

#4 FLEXBOX

intro

Dé bron der bronnen

Het naslagwerk met alles over Flexbox:

css-tricks.com/snippets/css/a-guide-to-flexbox

The image shows a screenshot of the CSS-Tricks website article titled "A Complete Guide to Flexbox" by Chris Coyier. The page has a dark header with the site logo and navigation links. The main content area is white and contains a table of contents with sections like "Background", "Basics and terminology", and "Flexbox properties". Under "Flexbox properties", there are two diagrams: "Properties for the Parent (flex container)" and "Properties for the Children (flex items)". The parent diagram shows a purple box containing three orange boxes, with a purple arrow pointing to the purple box labeled "container". The child diagram shows the same purple box with three orange boxes inside, with orange arrows pointing to the orange boxes labeled "items".

Annotations on the screenshot:

- A vertical label "Parent: flex container" on the left side with an arrow pointing to the parent diagram.
- A vertical label "Child: flex items" on the right side with an arrow pointing to the child diagram.

Page content:

* CSS-TRICKS ARTICLES VIDEOS ALMANAC NEWSLETTER GUIDES BOOKS

HOME / GUIDES /

A Complete Guide to Flexbox

Chris Coyier on Apr 8, 2013 (Updated on Aug 16, 2021)

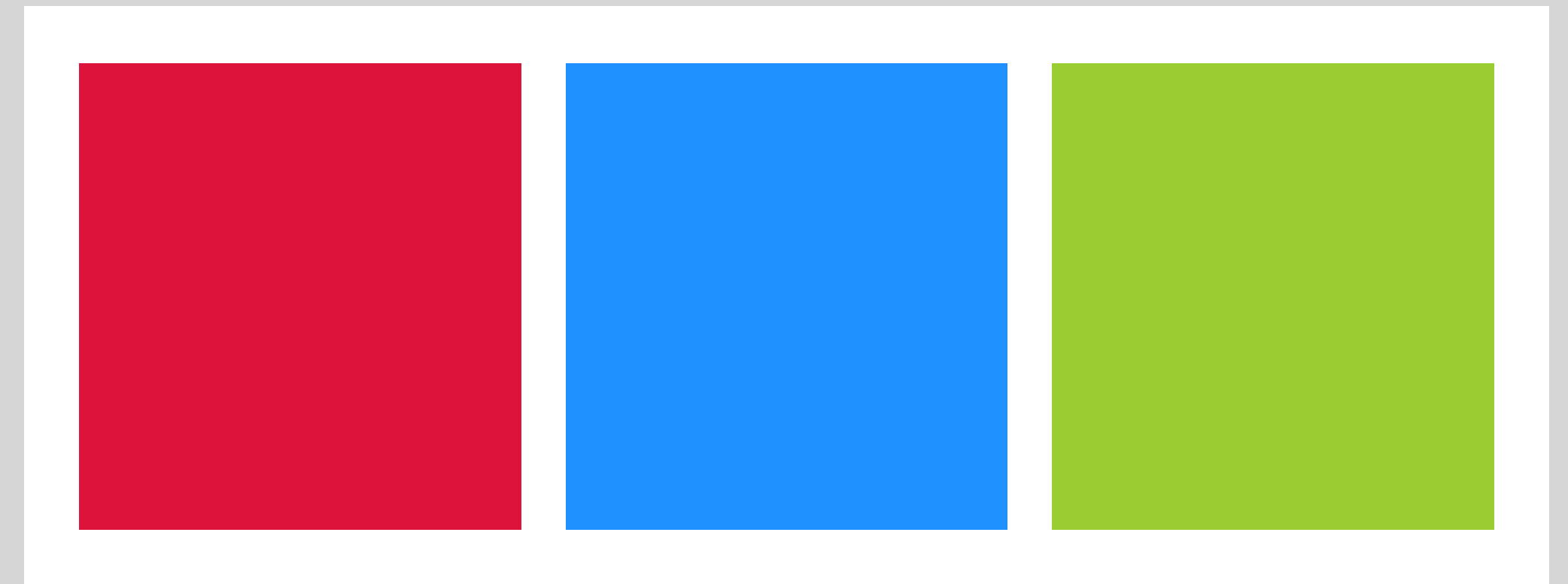
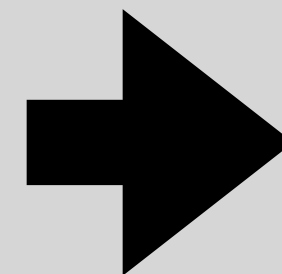
Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.

- ▶ Background
- ▶ Basics and terminology
- ▼ Flexbox properties
 - Properties for the Parent (flex container)
 - Properties for the Children (flex items)

Items naast elkaar flex-en :-)

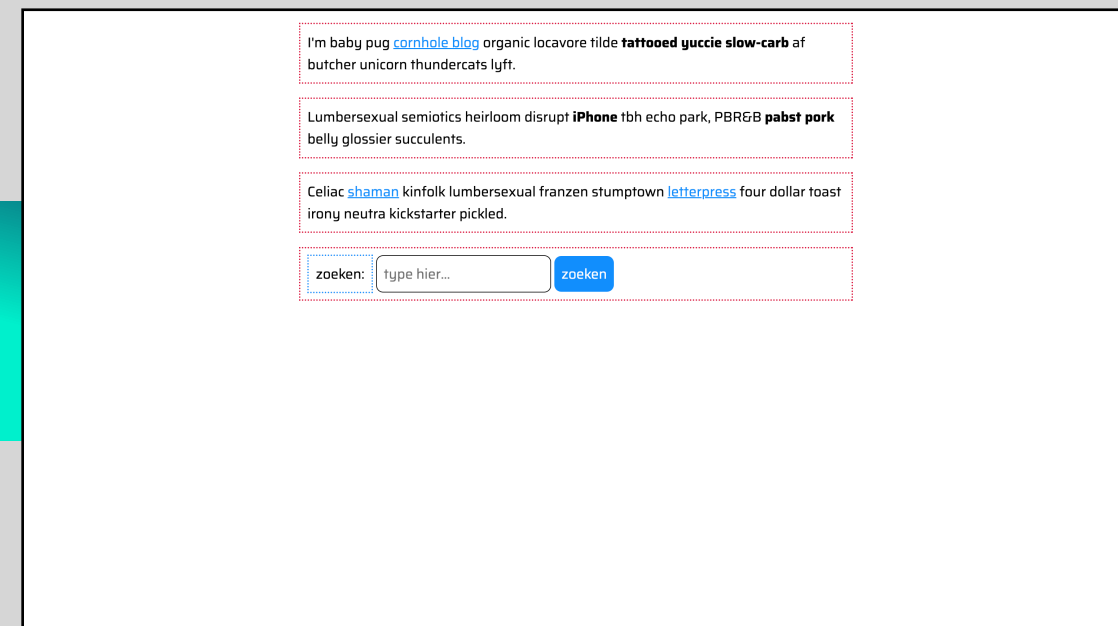


Block elementen staan van nature onder elkaar



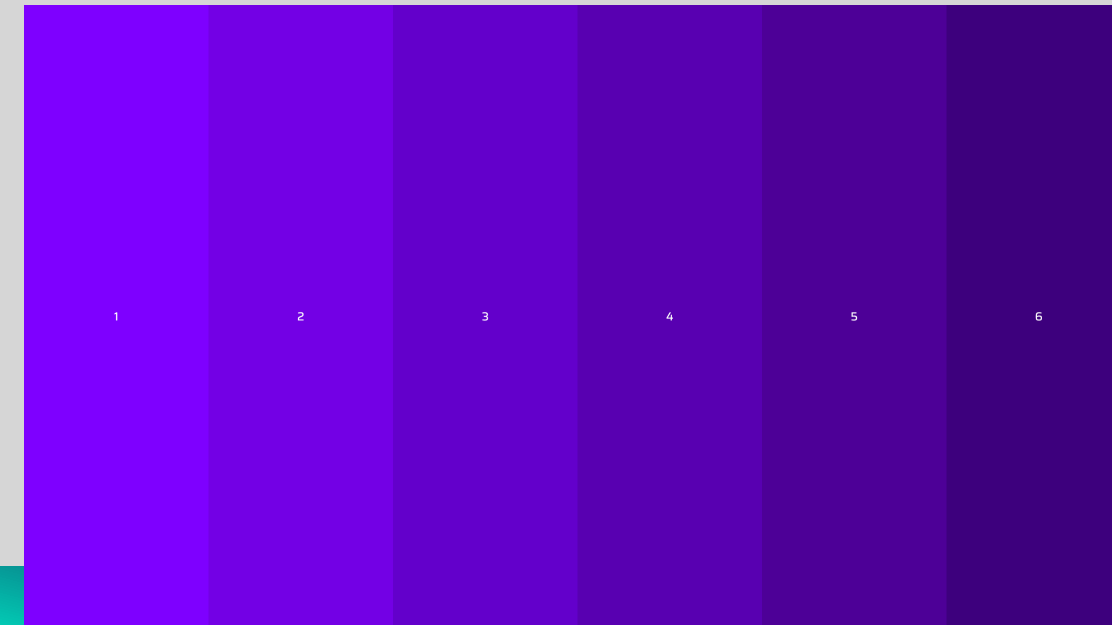
Met flexbox kun je die naast elkaar zetten
(...en nog veel meer :-)

Layout overview

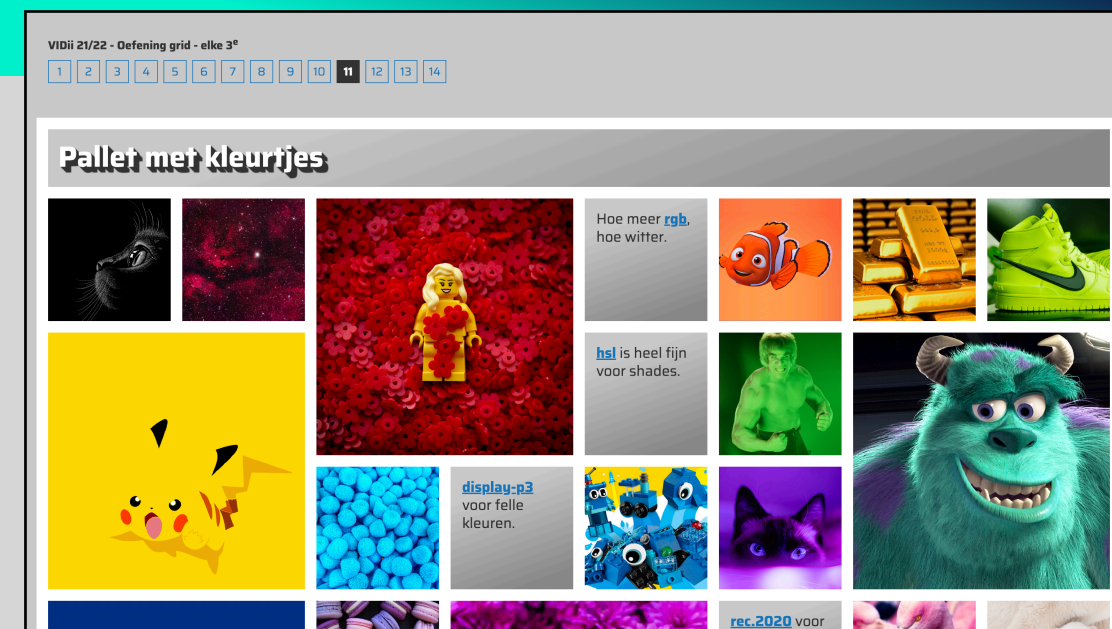


Standaard: flow layout
Block-elementen onder elkaar
Inline-elementen naast elkaar

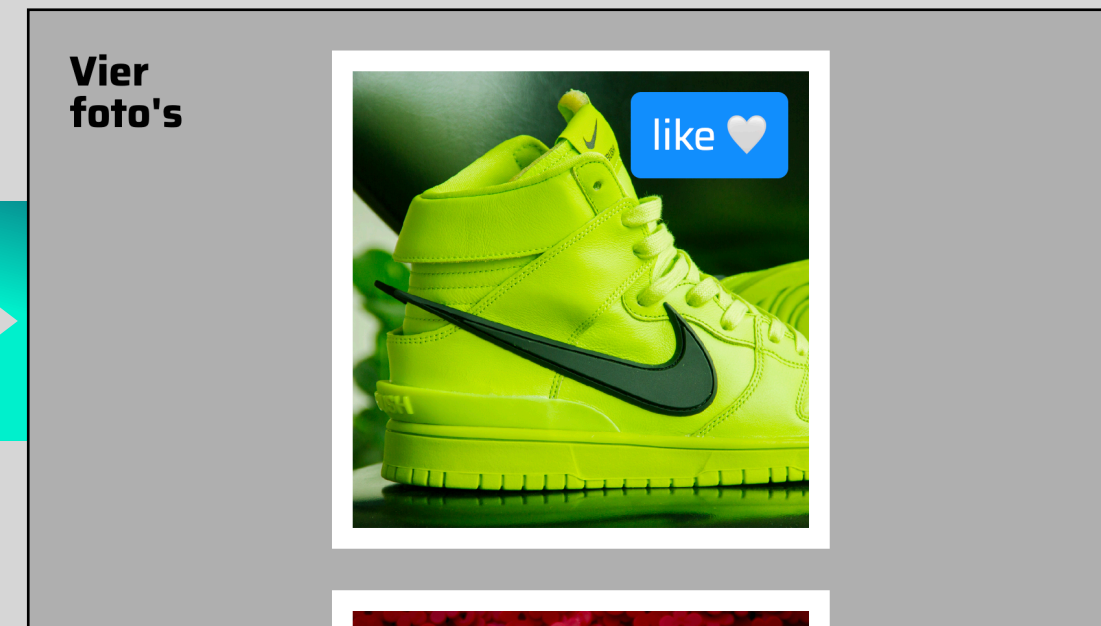
Display flex
Elementen naast elkaar zetten. De kinderen vullen de parent flexibel (over meer rijen indien nodig).



Vandaag



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



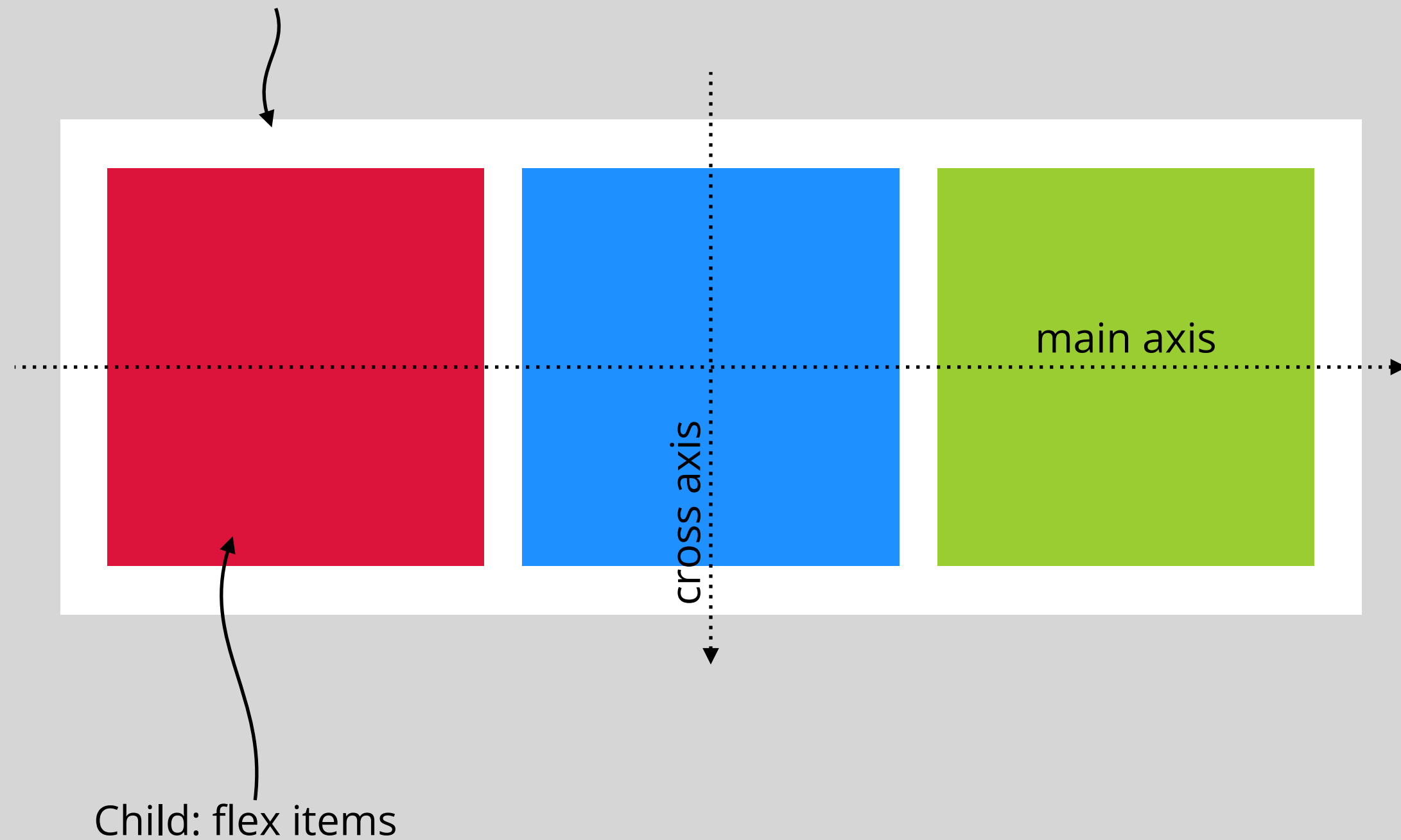
Position: absolute, fixed en sticky
Elementen op een specifieke plek in een parent of in het scherm zetten.

Margin, padding, translate, transform...

Last resort
(vaak beetje viezig 🤢)

Intro flexbox 1/6

Om Flexbox toe te passen heb je dus een ouder (container) en kinderen (items) nodig.



Daarnaast zul je regelmatig de termen main axis (standaard horizontaal) en cross axis (standaard verticaal) tegenkomen.

Intro flexbox 2/6

Standaard staan elementen onder elkaar.

De HTML

```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

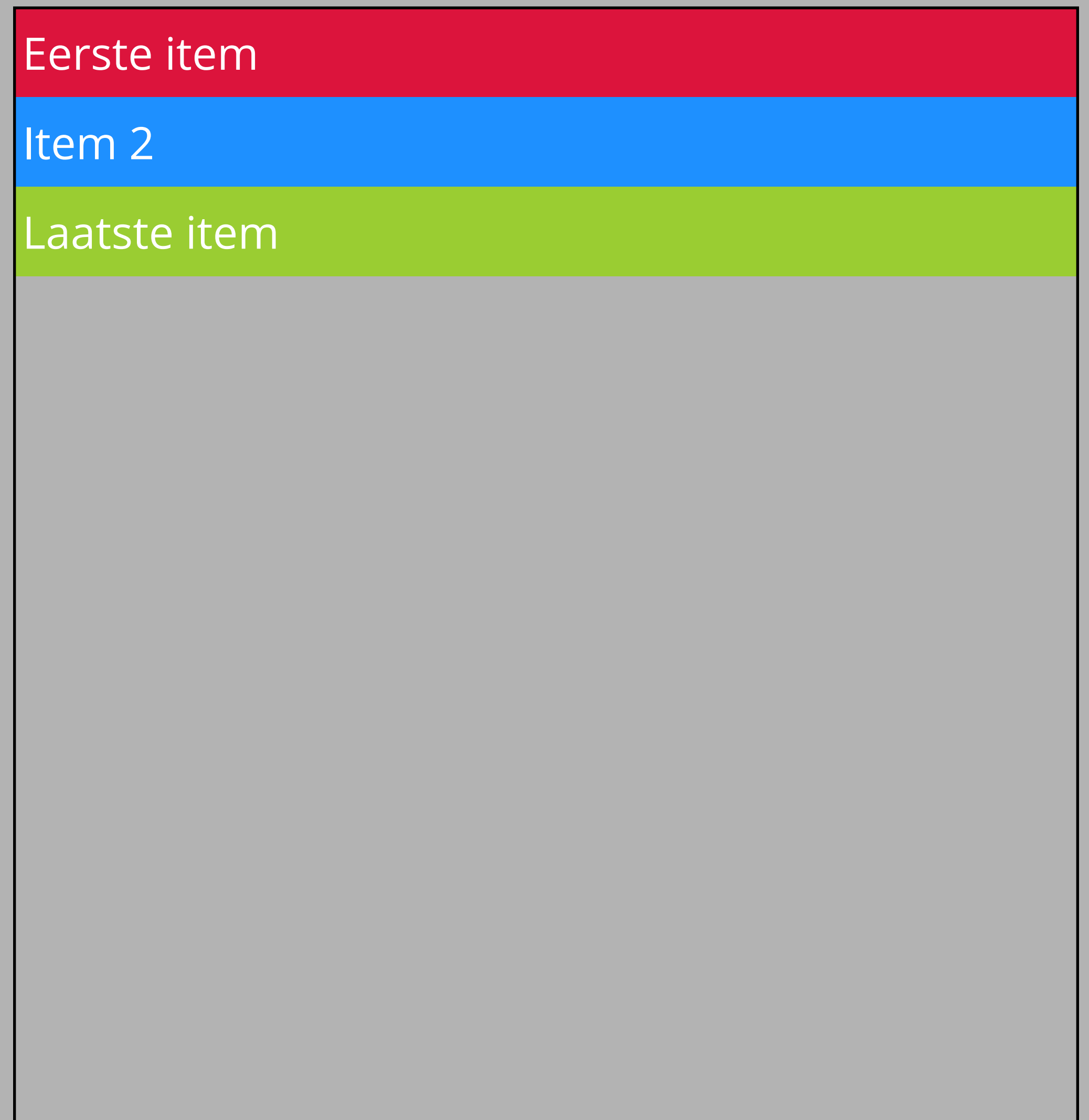
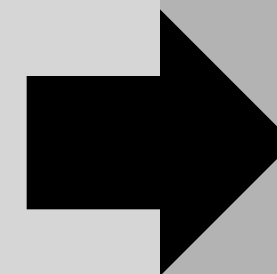
De CSS

```
main {
  height:100vh;
}

article {
  color:white;
  background-color:crimson;
}

article:nth-of-type(2) {
  background-color:dodgerblue;
}

article:last-of-type {
  background-color:yellowgreen;
}
```



Intro flexbox 3/6

Met Flexbox kun je elementen naast elkaar zetten.

Daarvoor zeg je tegen de ouder (hier de <main>) dat zijn kinderen moeten flexen.

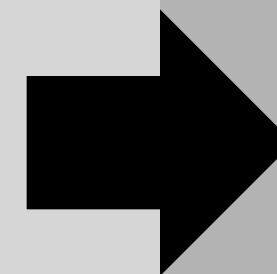
De HTML

```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

De CSS

```
main {
  display: flex;
}
```

De kinderen (hier de <article>s) staan daarmee naast elkaar. Horizontaal worden ze zo breed als hun eigen content. Verticaal worden ze zo hoog als de container (de <main>).



Intro flexbox 4/6

Om de kinderen samen de hele breedte in te laten nemen, kun je Flex-grow gebruiken.

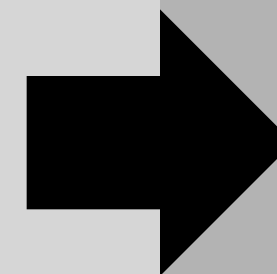
De HTML

```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

De CSS

```
main {
  display: flex;
}

article {
  flex-grow: 1;
}
```



Intro flexbox 5/6

Om de kinderen niet de hele hoogte te laten innemen, kun je align-items gebruiken.

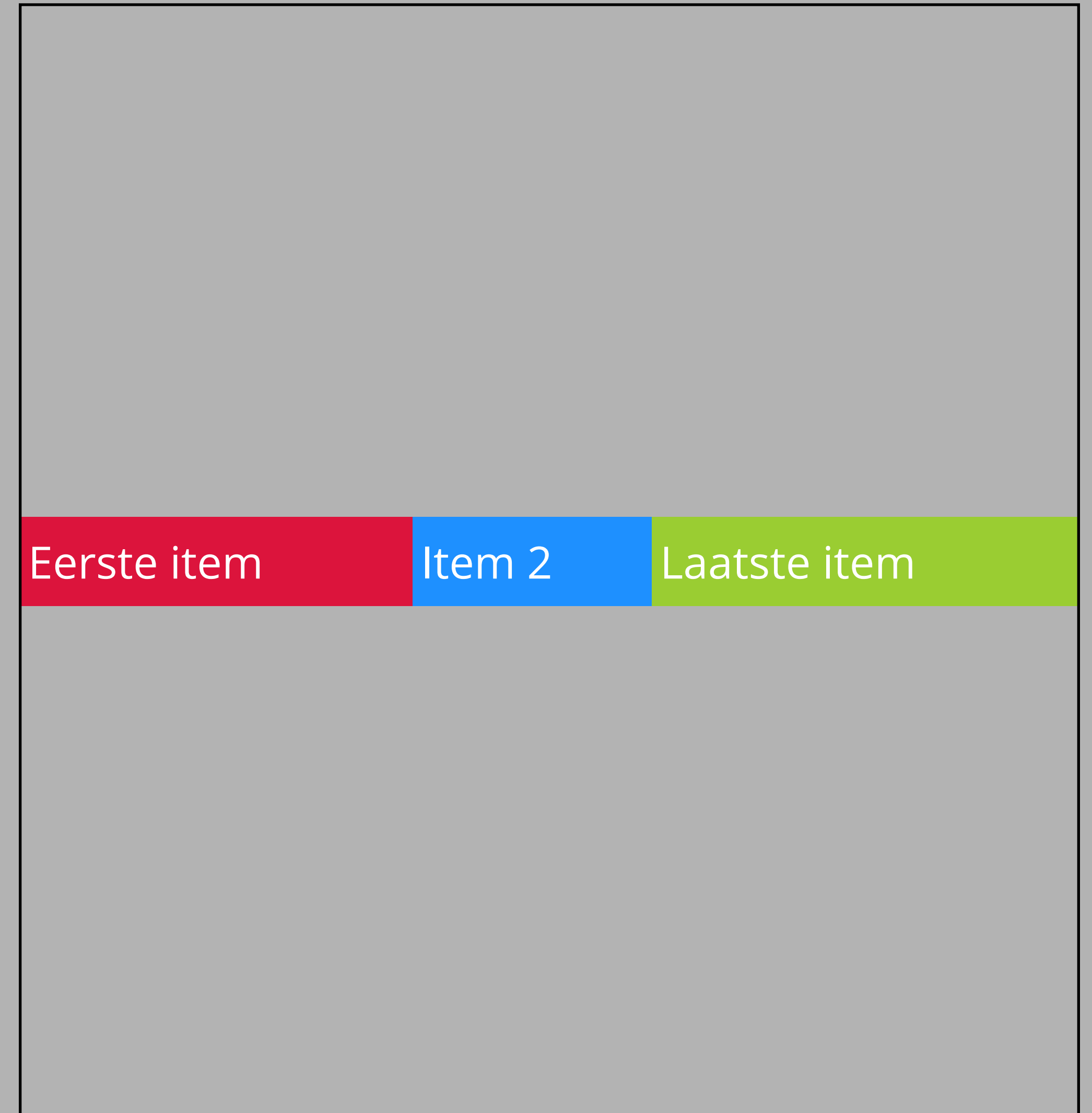
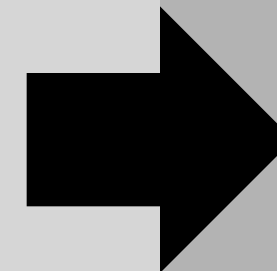
De HTML

```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

De CSS

```
main {
  display:flex;
  align-items:center;
}

article {
  flex-grow:1;
}
```



Intro flexbox 6/6

En met justify-content kun je de kinderen ook horizontaal in het midden zetten. Dat was voor Flexbox erg lastig.

De HTML

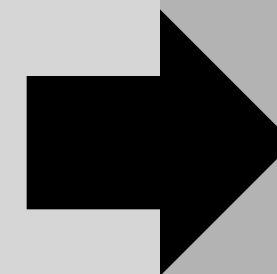
```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

De CSS

```
main {
  display: flex;
  justify-content: center;
  align-items: center;
}

article { }
```

En zo kan er nog veel meer. Daar gaan we deze week mee oefenen.



Eerste item Item 2 Laatste item

Leer
flexbox van
Wes Bos

WHAT THE FLEXBOX?!

A simple, free 20 video course that will help you master CSS Flexbox!

Email Address

Join the Course →

I agree to the [Terms](#) and [Privacy Policy](#).

Get access to all 20 videos and download the exercises!

Join 102,000+ others in learning with Wes Bos!

Wes Bos:
flexbox.io



Flexbox or grid - How to decide?

YouTube

Flexbox of Grid

YouTube · Kevin Powell · 18 jan 2022



In deze video

- 00:00 Introduction
- 01:13 Comparing their behaviors
- 09:54 When you should use flexbox
- 13:40 When you should use grid
- 17:23 Mixing flexbox and grid

Gerelateerde onderwerpen



CSS Flexible Box Layout



CSS grid layout



Grid



Flexbox vs grid Reddit



Flexbox layout examples

Kevin Powel:

youtu.be/3eIGSZSWTbM

intro!

#1 EΓEXBOX

css challenge

EDND 54-52