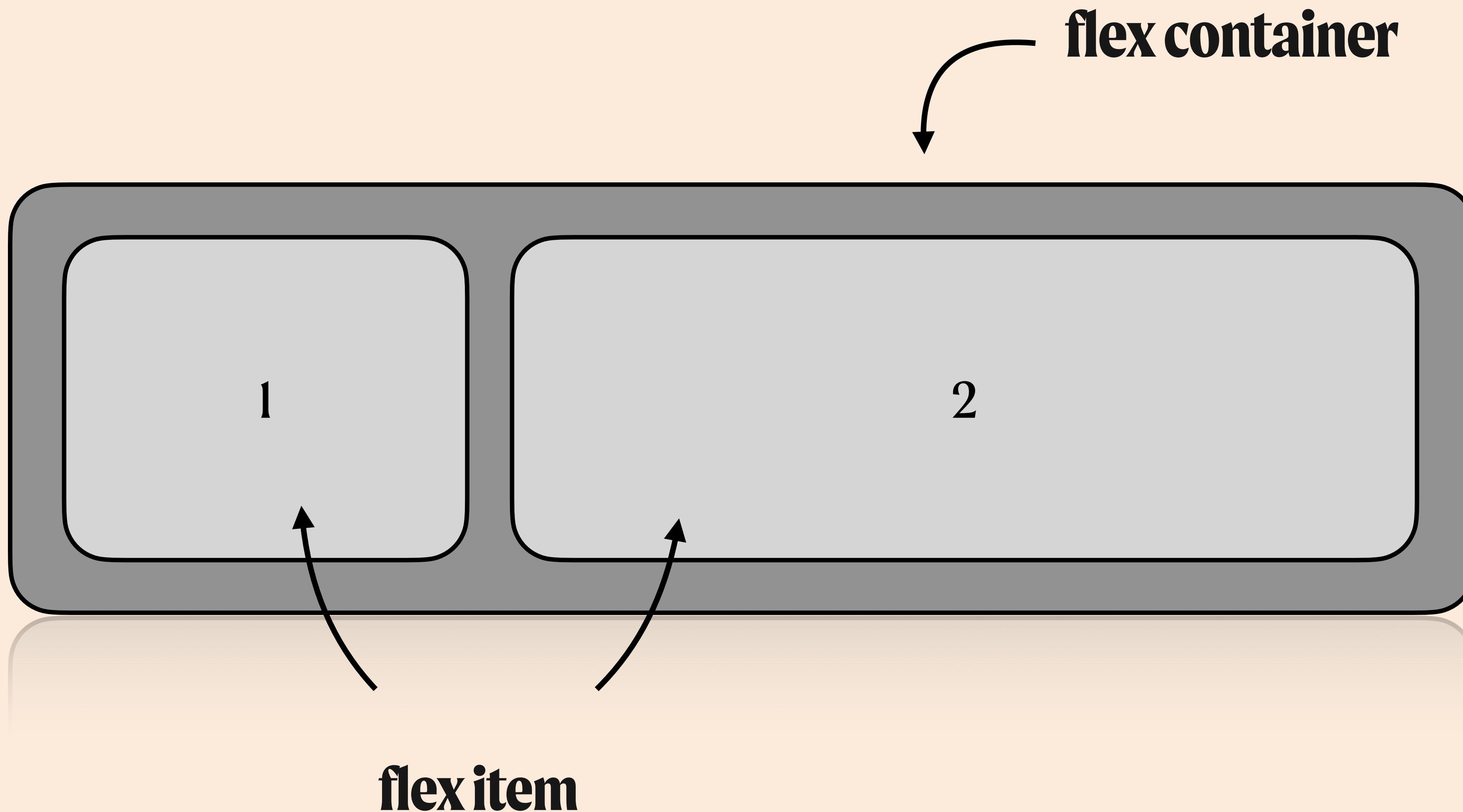
# Flexbox is... awesome!

- Flexbox elementen passen hun afmeting **automatisch** aan op basis van de beschikbare ruimte op de pagina
- Het is nu veel makkelijker om elementen te positioneren dan het vroeger was met  float 
- We kunnen de volgorde en weergave-richting van elementen aanpassen zonder aan de originele HTML te komen

# Flexbox is science

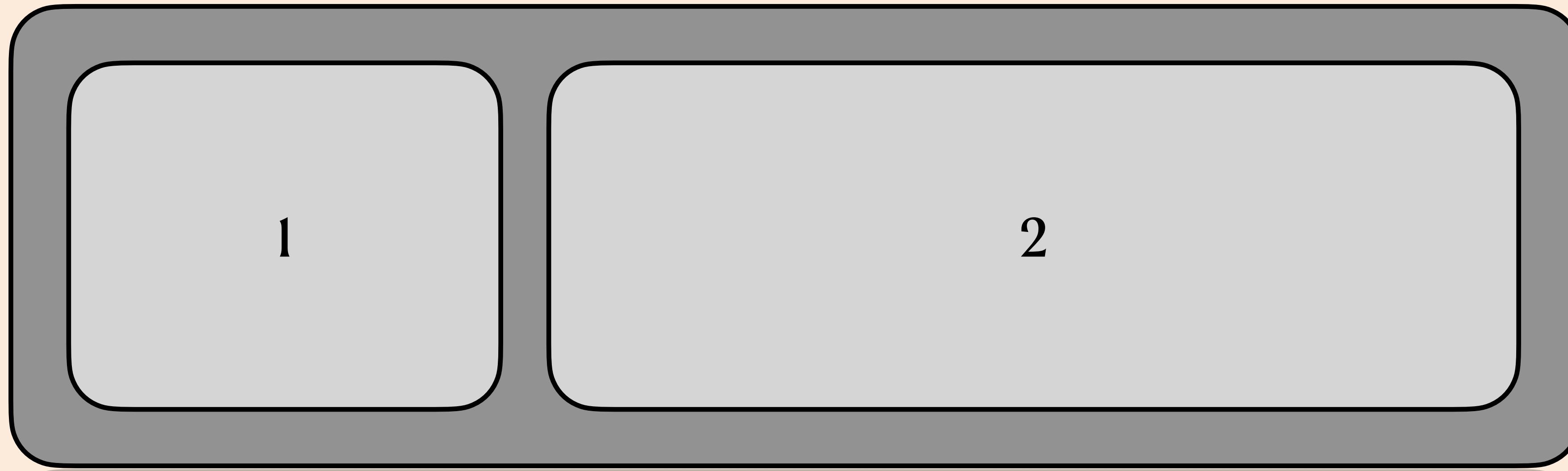


# Flexbox is science



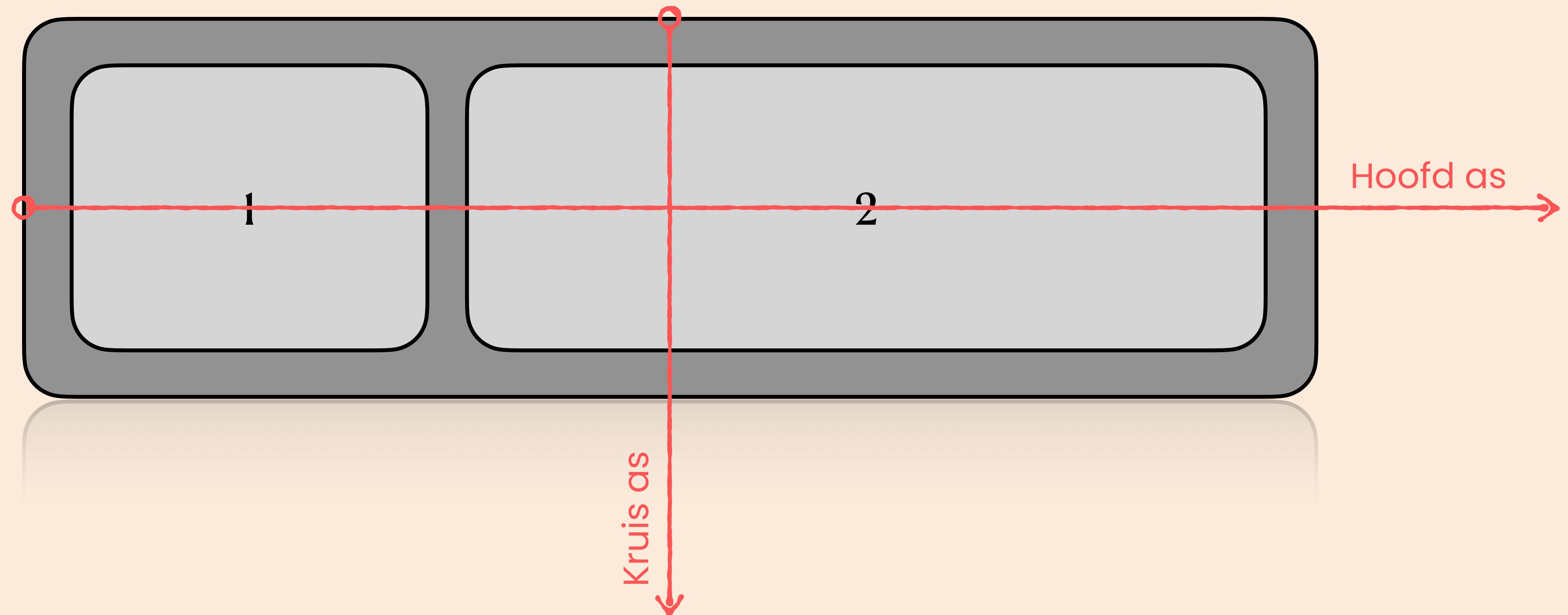
# Flexbox is science

We hebben flexbox properties die alleen op een specifieke as werken



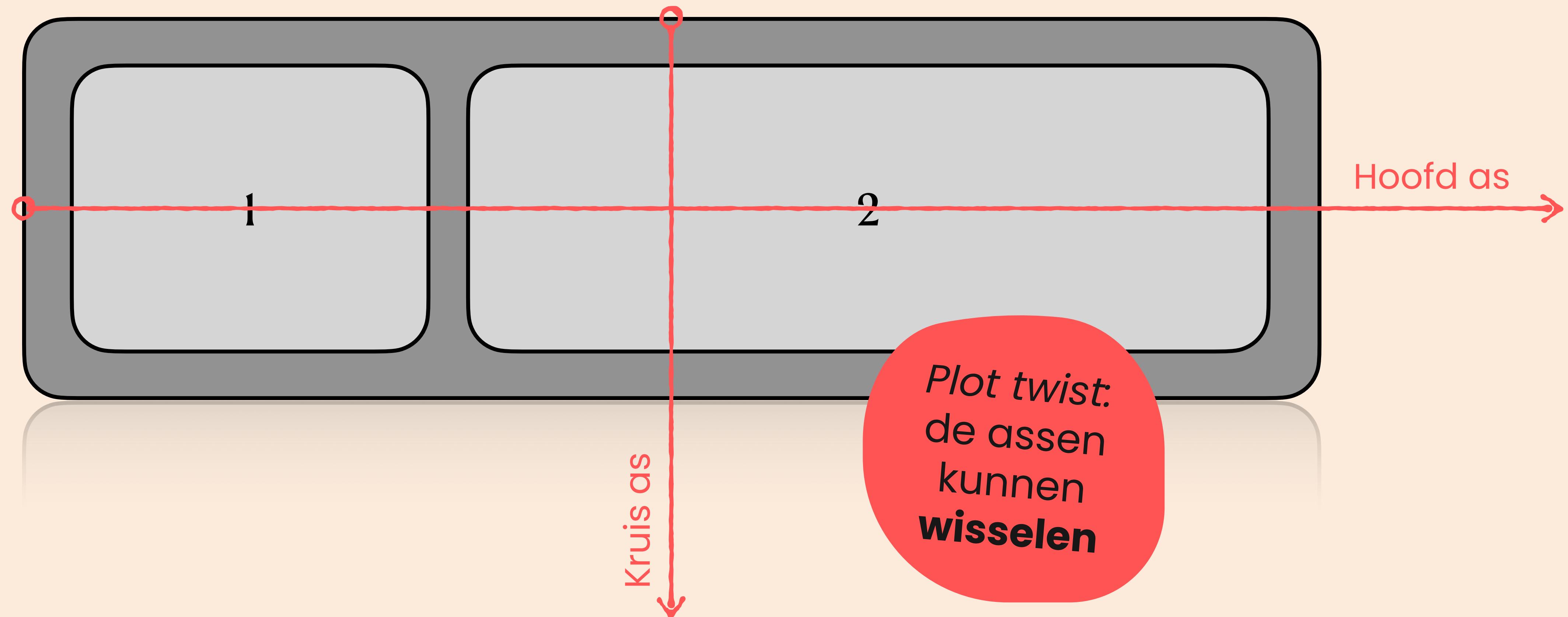
# Flexbox is science

We hebben flexbox properties die alleen op een specifieke as werken

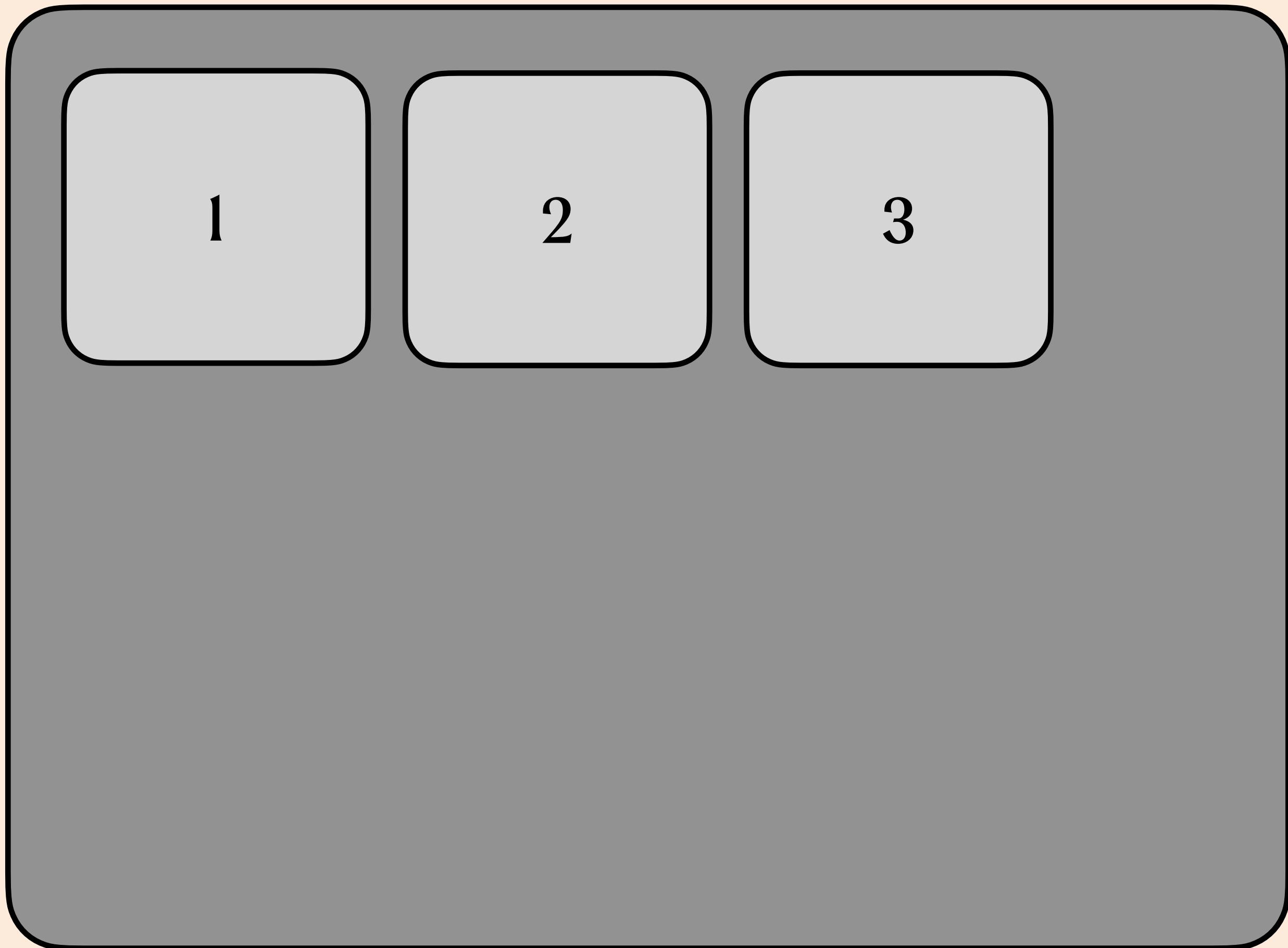


# Flexbox is science

We hebben flexbox properties die alleen op een specifieke as werken

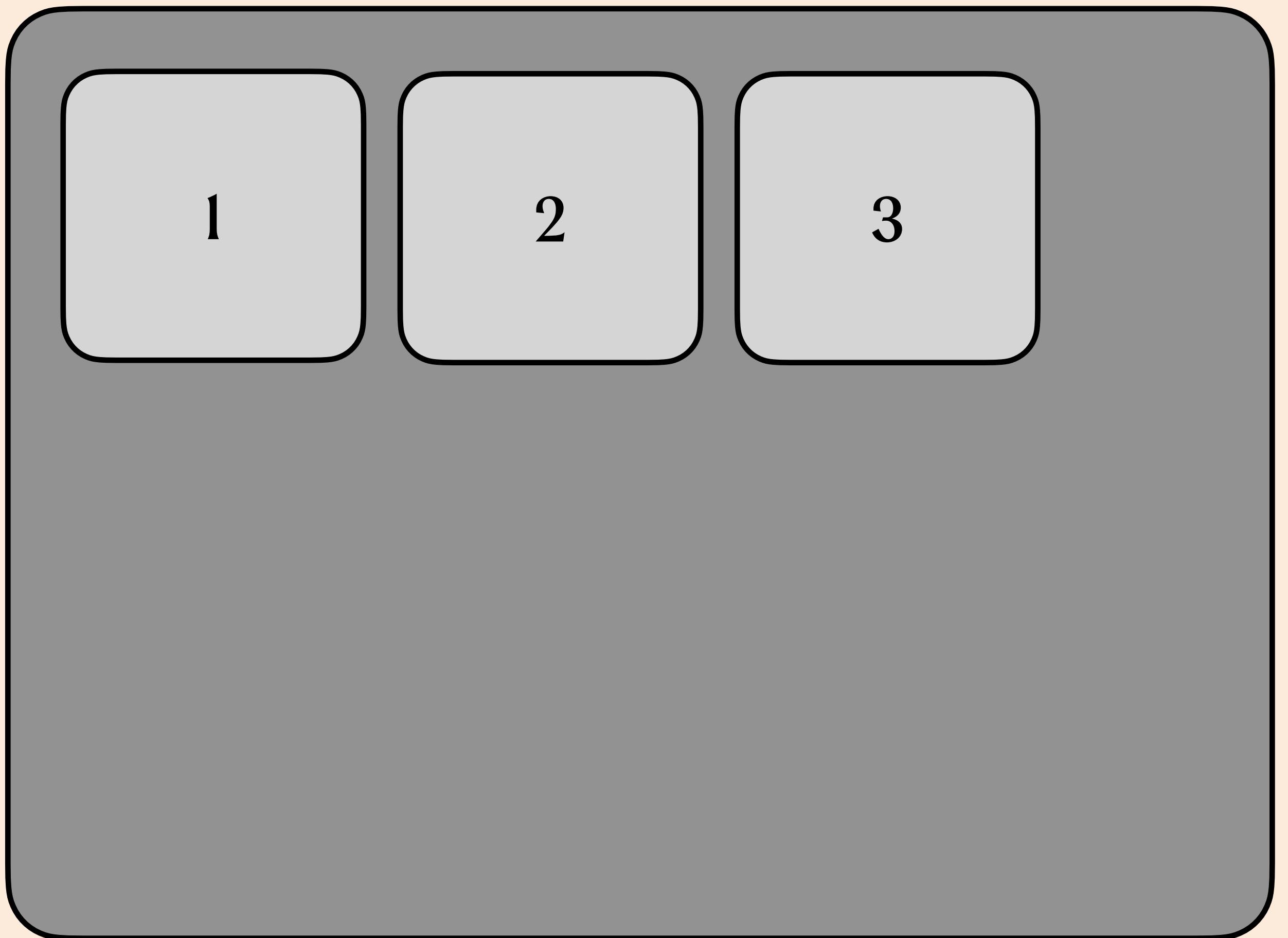


# Flexbox is science



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

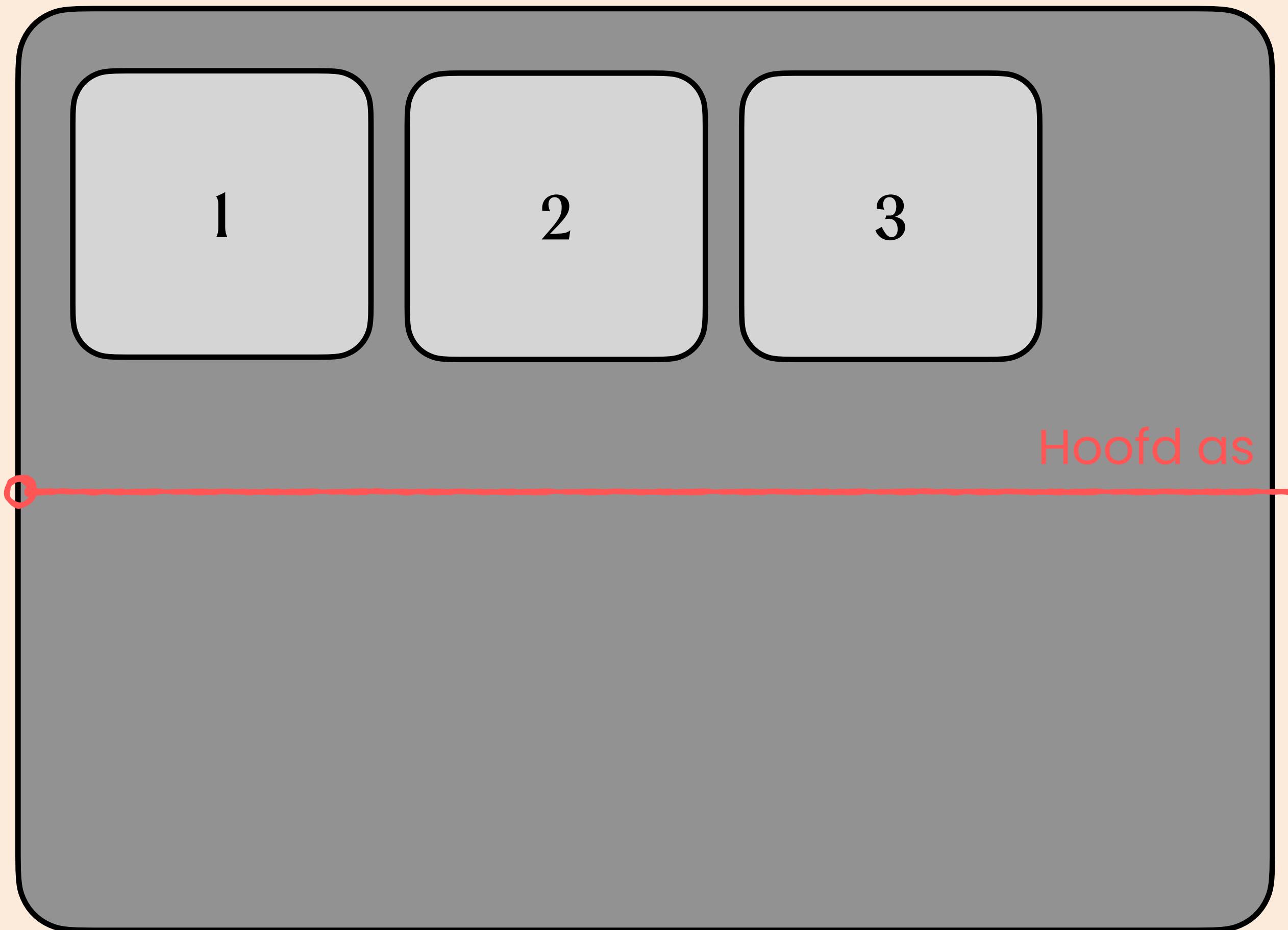
# Flexbox is science



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

De **direction**  
bepaalt wat  
de hoofdas  
is

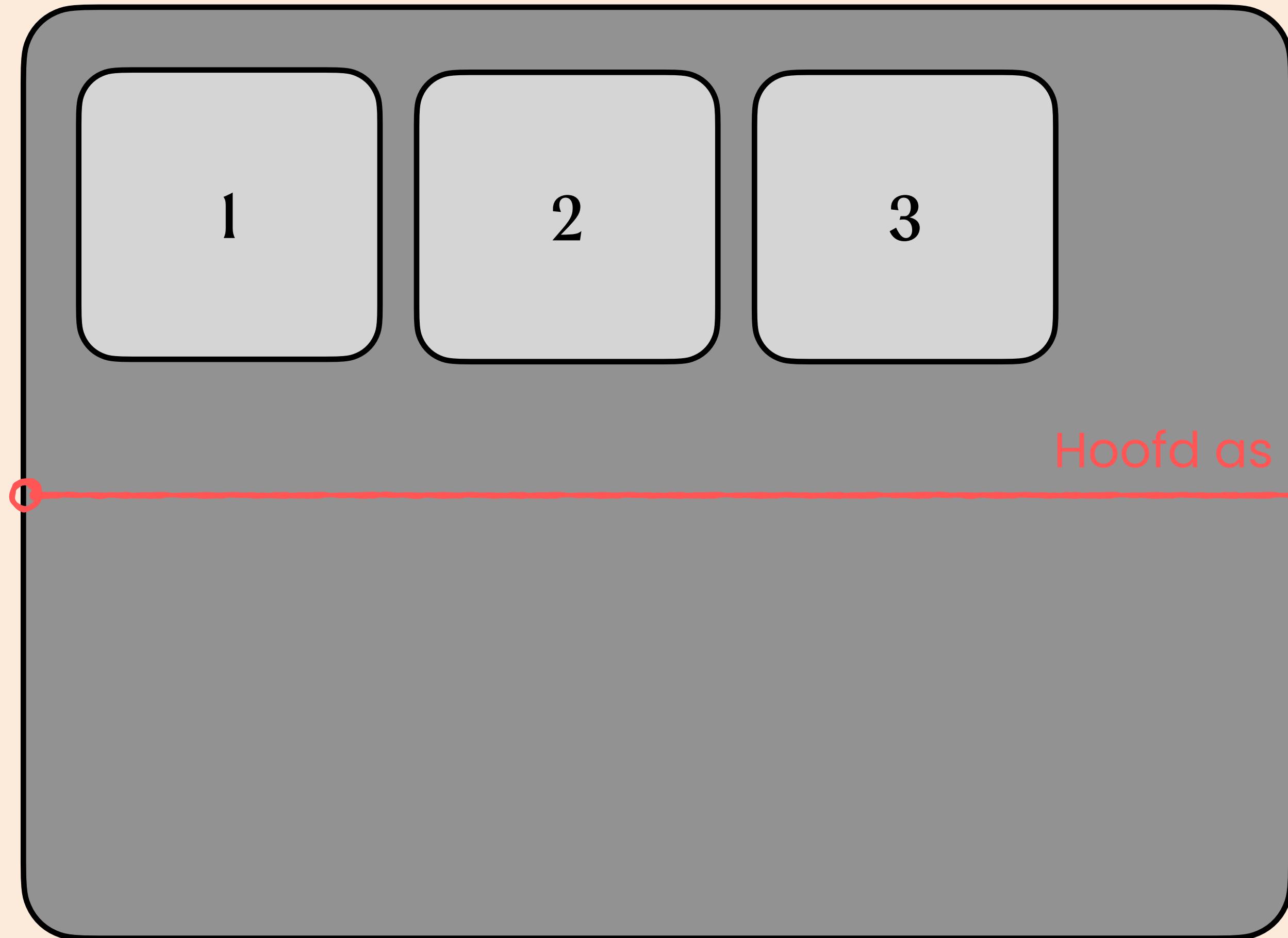
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
}
```

De **direction**  
bepaalt wat  
de hoofdas  
is

# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
}
```

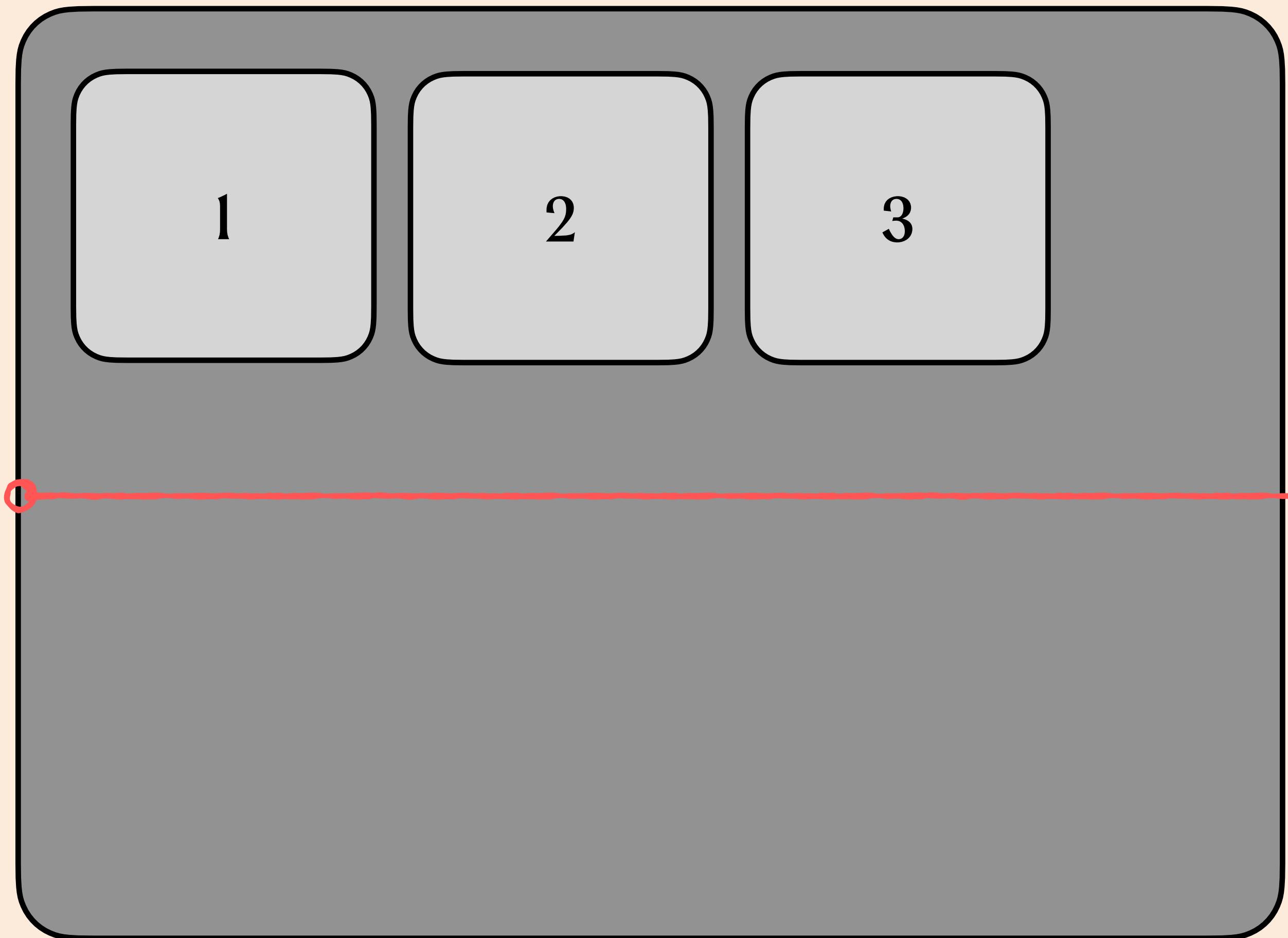
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
}
```

**justify-content**  
is een hoofd-as  
property

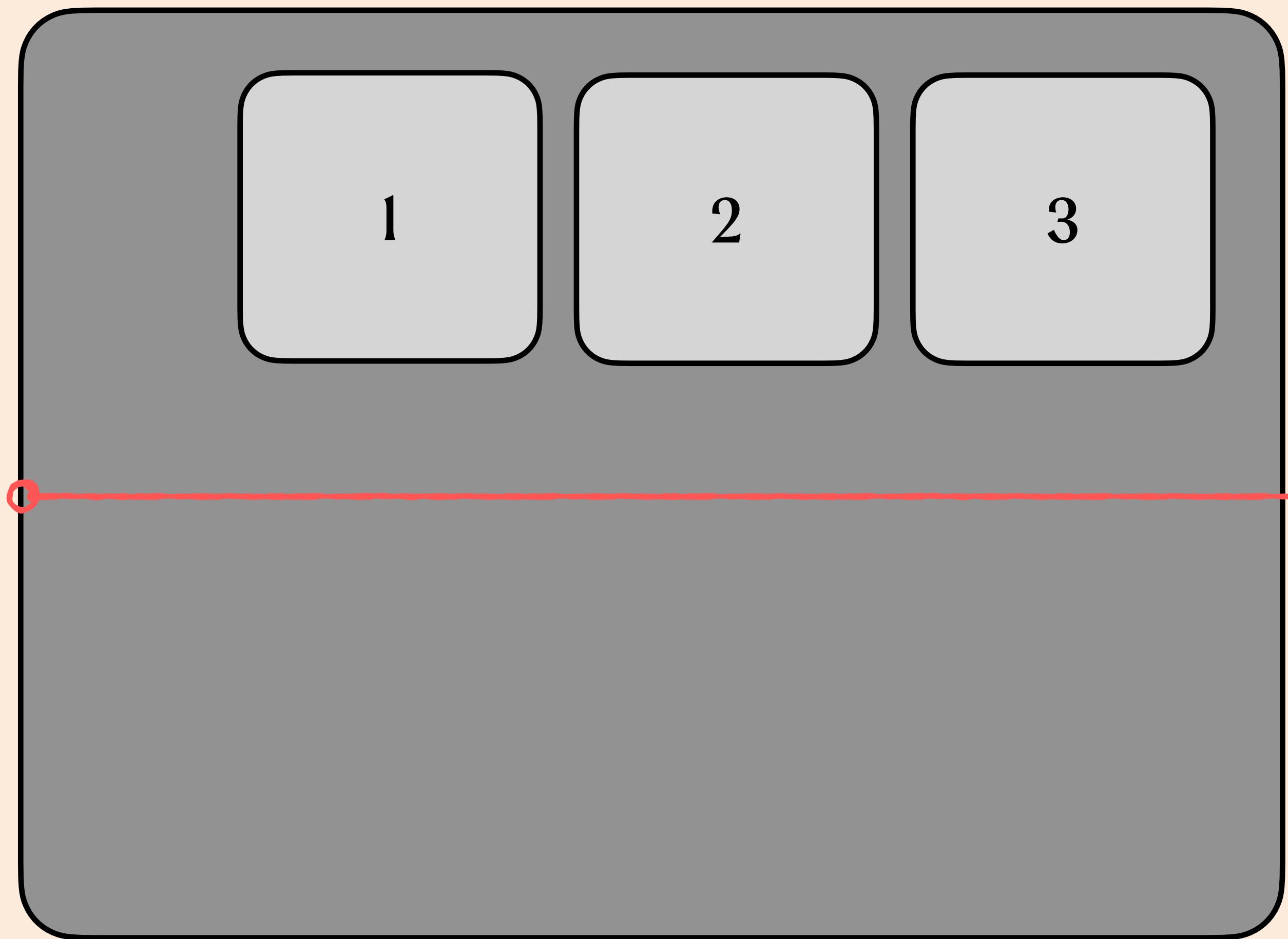
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
}
```

**justify-content**  
is een hoofd-as  
property

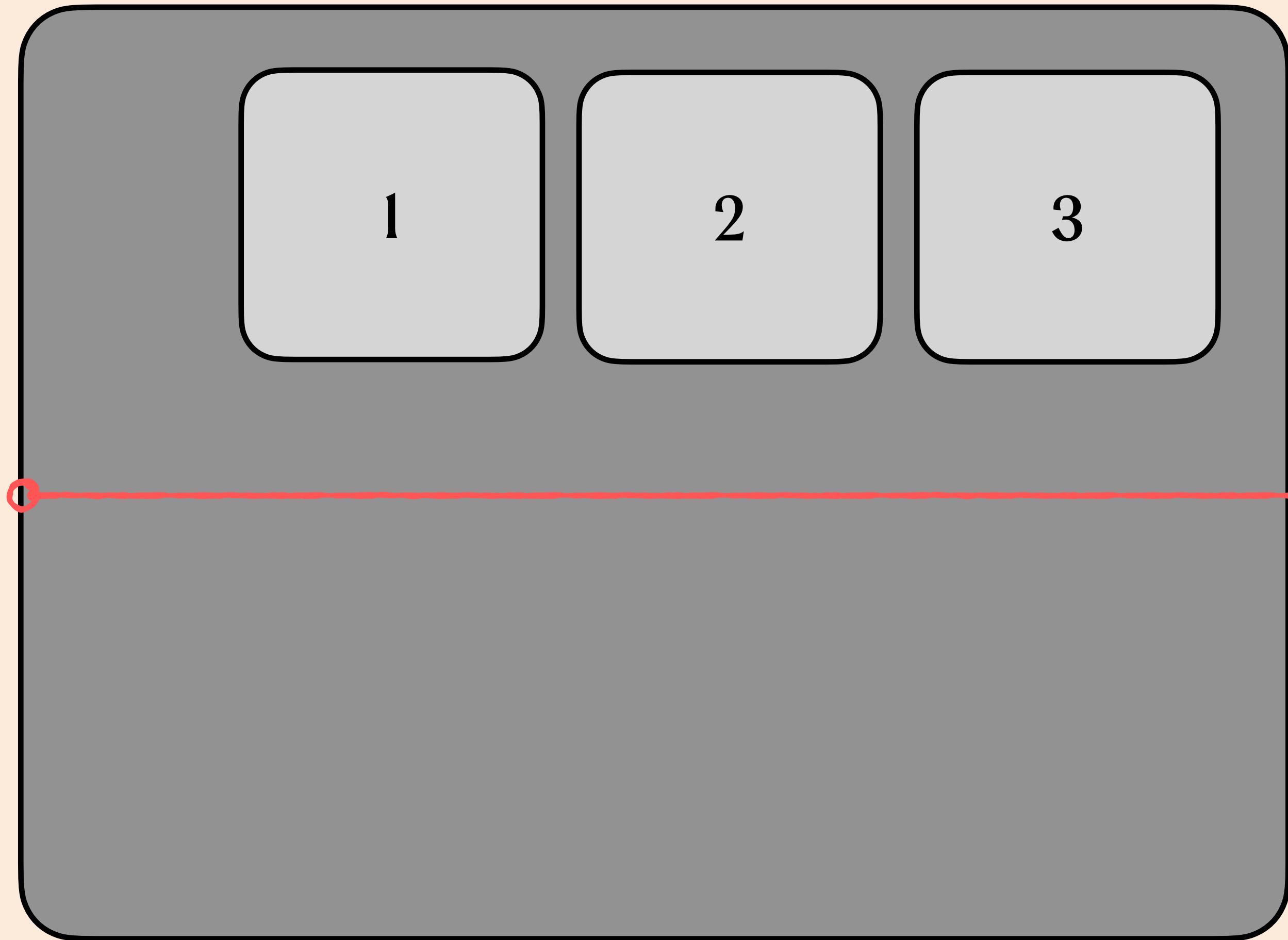
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
}
```

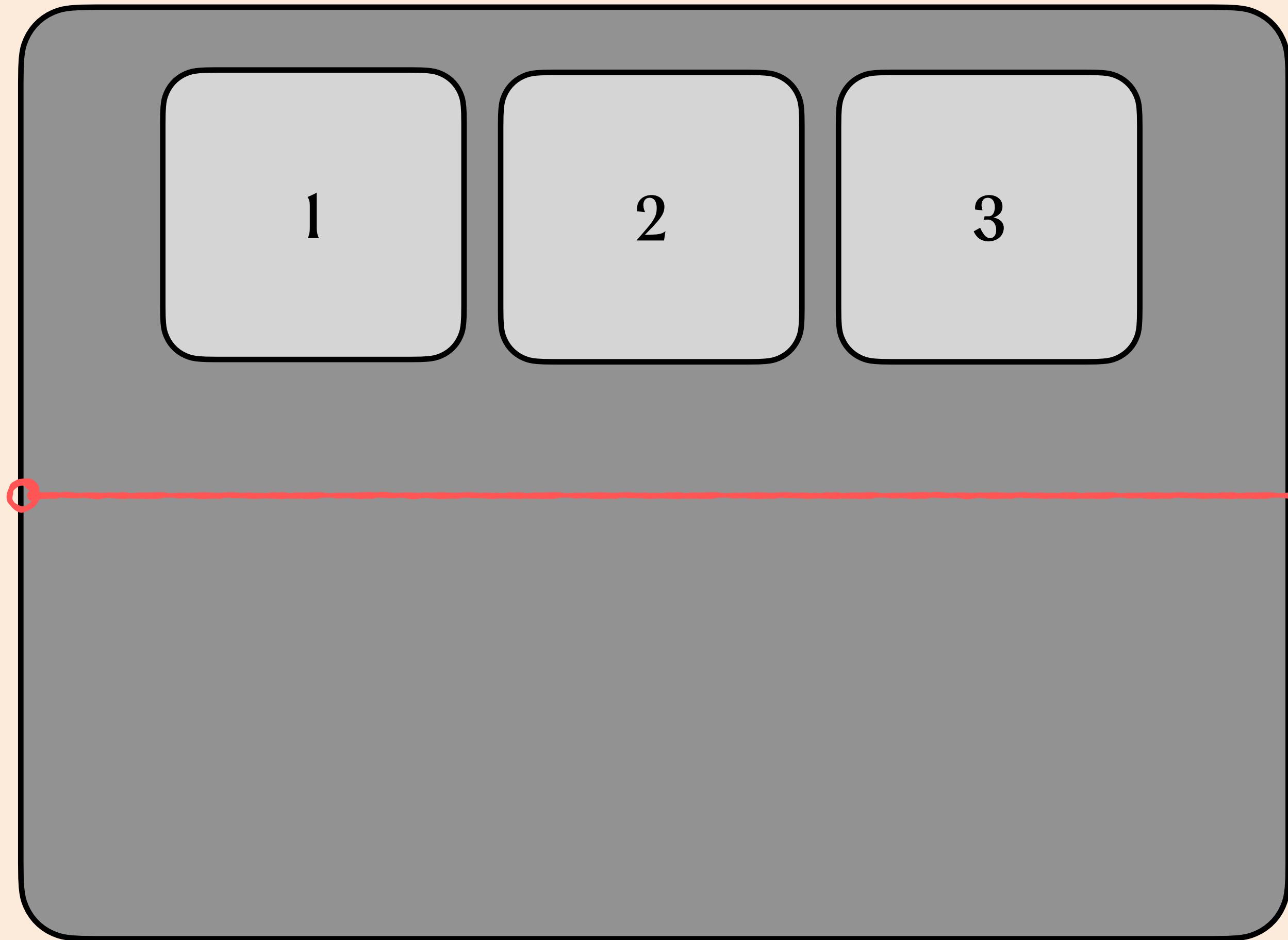
**justify-content**  
is een hoofd-as  
property

# Flexbox is science



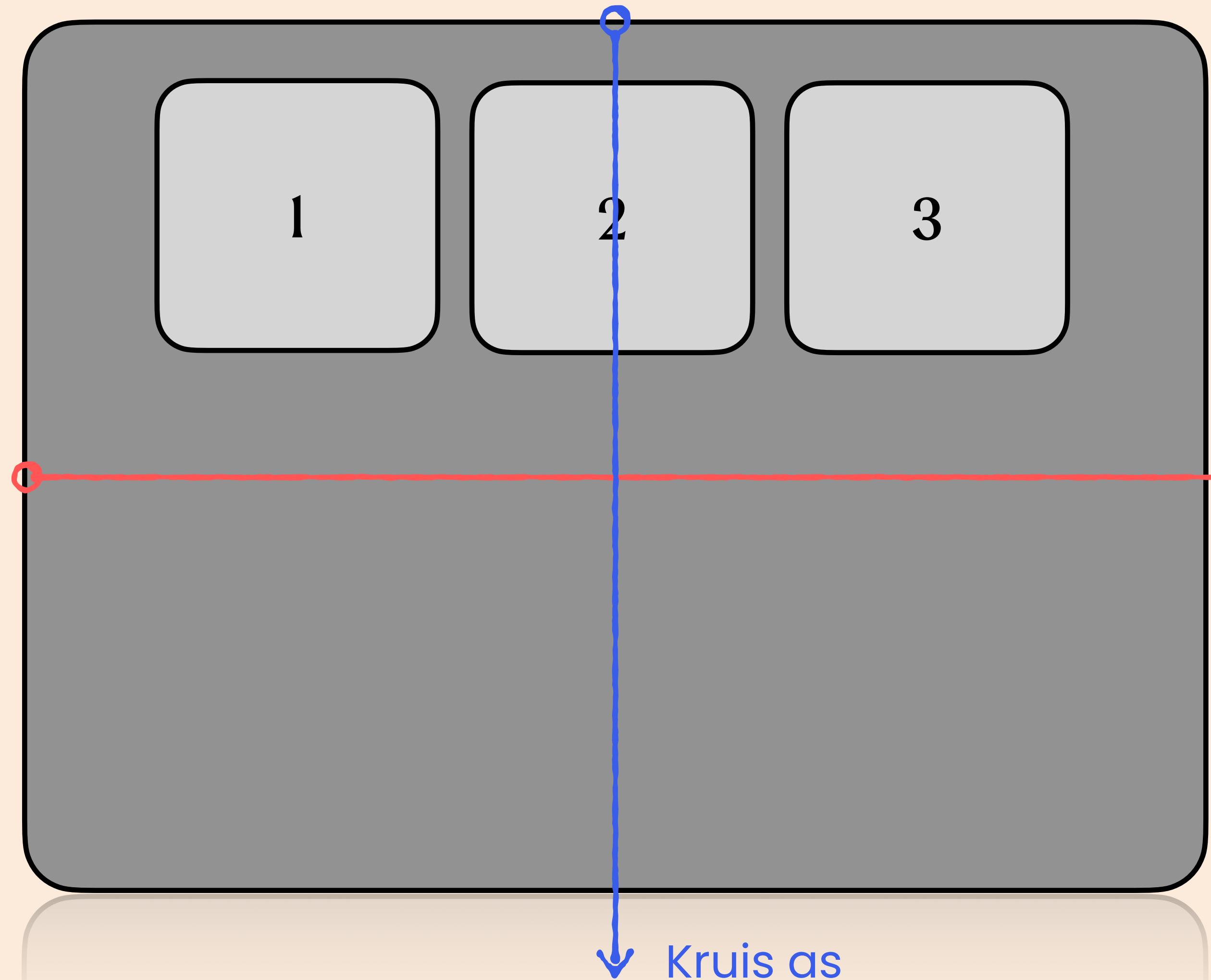
```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

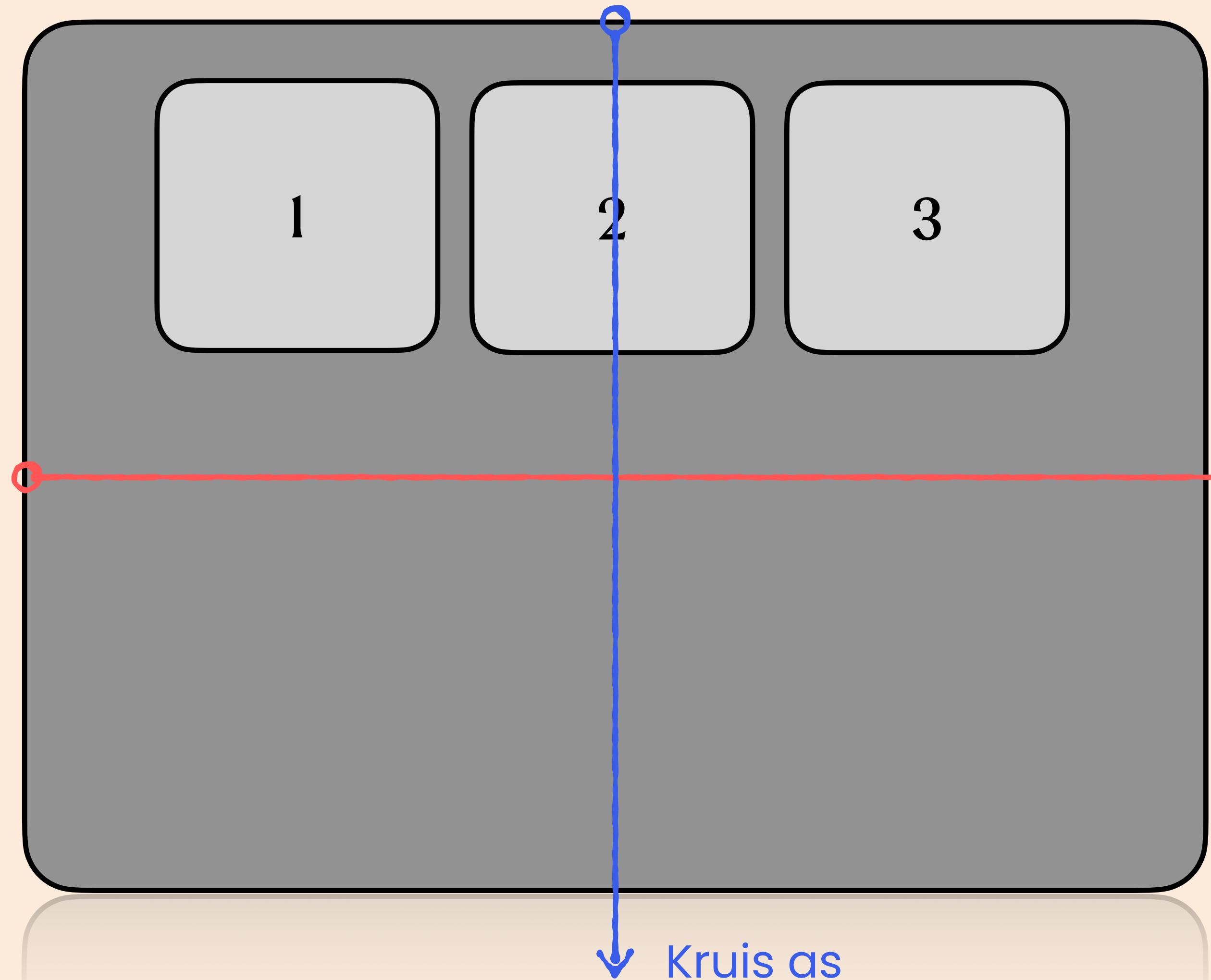
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

**align-items**  
is een kruis-as  
property

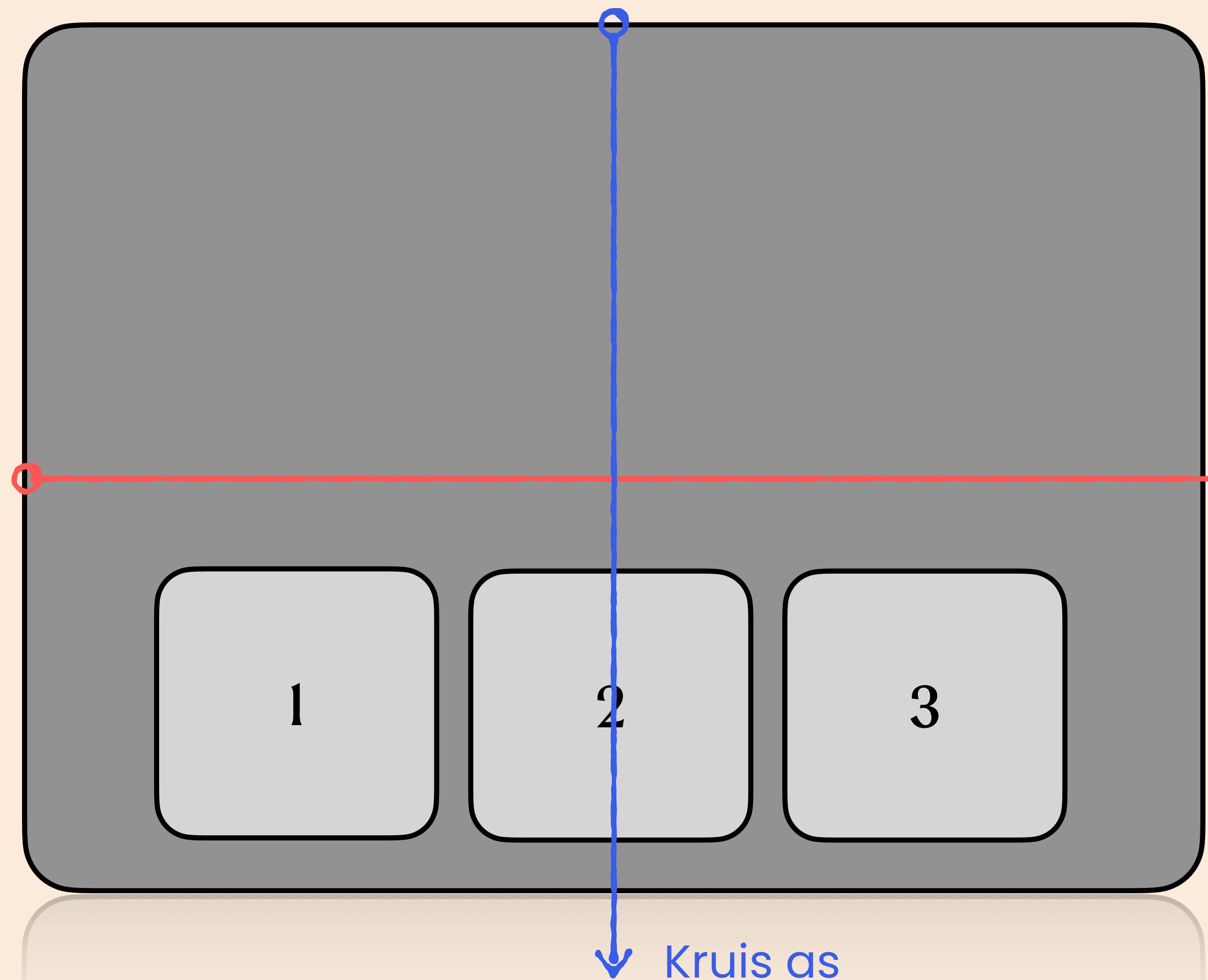
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: flex-end;  
}
```

**align-items**  
is een kruis-as  
property

# Flexbox is science



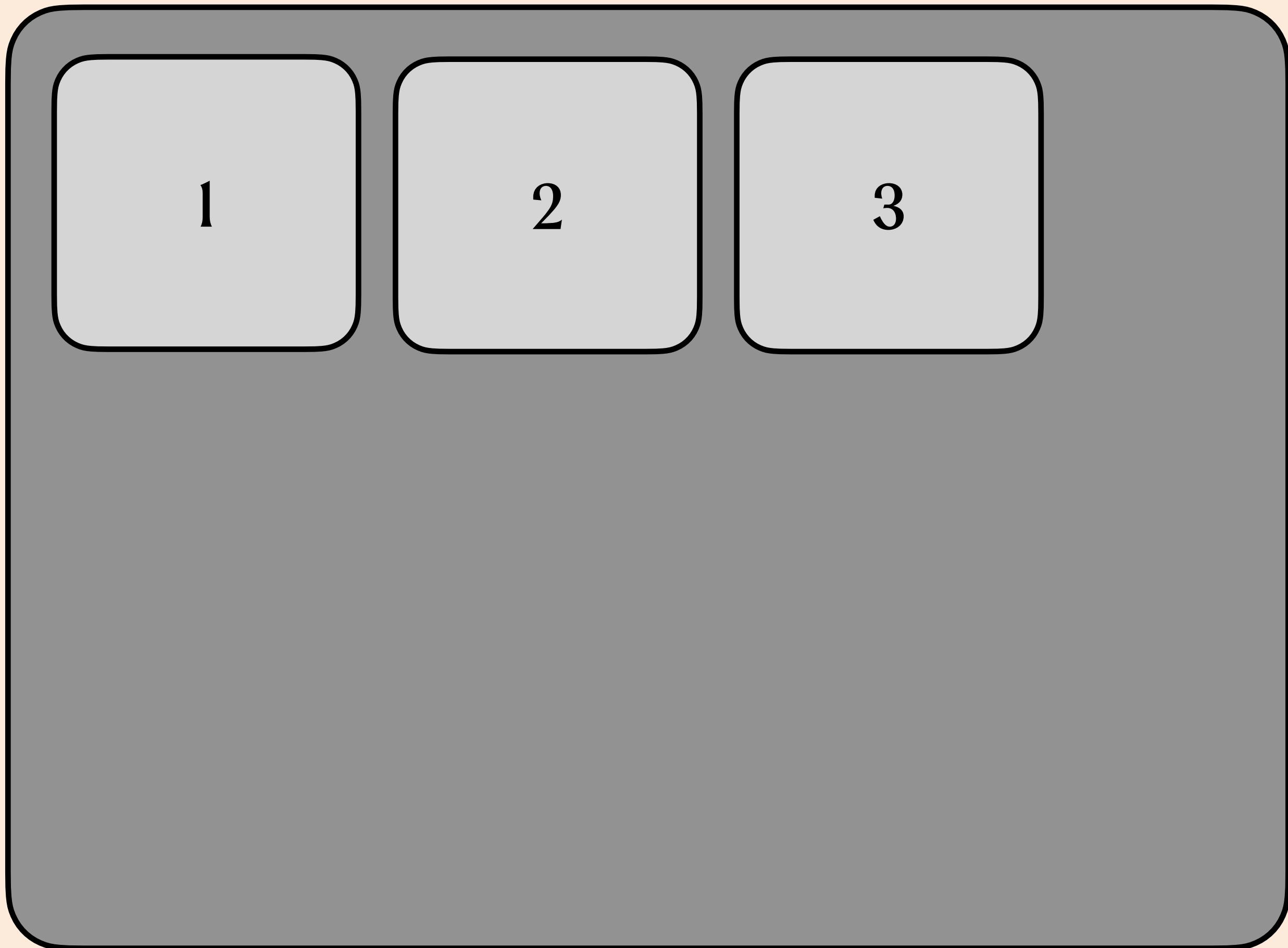
```
.container {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: flex-end;  
}
```

**align-items**  
is een kruis-as  
property

# Flexbox is science

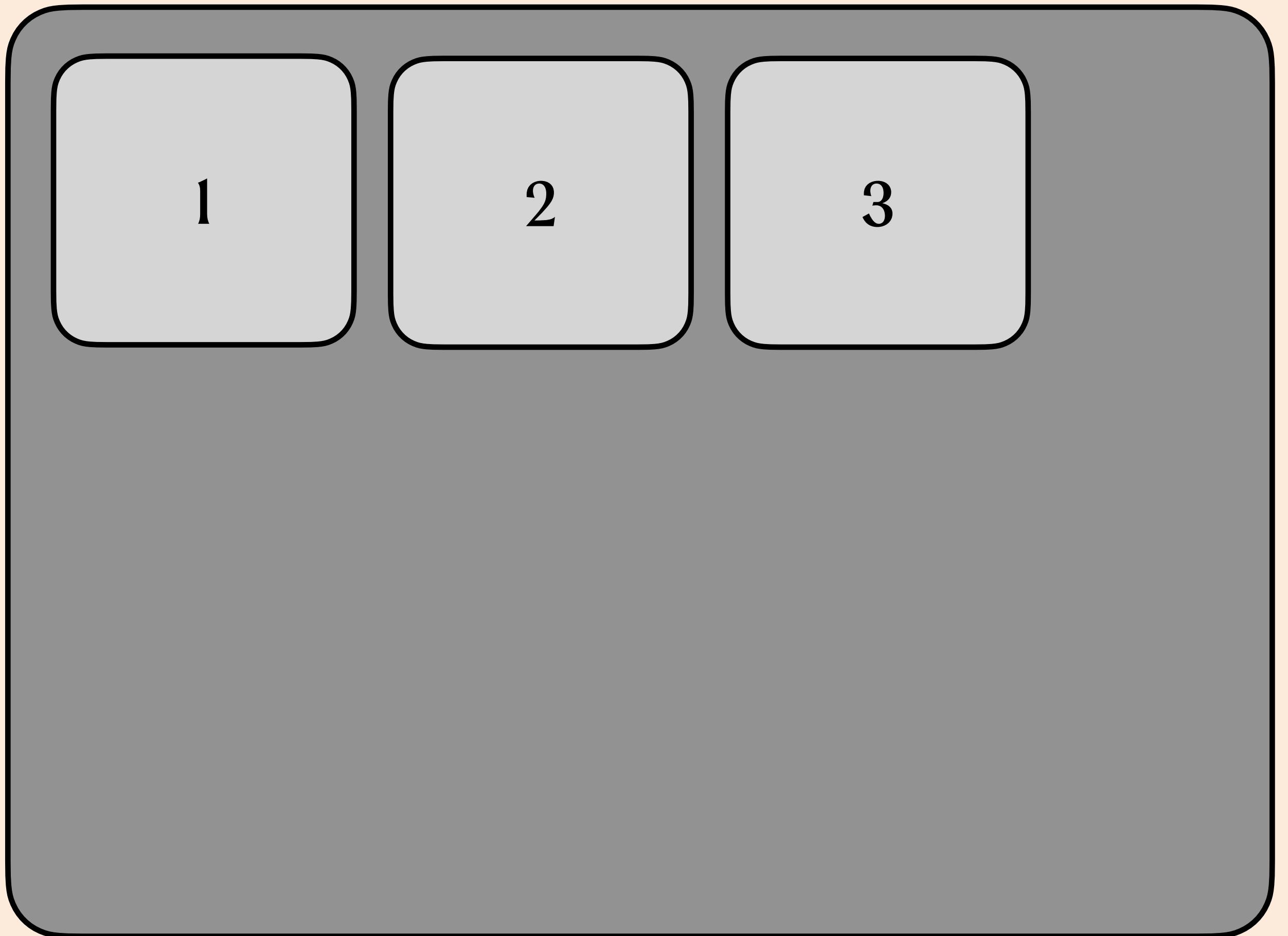
Nu gaan we **dezelfde properties** toepassen,  
maar veranderen we de hoofd-as

# Flexbox is science



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

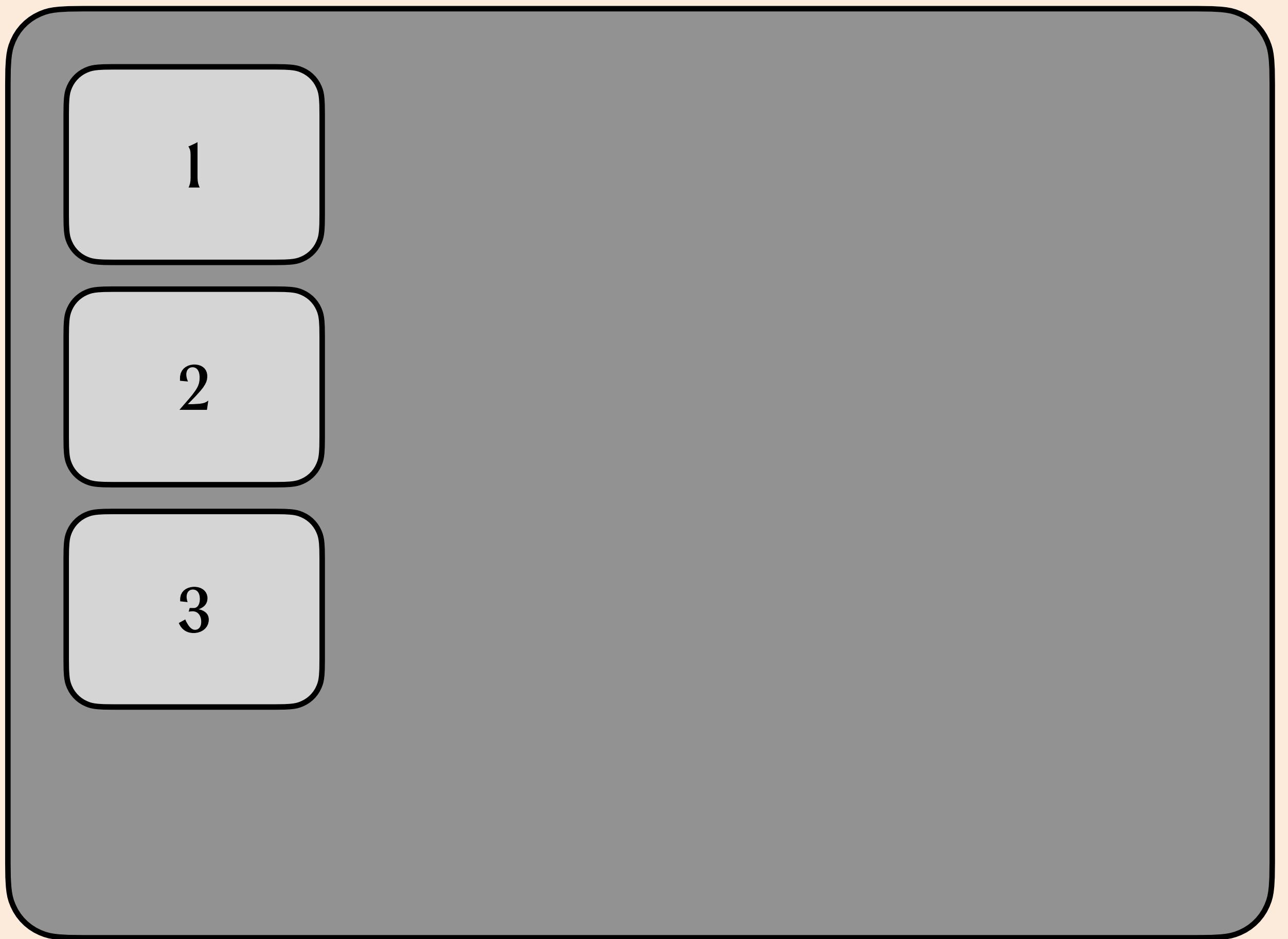
# Flexbox is science



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

De **direction**  
bepaalt wat  
de hoofdas  
is

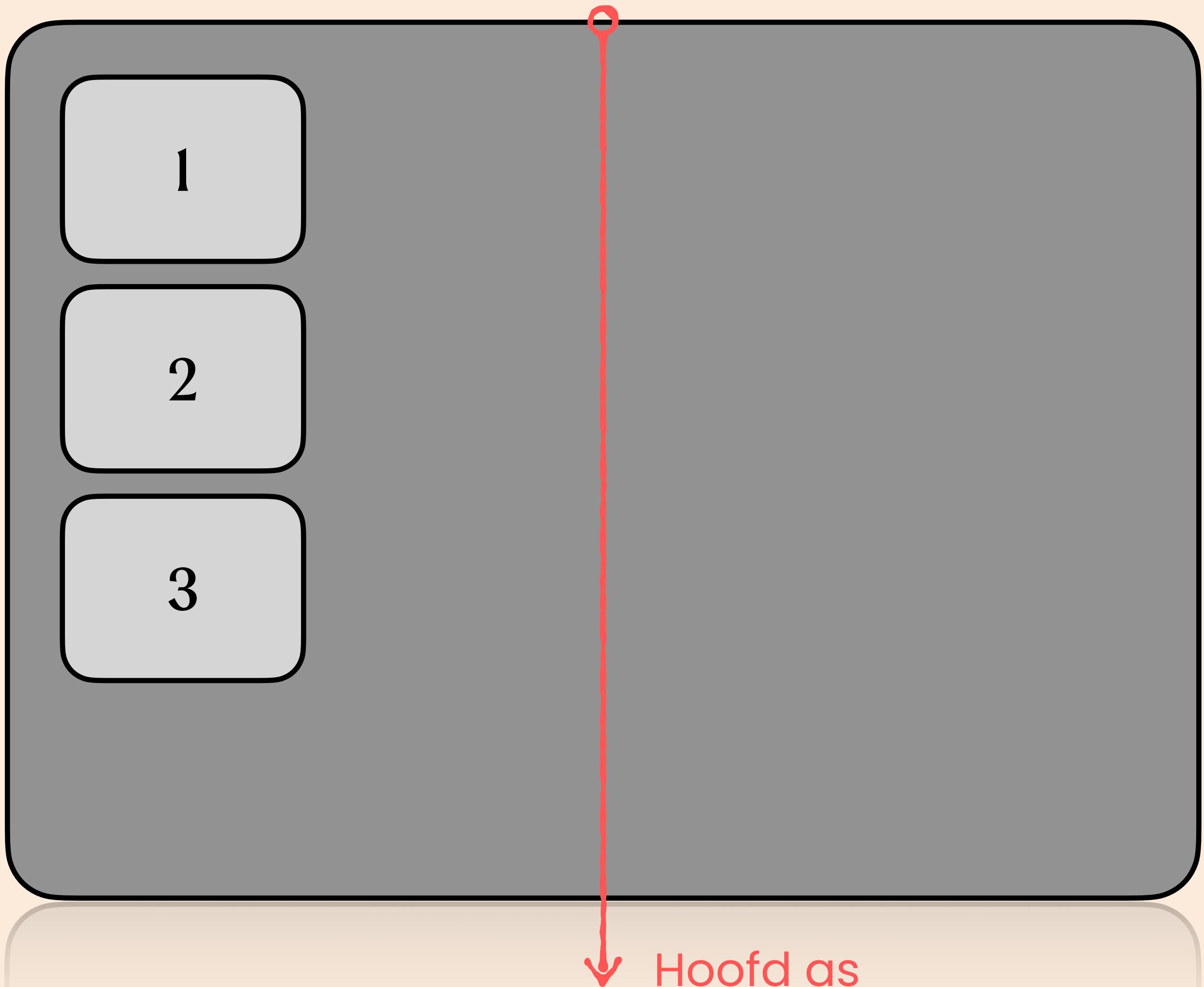
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

De **direction**  
bepaalt wat  
de hoofdas  
is

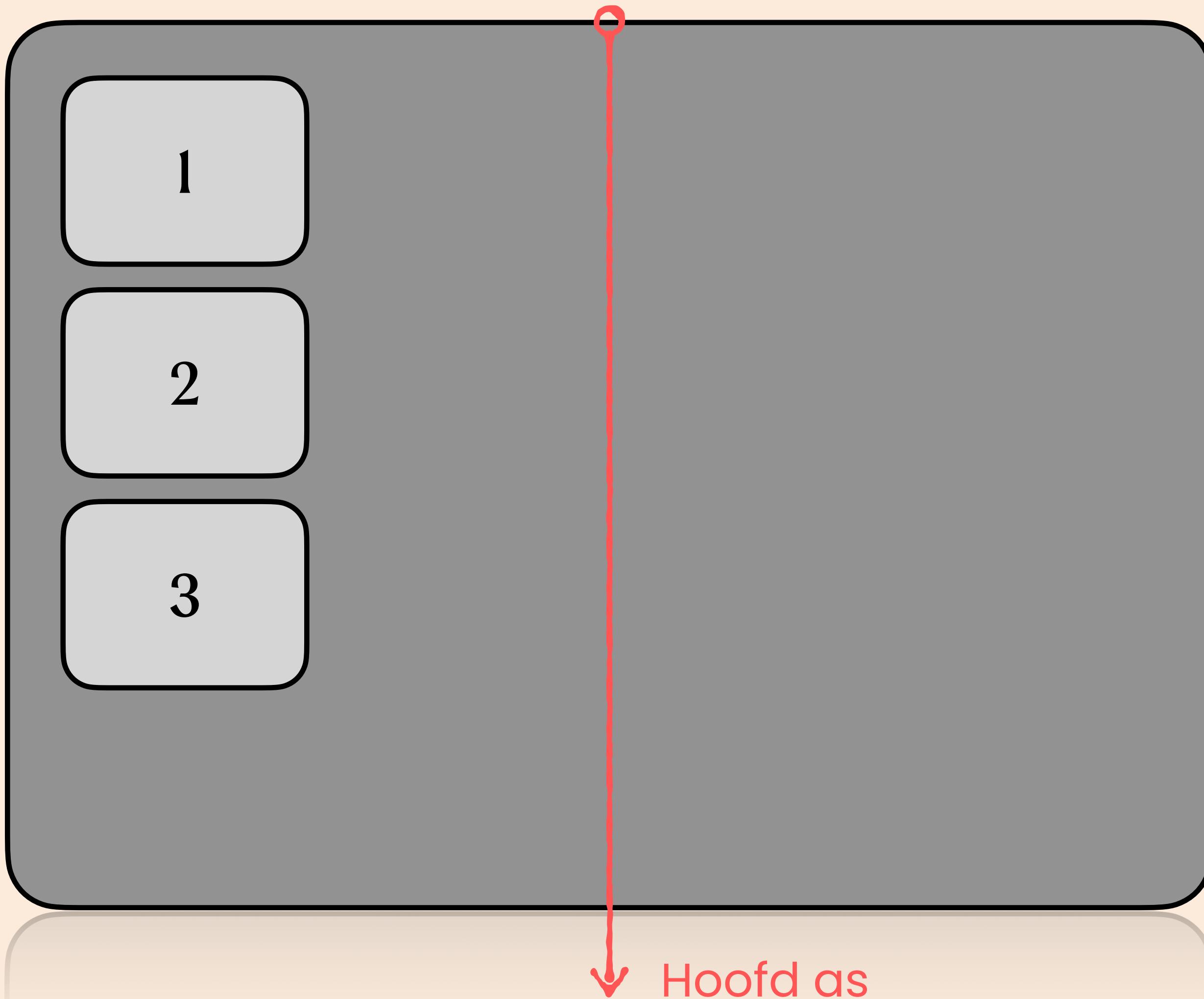
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

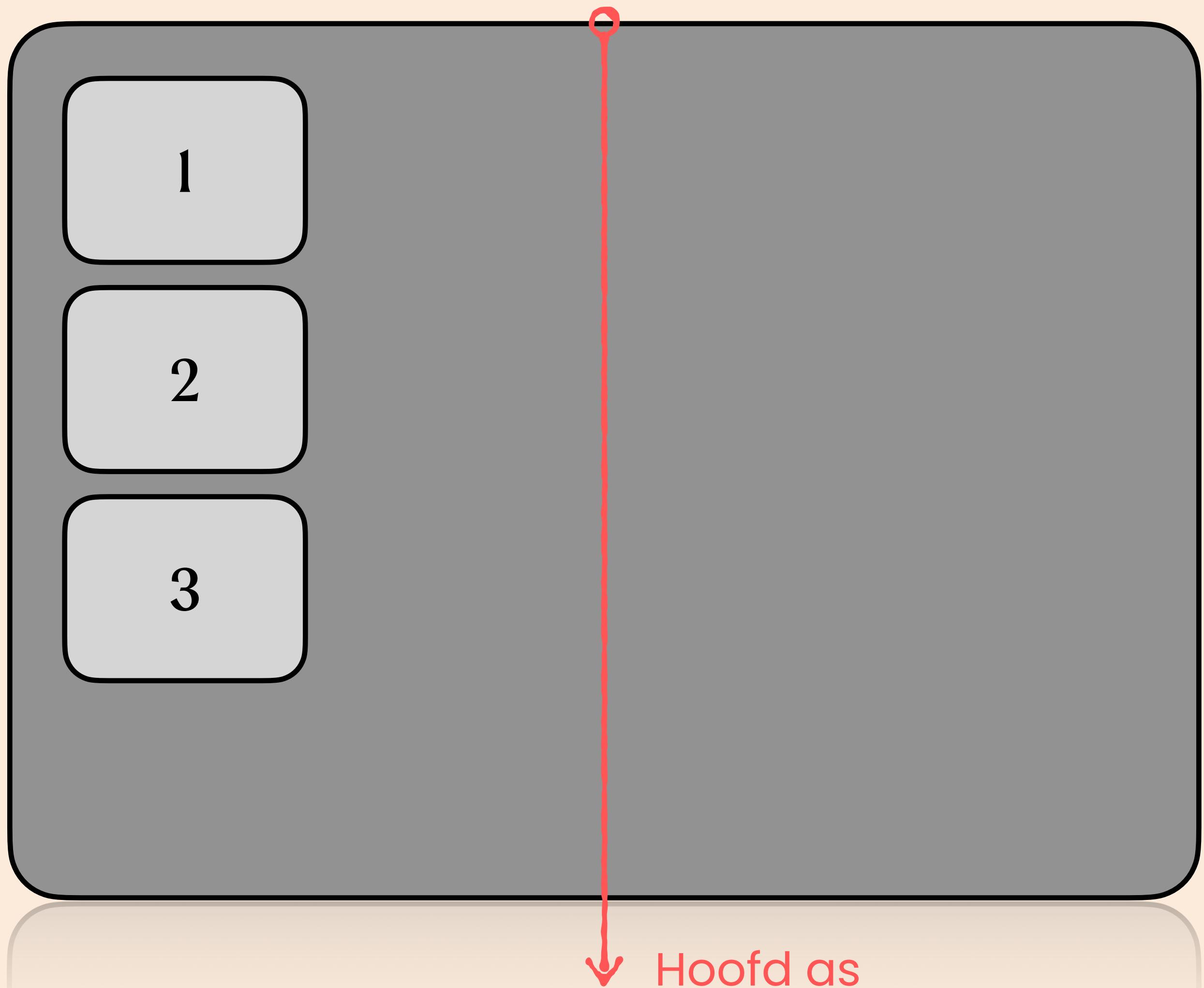
De **direction**  
bepaalt wat  
de hoofdas  
is

# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

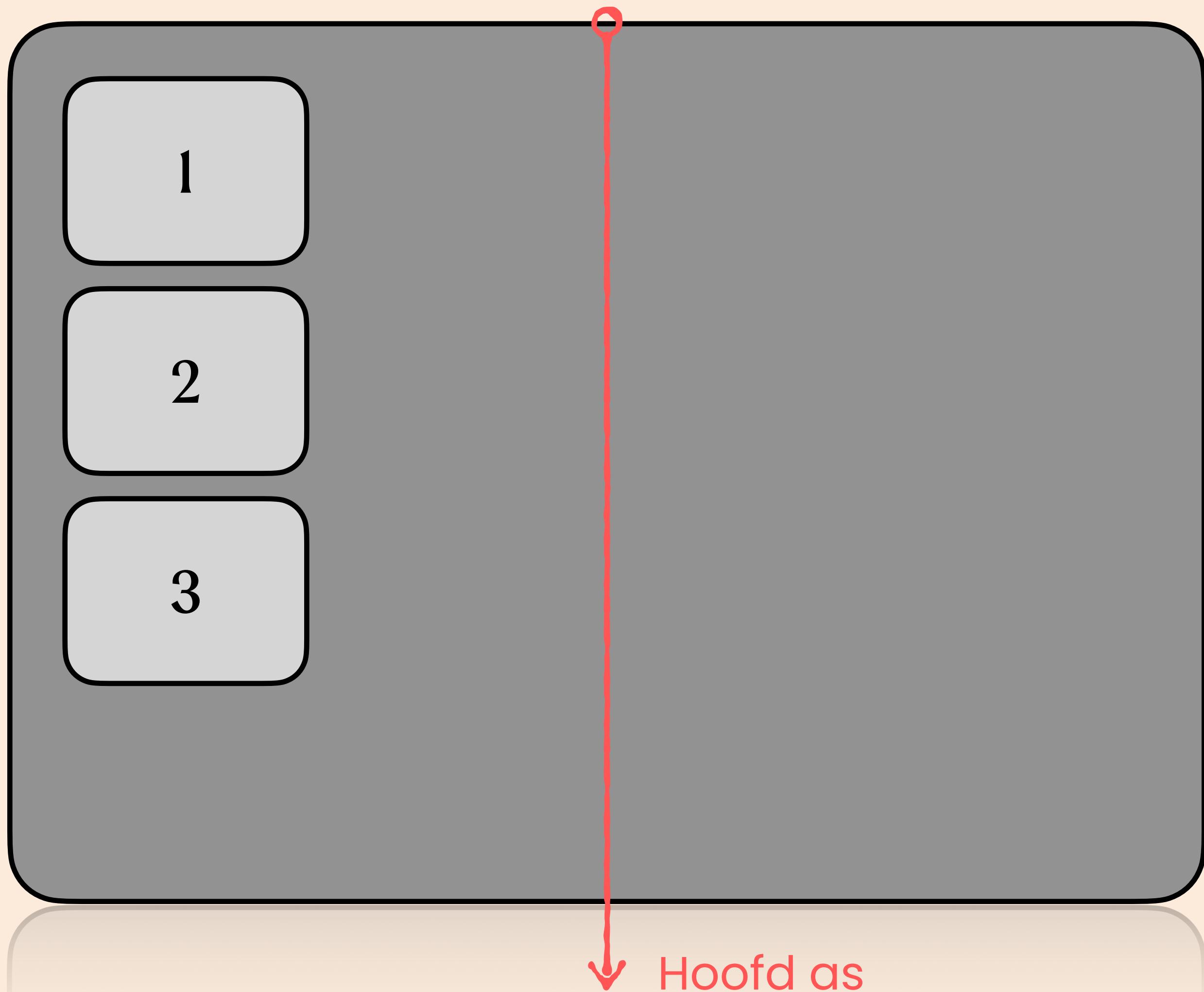
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

**justify-content**  
is een hoofd-as  
property

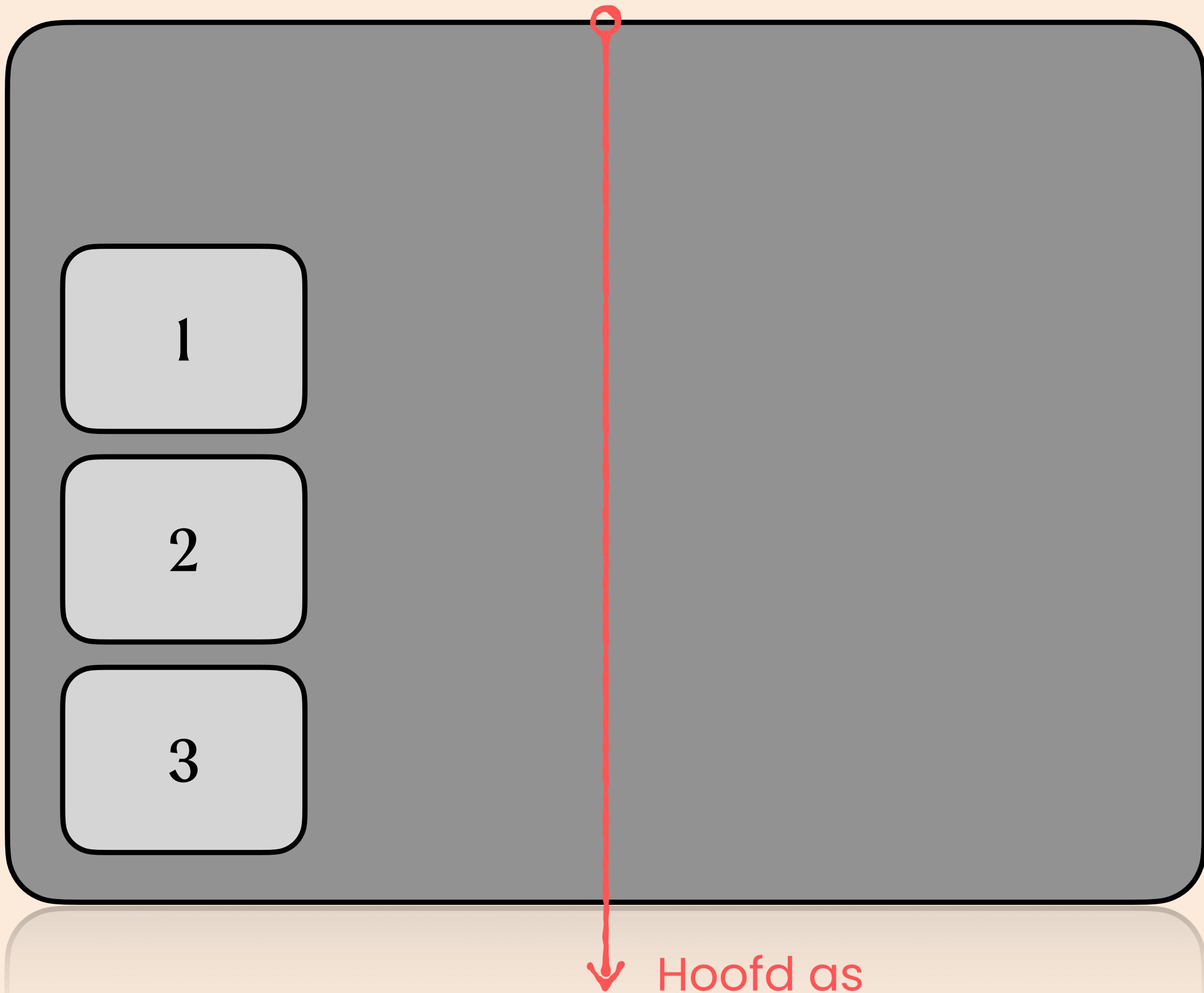
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-end;  
}
```

**justify-content**  
is een hoofd-as  
property

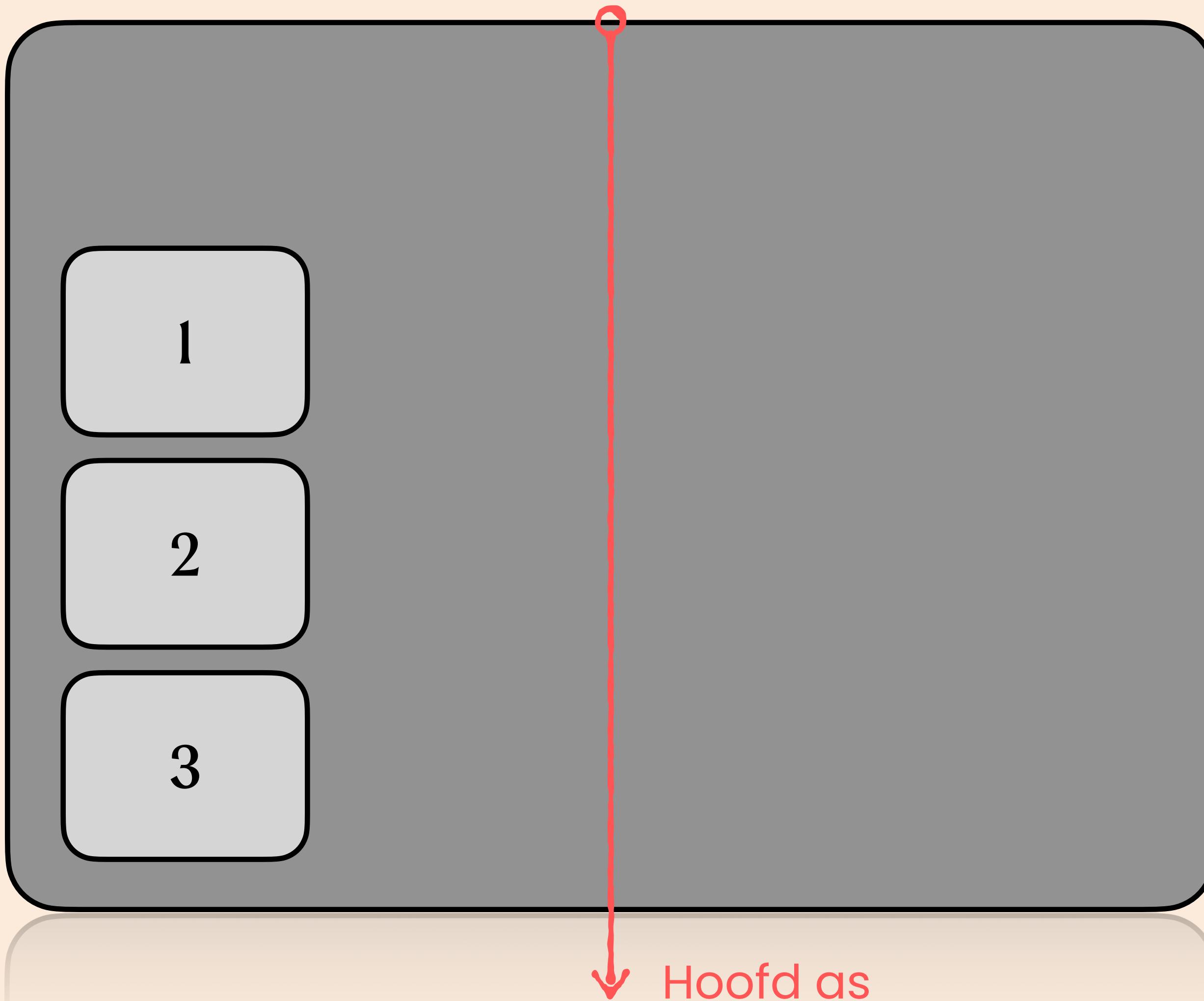
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-end;  
}
```

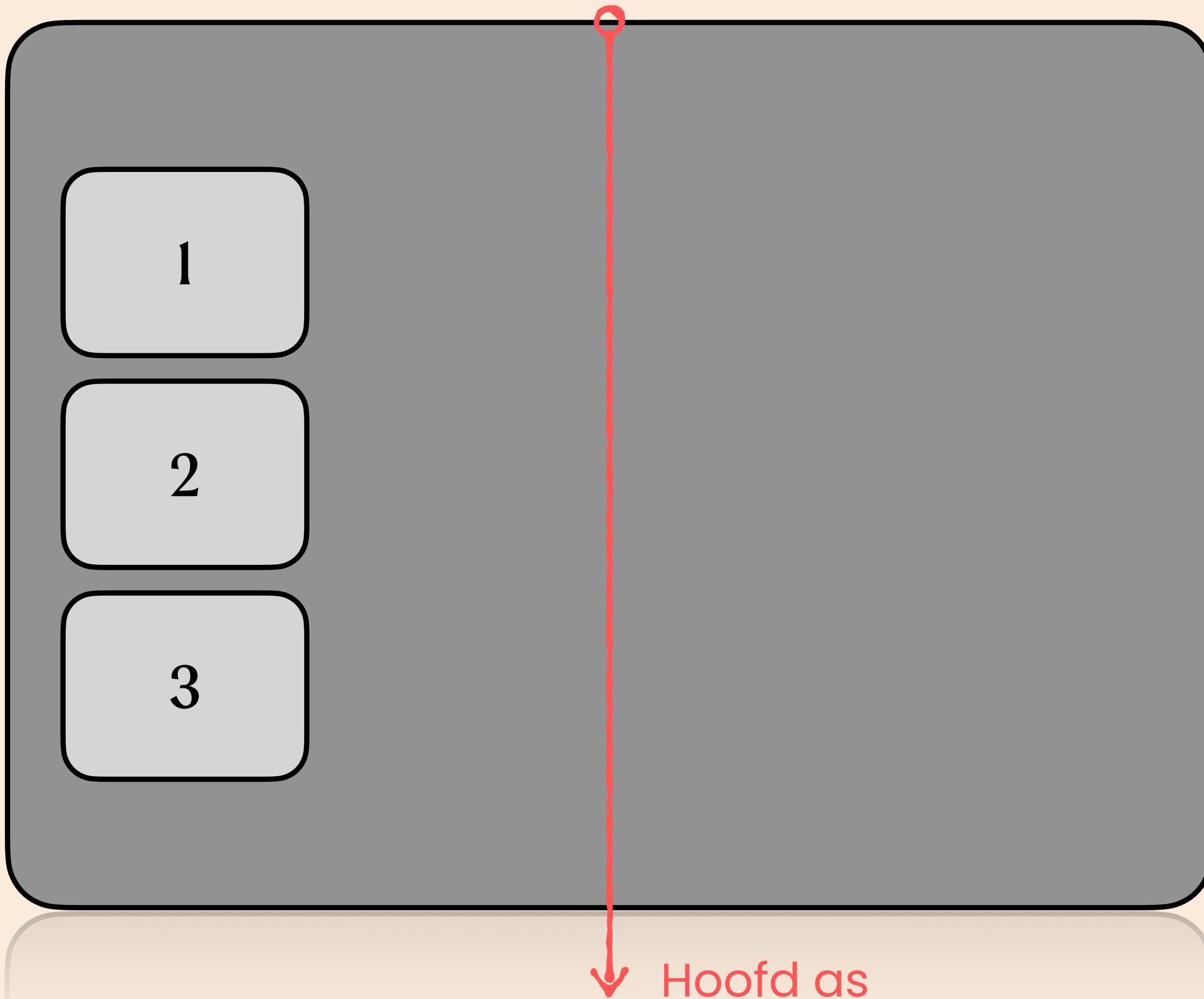
**justify-content**  
is een hoofd-as  
property

# Flexbox is science



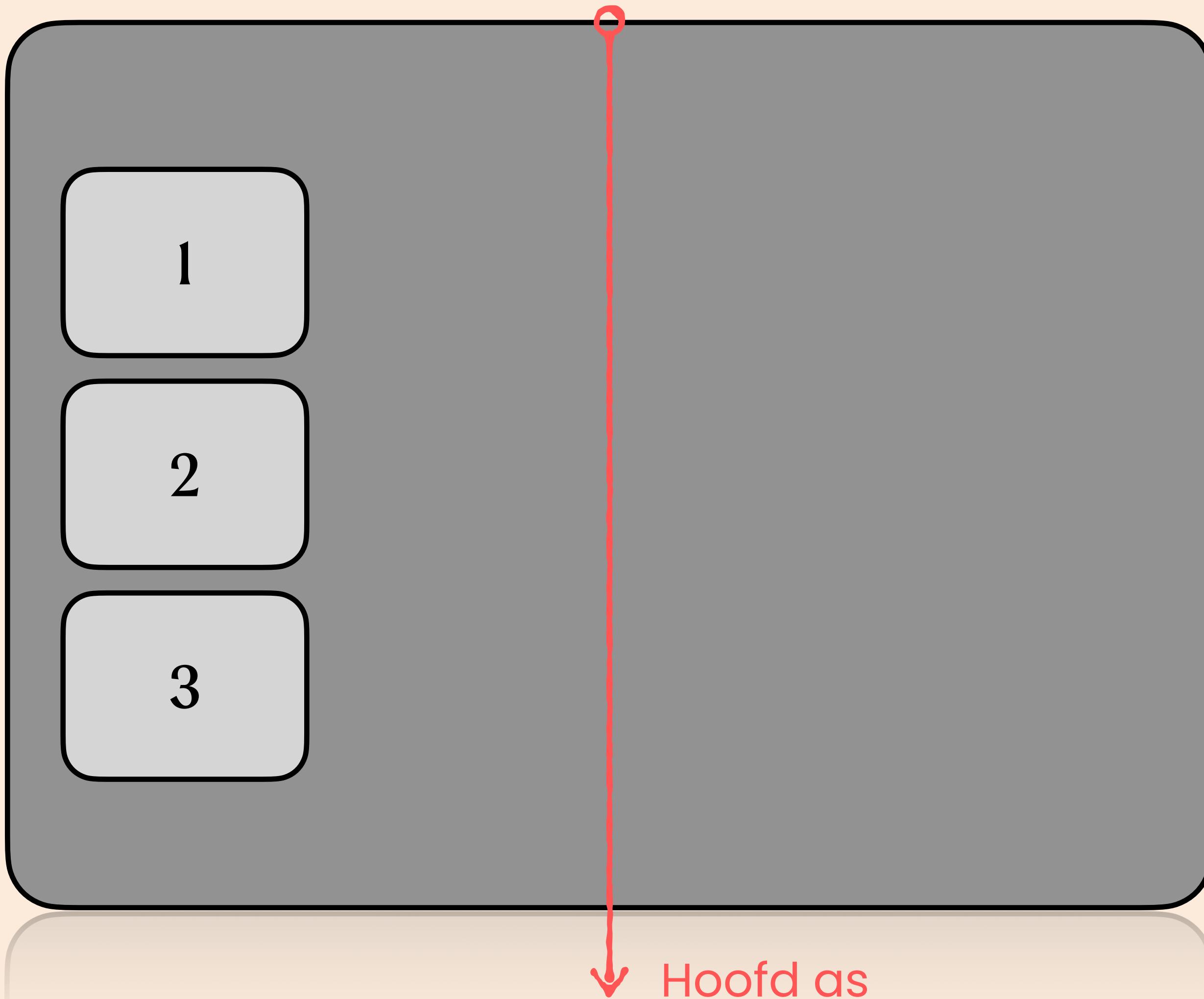
```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

# Flexbox is science



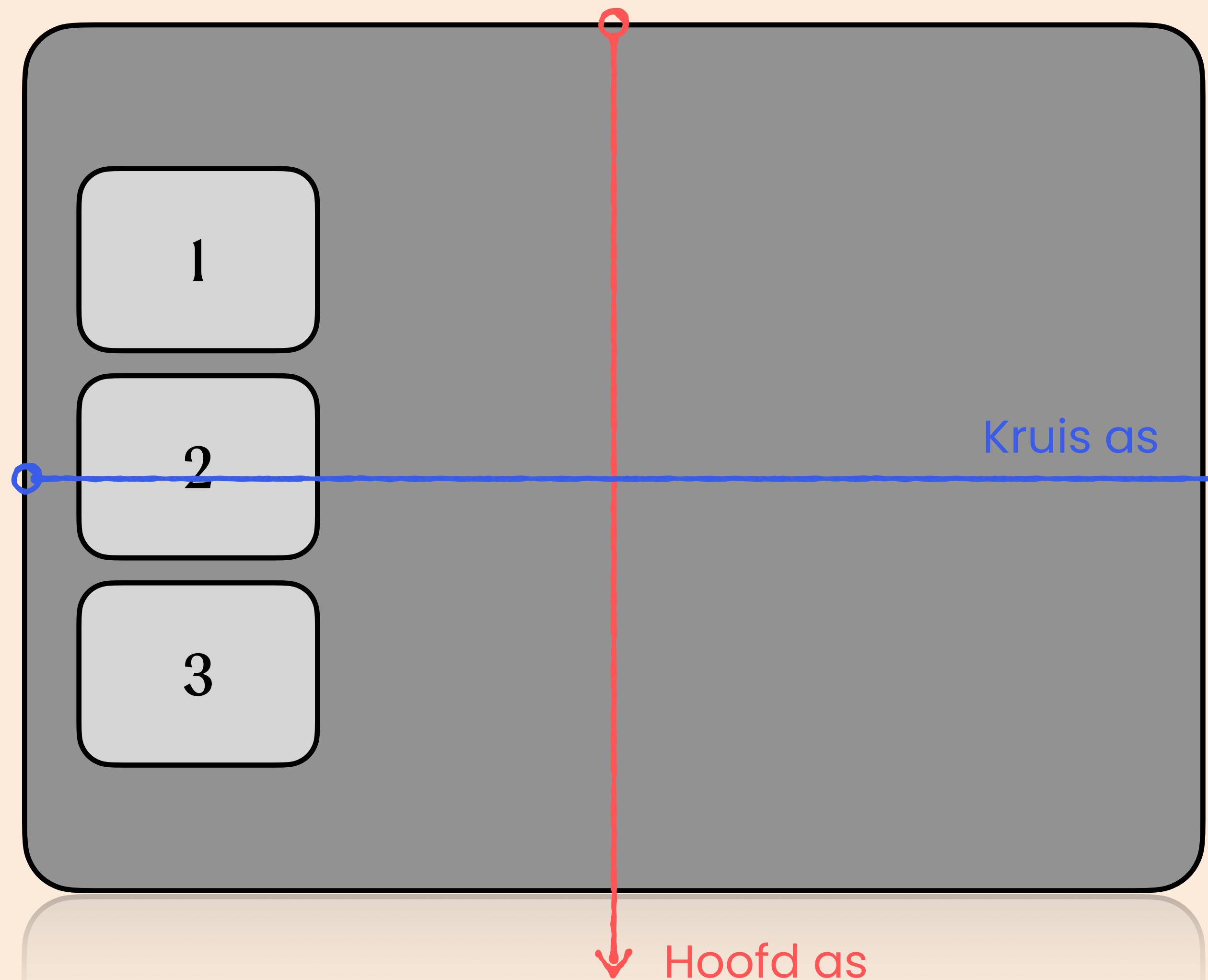
```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

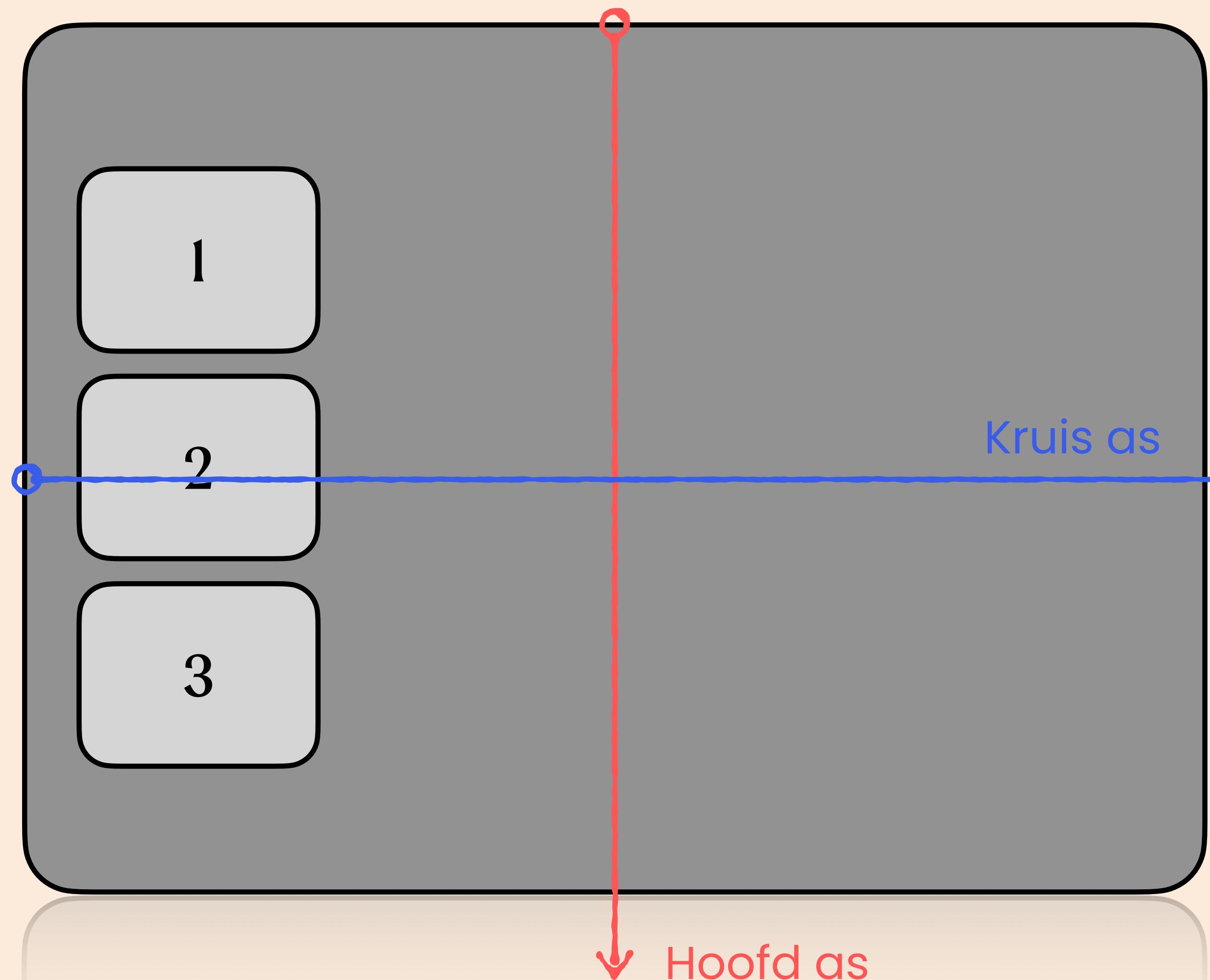
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

**align-items**  
is een kruis-as  
property

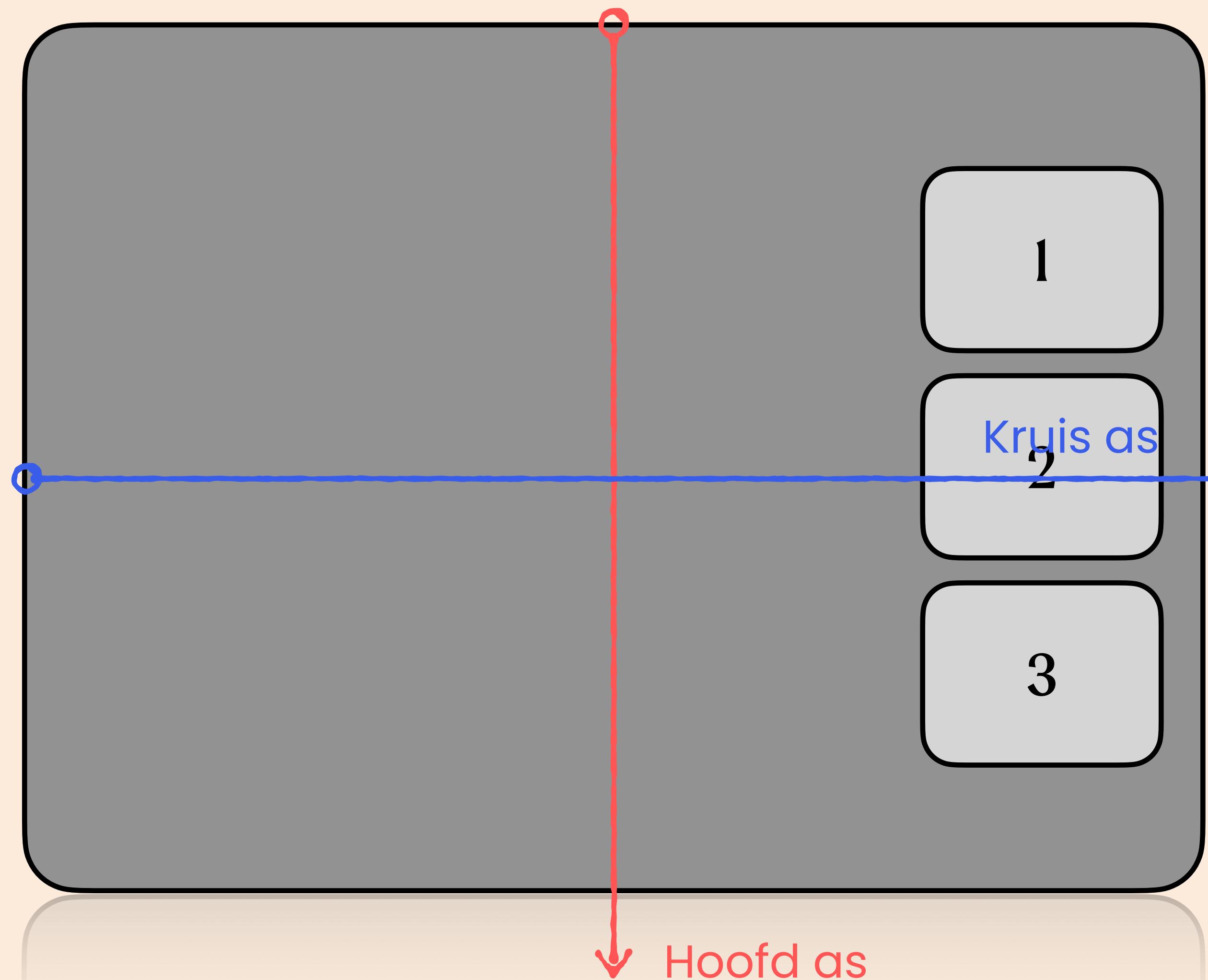
# Flexbox is science



```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: flex-end;  
}
```

**align-items**  
is een kruis-as  
property

# Flexbox is science



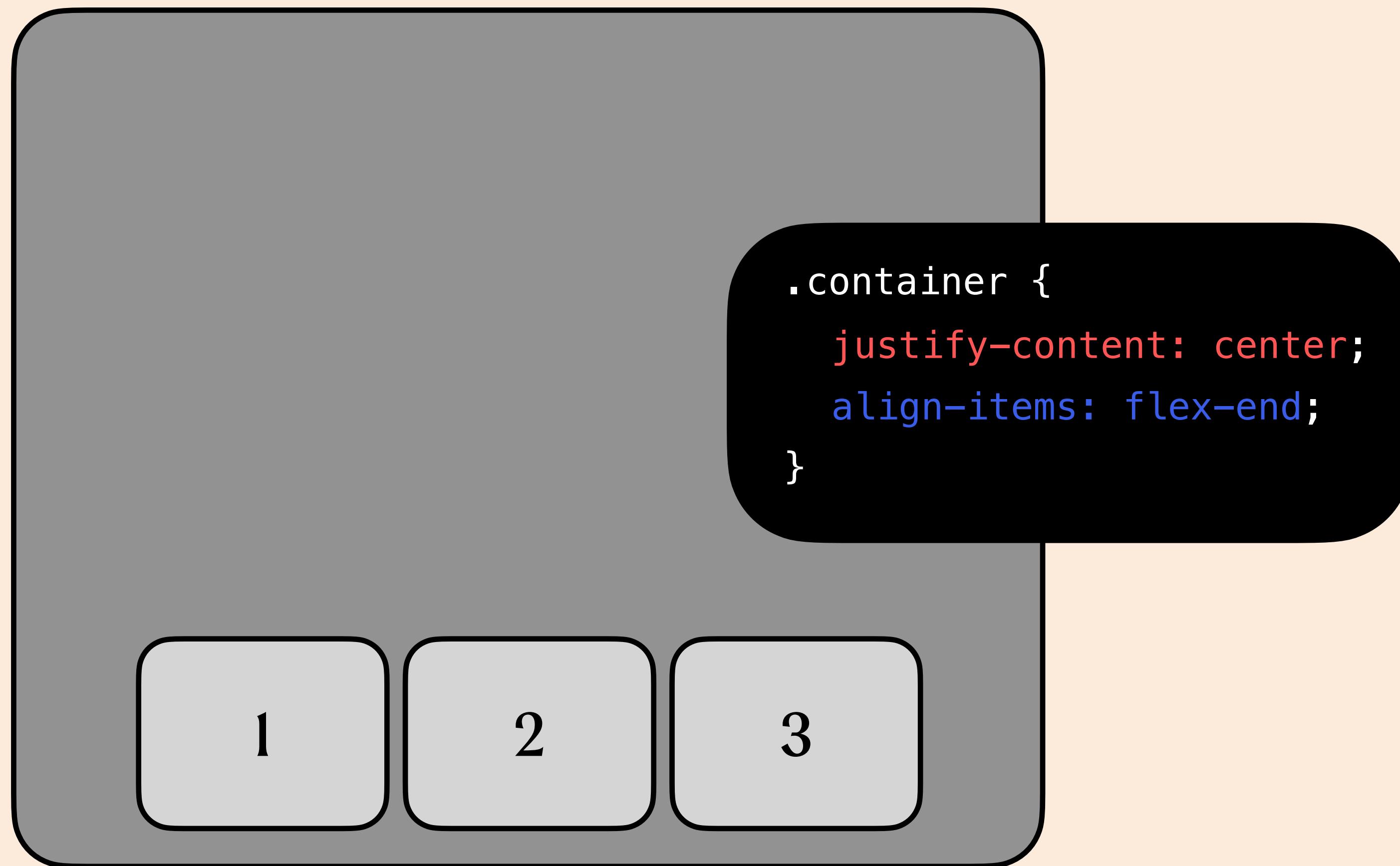
```
.container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: flex-end;  
}
```

**align-items**  
is een kruis-as  
property

# Zelfde properties – andere as

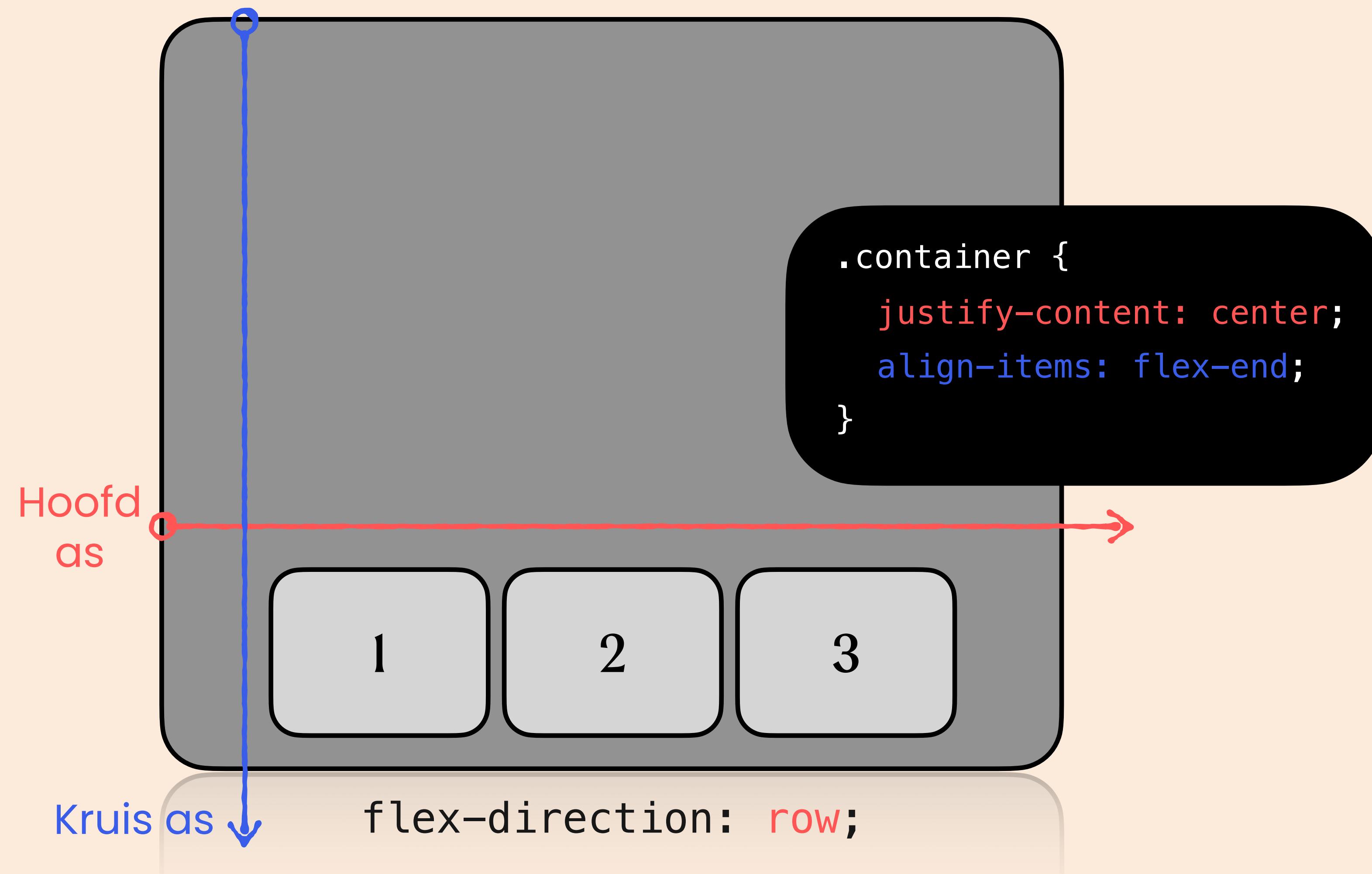
```
.container {  
    justify-content: center;  
    align-items: flex-end;  
}
```

# Zelfde properties - andere as

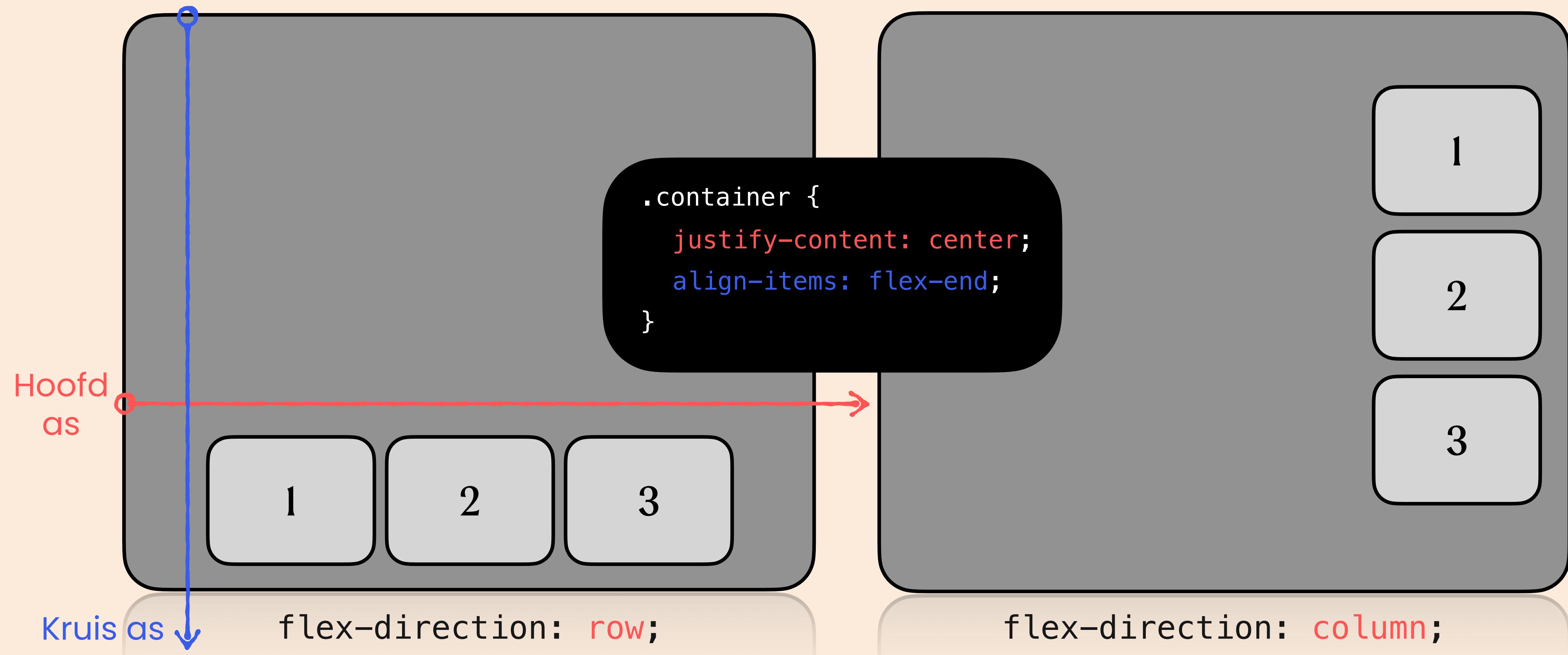


flex-direction: **row**;

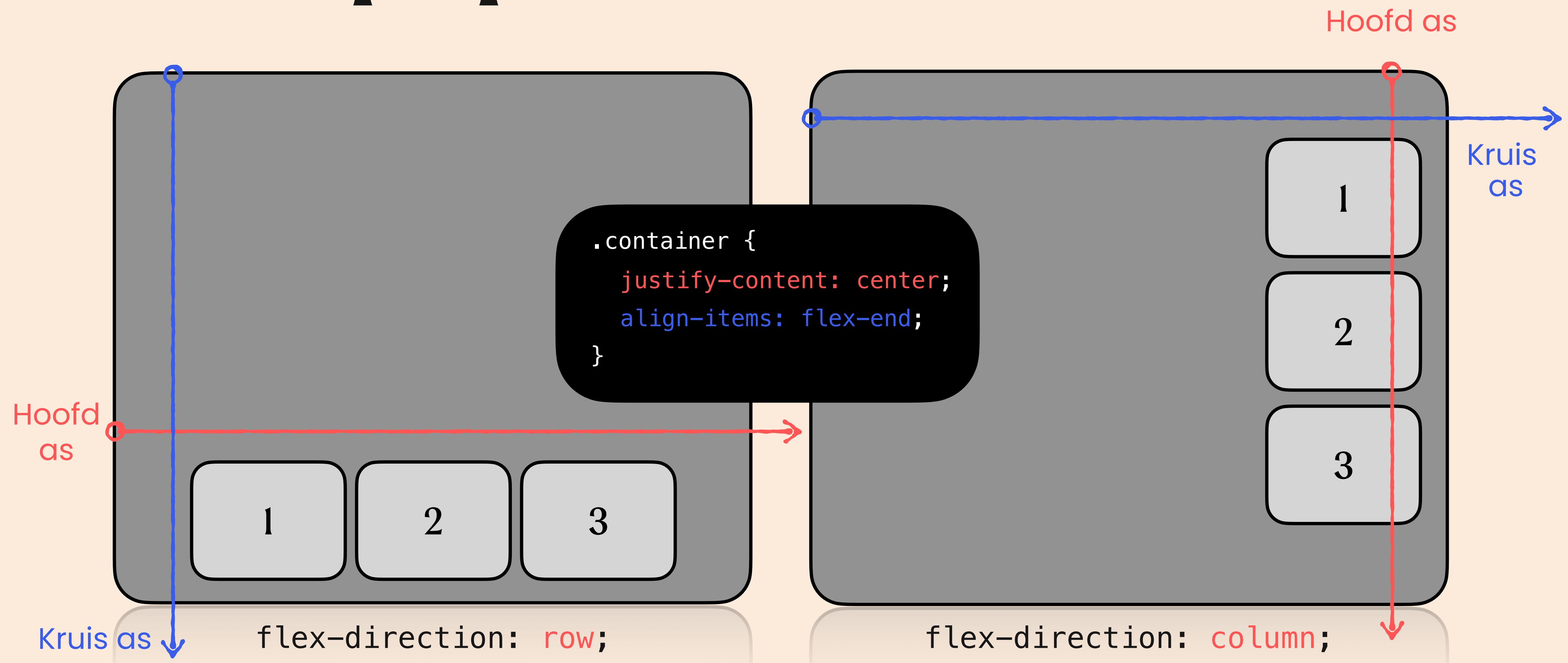
# Zelfde properties - andere as



# Zelfde properties - andere as



# Zelfde properties - andere as



# Demo!

# Flex-items

Hoe je ze de juiste grootte geeft

# Flex-items

Hoe je ze de juiste grootte geeft

→ Het is **best practise** om flexbox items een **flex-basis** property te geven als je de grootte wil beïnvloeden.

# Flex-items

Hoe je ze de juiste grootte geeft

- Het is **best practise** om flexbox items een **flex-basis** property te geven als je de grootte wil beïnvloeden.
- **flex-basis** neemt de plek van de **width** of **height** property over, afhankelijk van de **flex-direction** (richting van de hoofd-as)

# Flex-items

Hoe je ze de juiste grootte geeft

- Het is **best practise** om flexbox items een **flex-basis** property te geven als je de grootte wil beïnvloeden.
- **flex-basis** neemt de plek van de **width** of **height** property over, afhankelijk van de **flex-direction** (richting van de hoofd-as)
- Als de container geen **flex-basis** property op het item vindt én geen **width** of **height** property, zullen de items de grootte aannemen van de content die erin zit.

# Flex-items

Vergroten en verkleinen op basis van de container

# Flex-items

Vergroten en verkleinen op basis van de container

→ **flex-basis** zorgt ervoor dat we ook kunnen bepalen in hoeverre items mogen **krimpen** als er weinig plek is - of **vergroten** als er plek over is in de container

# Flex-items

Vergroten en verkleinen op basis van de container

- **flex-basis** zorgt ervoor dat we ook kunnen bepalen in hoeverre items mogen **krimpen** als er weinig plek is - of **vergroten** als er plek over is in de container
- Het krimpen bepaal je met **flex-shrink** uitgedrukt in eenheden (indien afwezig standaard 1)

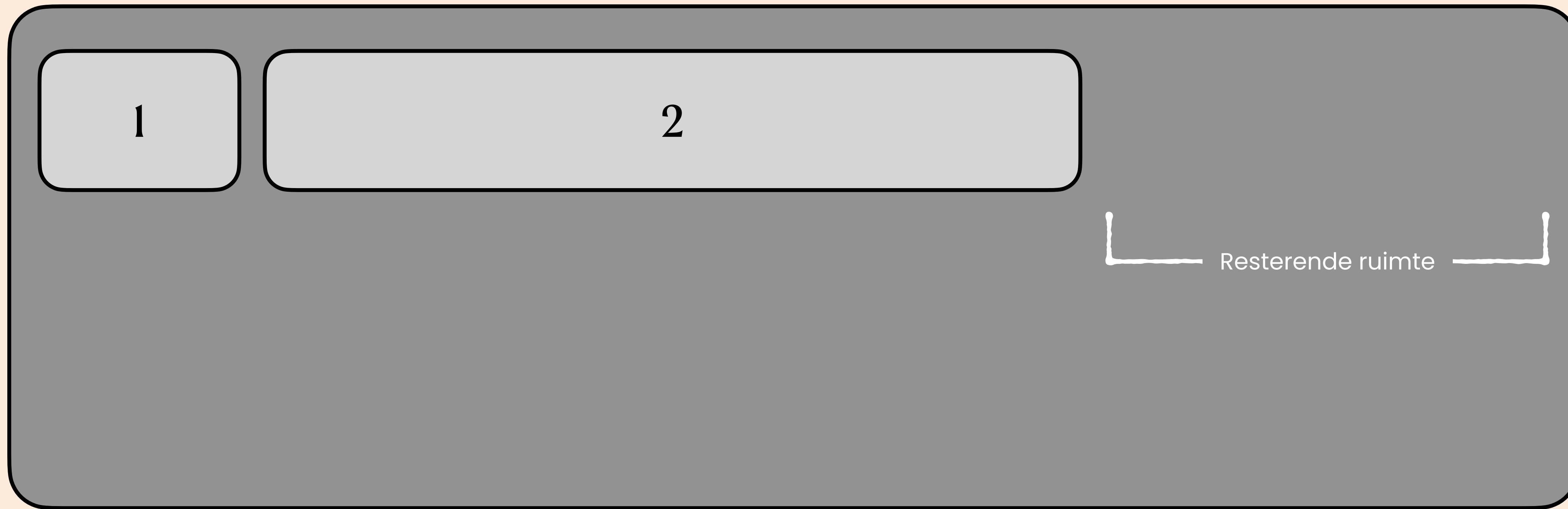
# Flex-items

Vergroten en verkleinen op basis van de container

- **flex-basis** zorgt ervoor dat we ook kunnen bepalen in hoeverre items mogen **krimpen** als er weinig plek is - of **vergroten** als er plek over is in de container
- Het krimpen bepaal je met **flex-shrink** uitgedrukt in eenheden (indien afwezig standaard **1**)
- Het groeien bepaal je met **flex-grow** uitgedrukt in eenheden (indien afwezig standaard **0**)

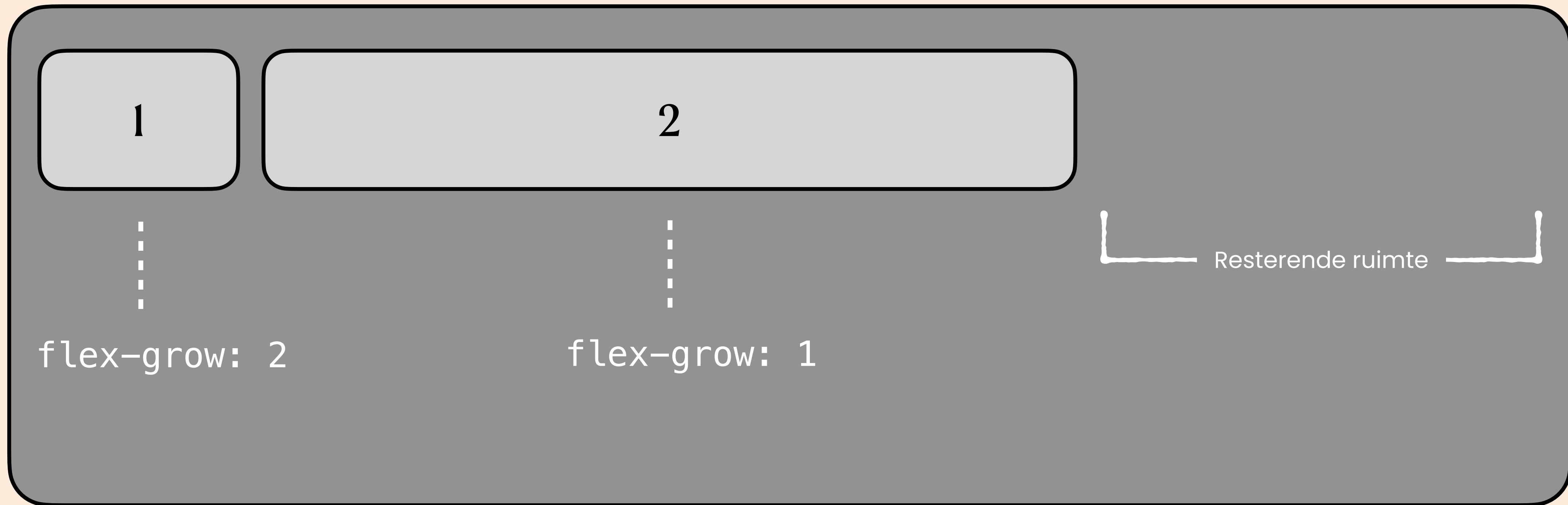
# Grow & shrink eenheden

Expectation...



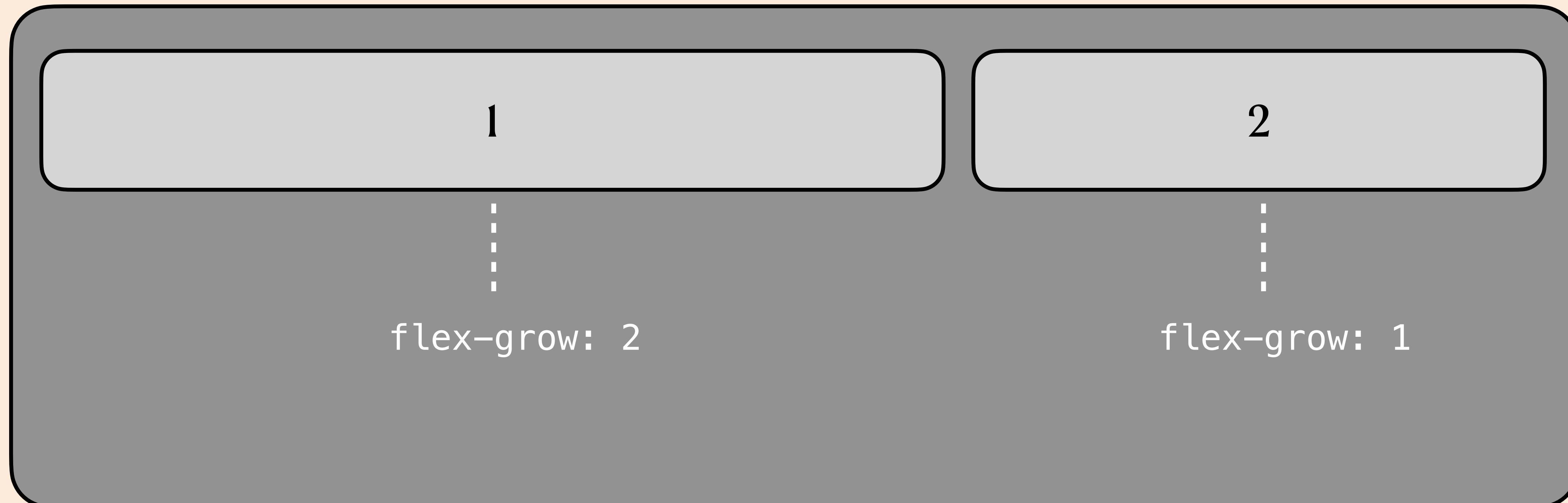
# Grow & shrink eenheden

Expectation...



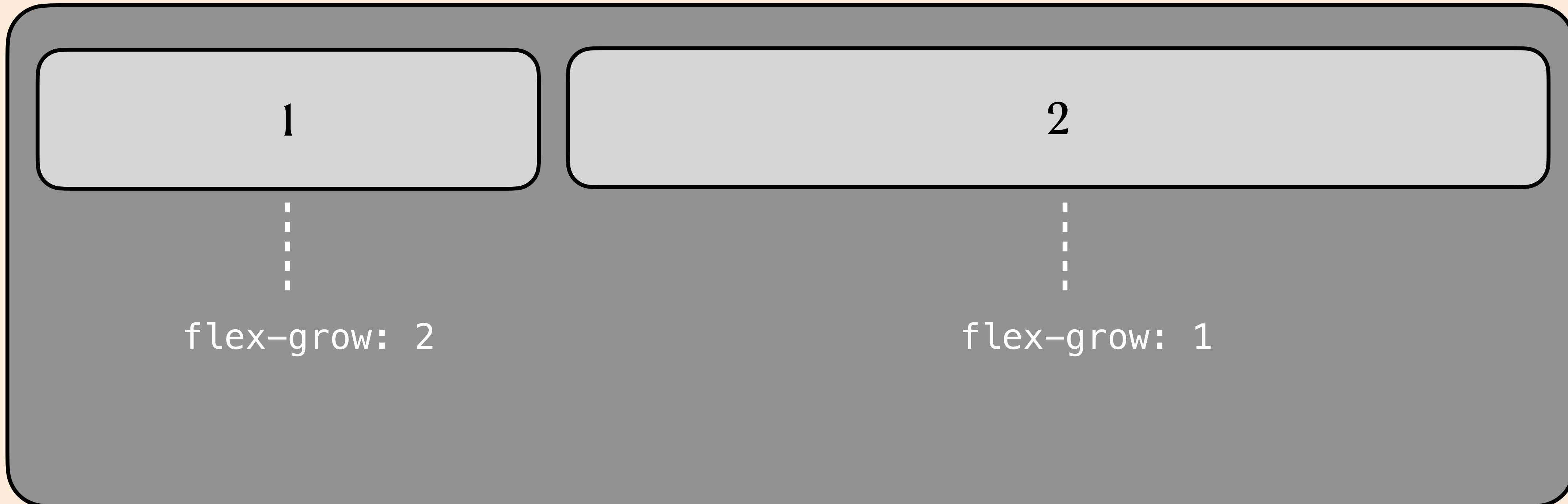
# Grow & shrink eenheden

Expectation...



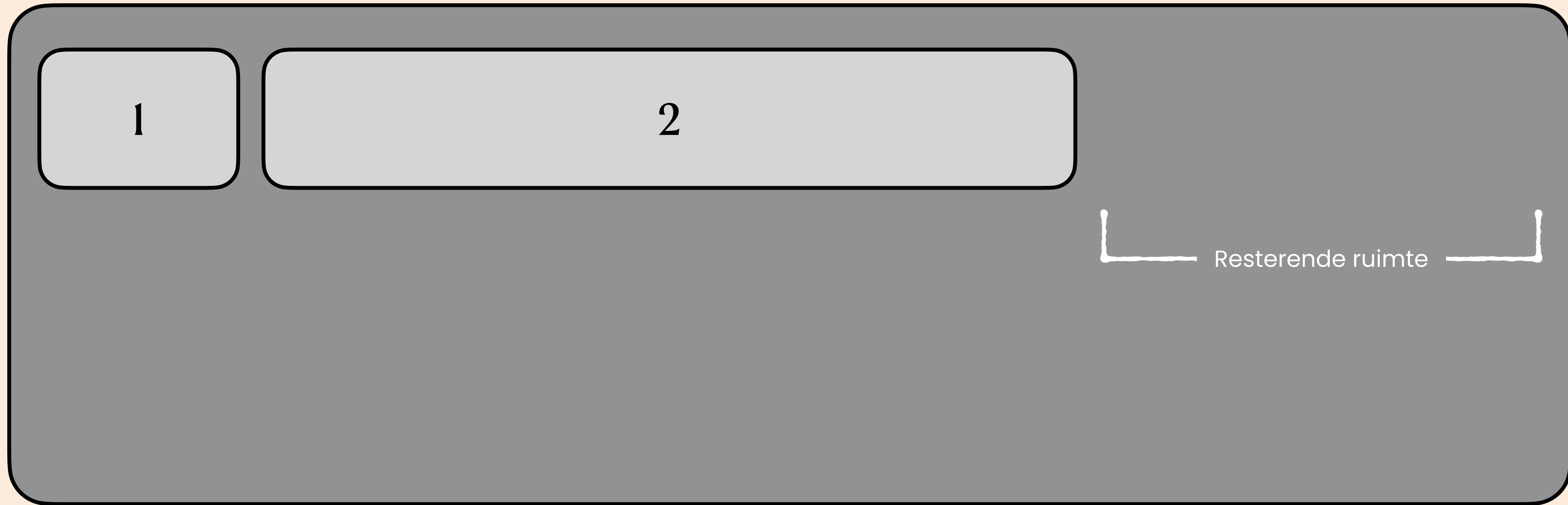
# Grow & shrink eenheden

Expectation...



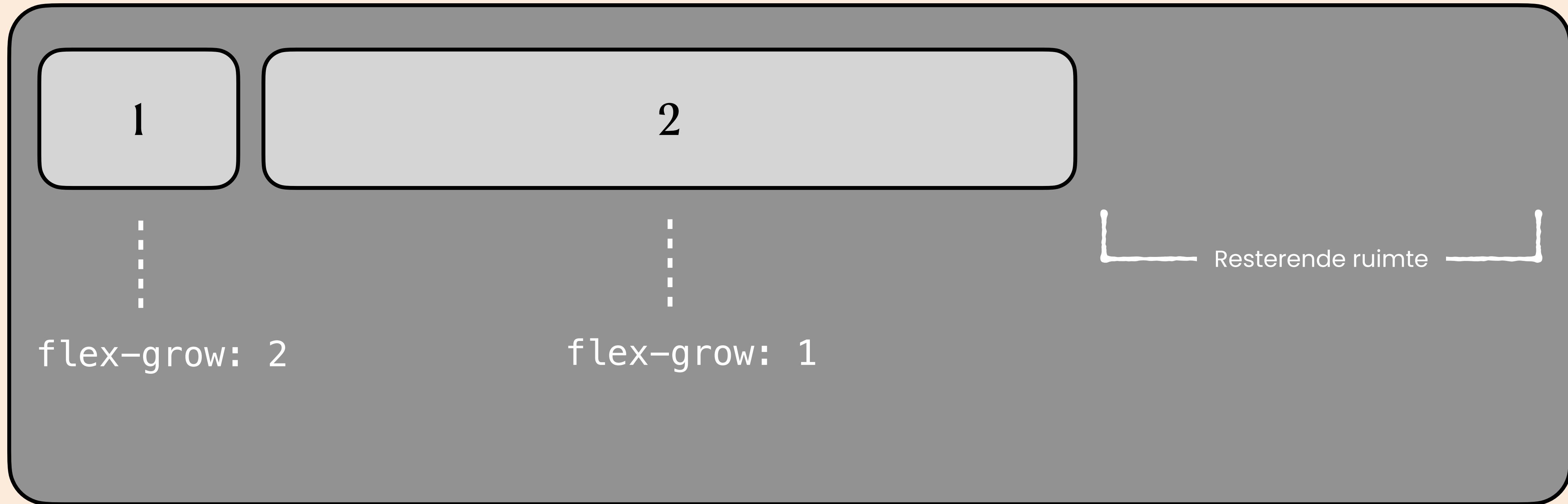
# Grow & shrink eenheden

Hoe het écht werkt



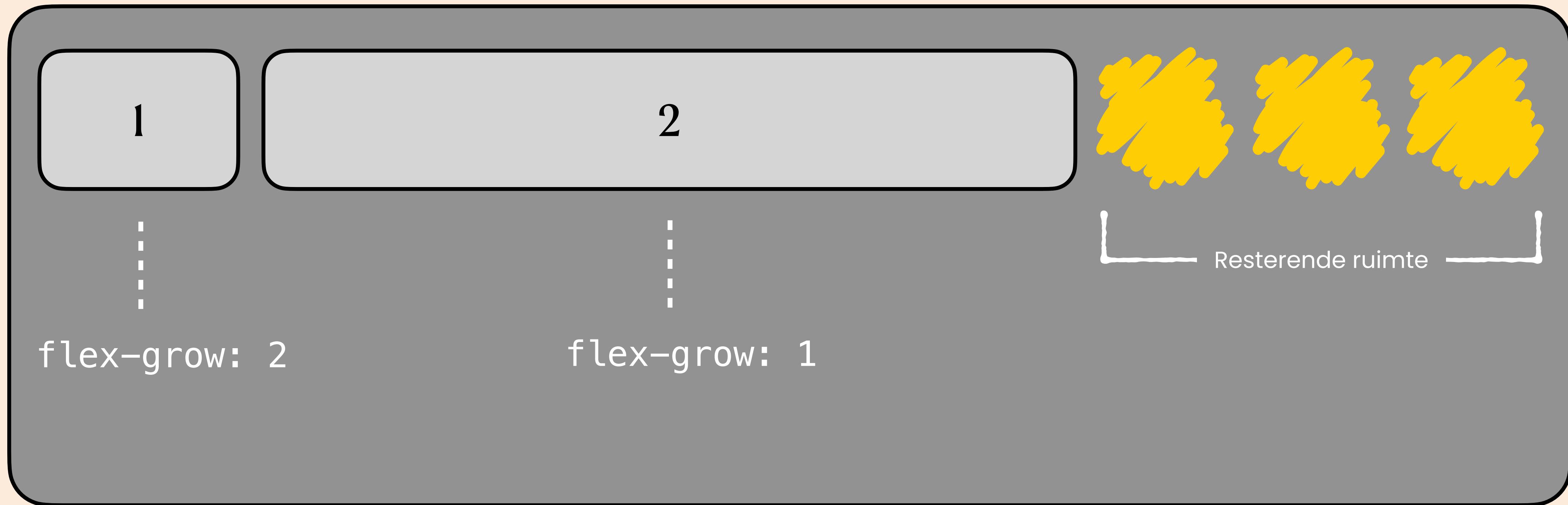
# Grow & shrink eenheden

Hoe het écht werkt



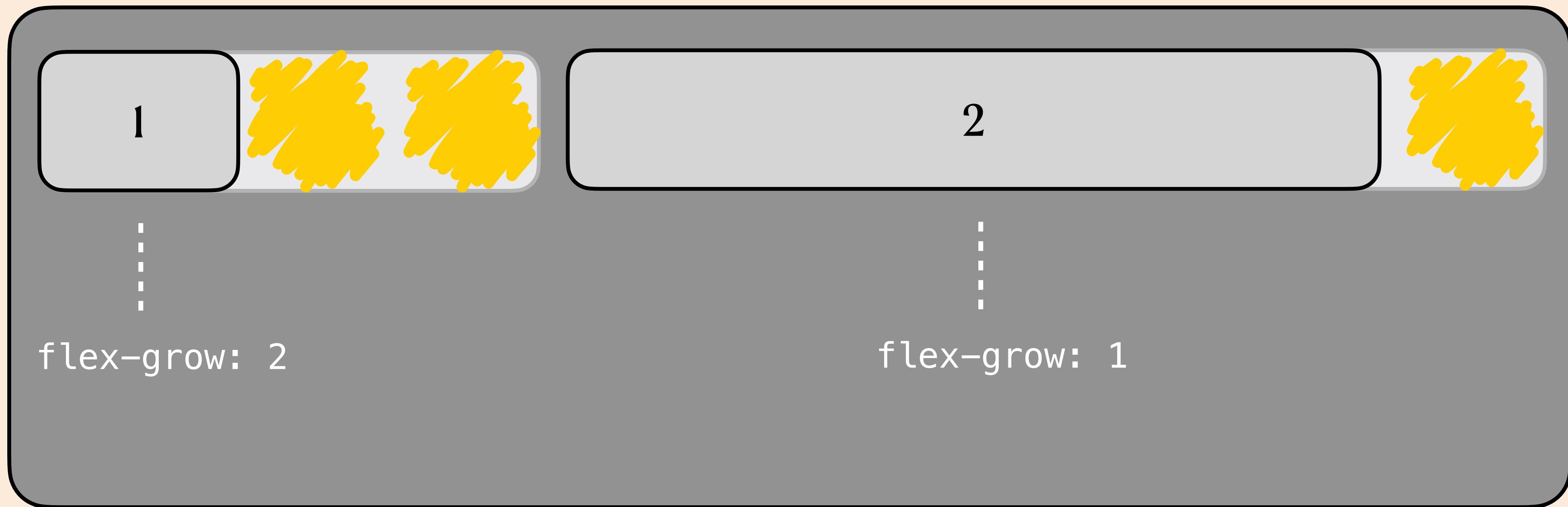
# Grow & shrink eenheden

Hoe het écht werkt



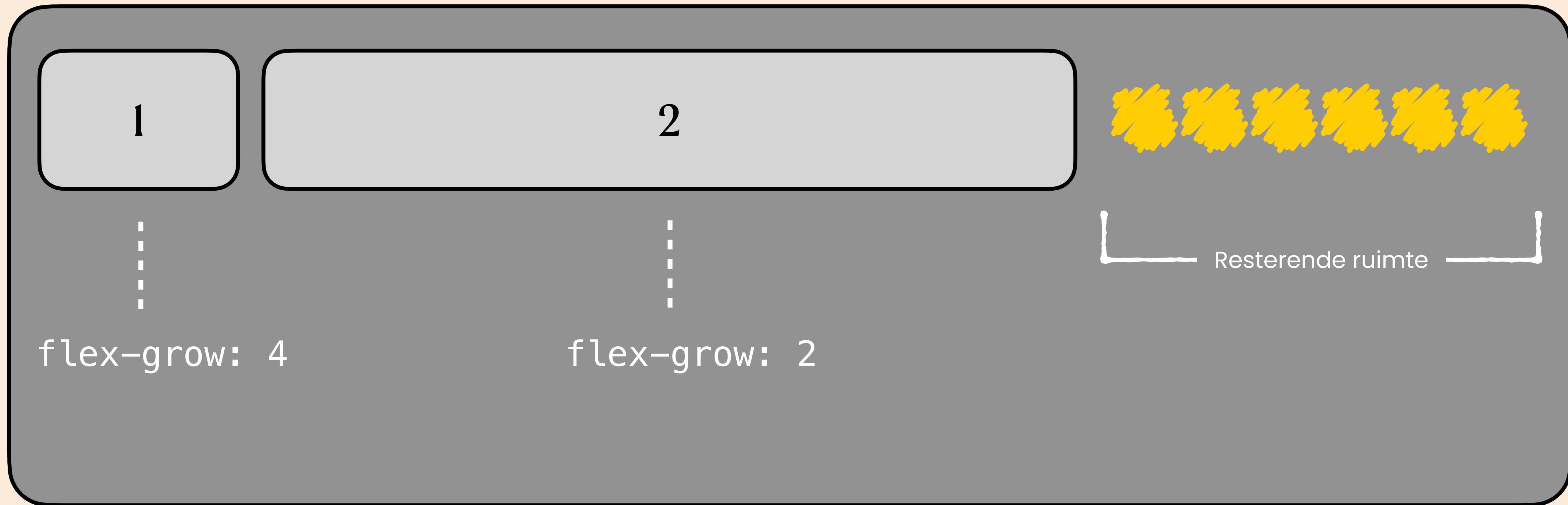
# Grow & shrink eenheden

Hoe het écht werkt



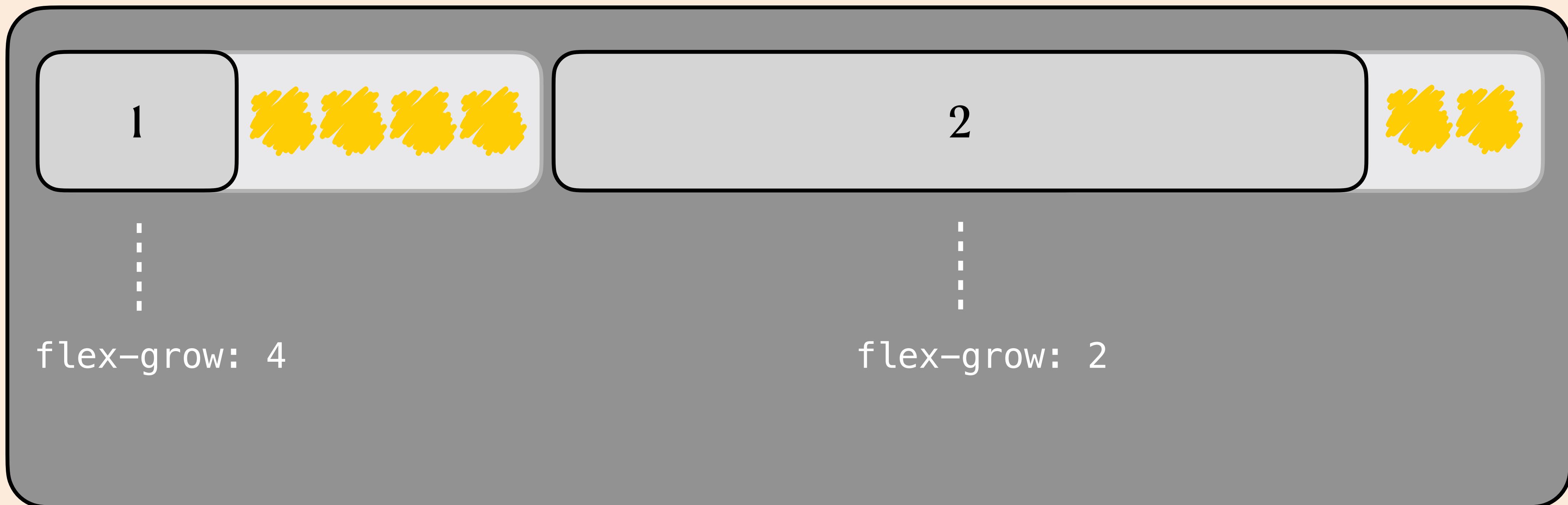
# Grow & shrink eenheden

Hoe het écht werkt



# Grow & shrink eenheden

Hoe het écht werkt



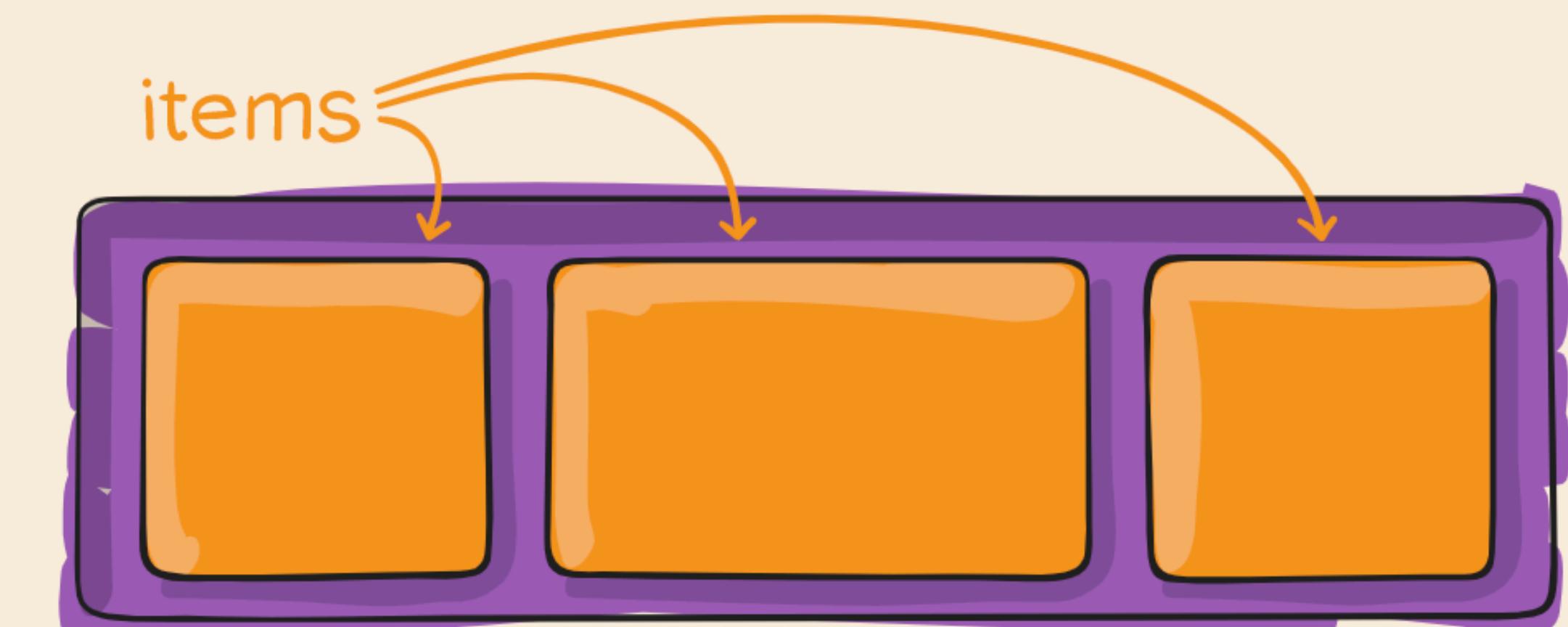
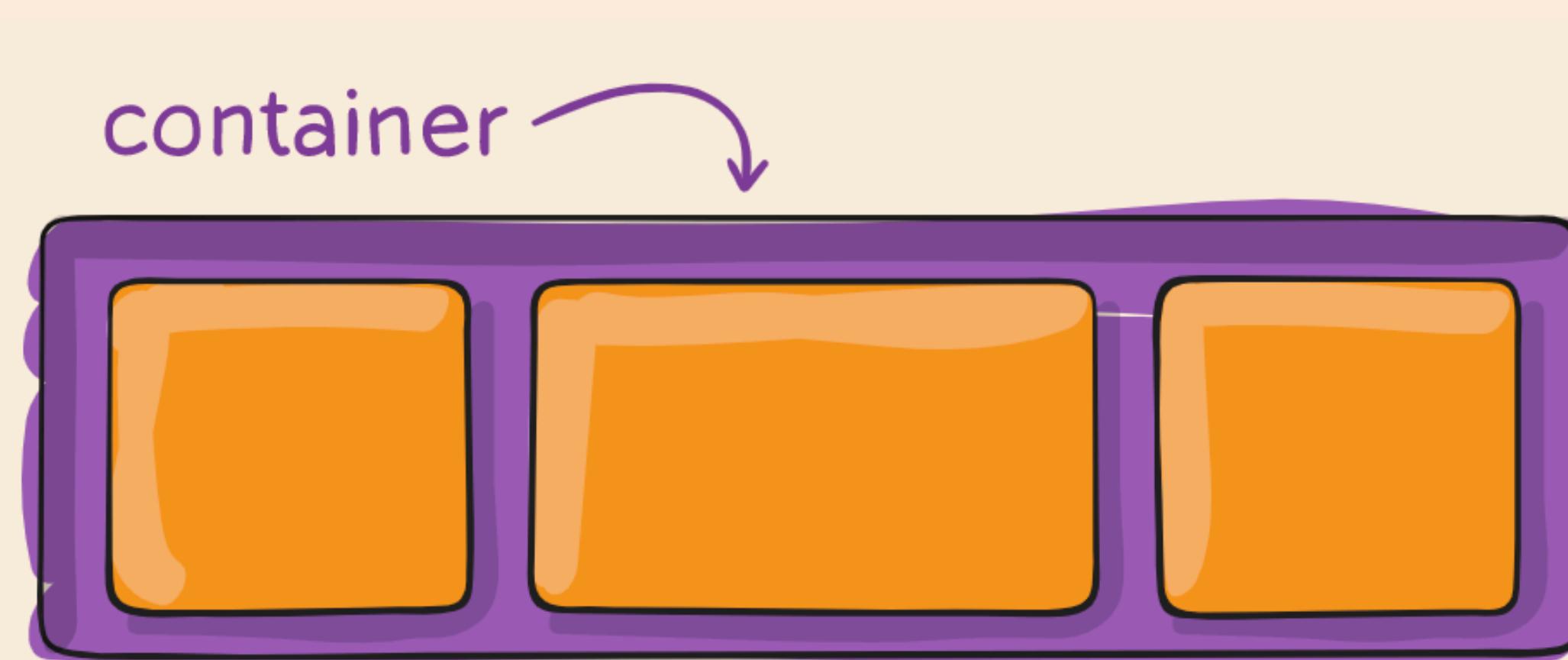
# Flexbox cheatsheet

In het begin ken je de properties natuurlijk nog niet uit je hoofd

# Flexbox cheatsheet

In het begin ken je de properties natuurlijk nog niet uit je hoofd

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



# Tijd voor een demo!

- Flexbox layout demonstratie
- Het gebruik van de `flex-basis` / `flex-grow` / `flex-shrink` properties
- Flex-items kunnen ook een flex-container zijn

# Huiswerk volgende les

- Maak de opdracht uit de les af en lever deze uiterlijk woensdag 25 mei in via Teams. Dit is een **peer-review** opdracht waarbij je gekoppeld wordt aan een medestudent. Je levert de link naar jouw GitHub repository in, dus geen echte bestanden!
- Lees voor volgende les hoofdstuk 7.5 t/m 8 (HTML & CSS).  
+ *Als je meer verdieping wil, is hoofdstuk 8.7 t/m 9 bonus!*

Wil je nog veel meer oefenen met Flexbox? **Dat kan!**

<https://github.com/hogeschoolnovi/frontend-html-css-extra-practise>

45 min

# Opdracht - Flexbox

<https://github.com/hogeschoolnovi/frontend-selectors-flexbox>

- Werk verder in hetzelfde project als de vorige opdracht
- Je wordt ingedeeld in breakout rooms
- Lees de opdracht beschrijving voor opdracht 2
- Overleg zo veel mogelijk met je groepje, maar tag de docent in de breakout-chat wanneer je vastloopt of ergens extra uitleg over wil!