

FED 24/25  
**Flexbox**  
**recap**



# Intro flexbox 1/6

Standaard staan elementen onder elkaar.

## De HTML

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

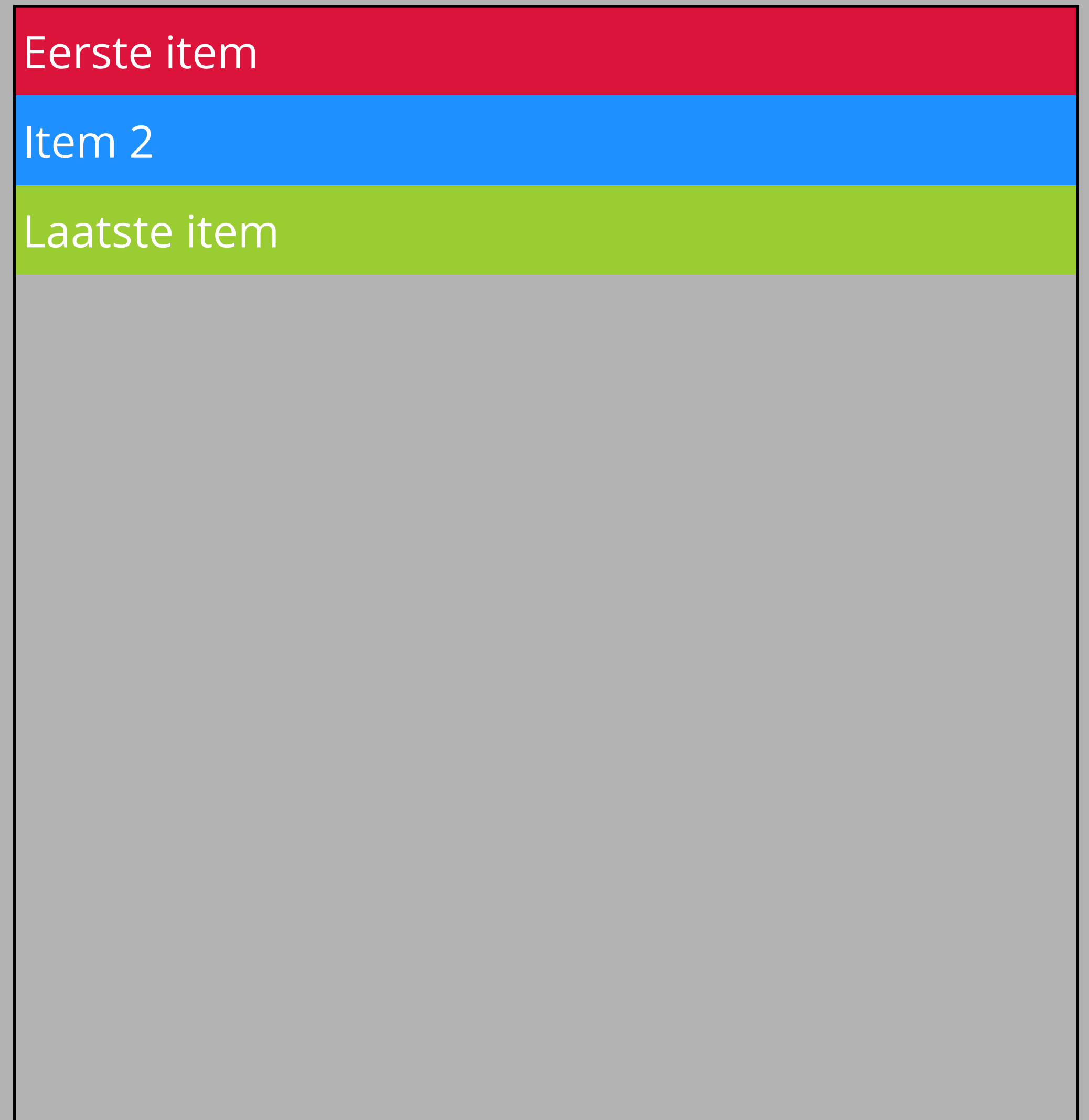
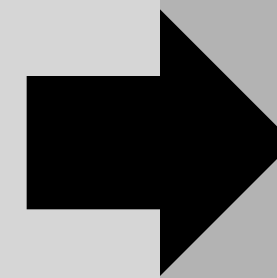
## De CSS

```
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 2/6

Met Flexbox kun je elementen naast elkaar zetten.

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

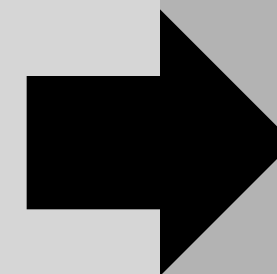
## De HTML

```
<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 3/6

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

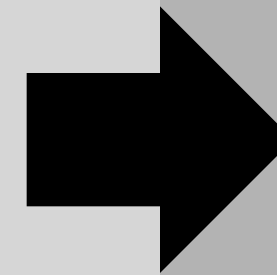
## De HTML

```
<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 4/6

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

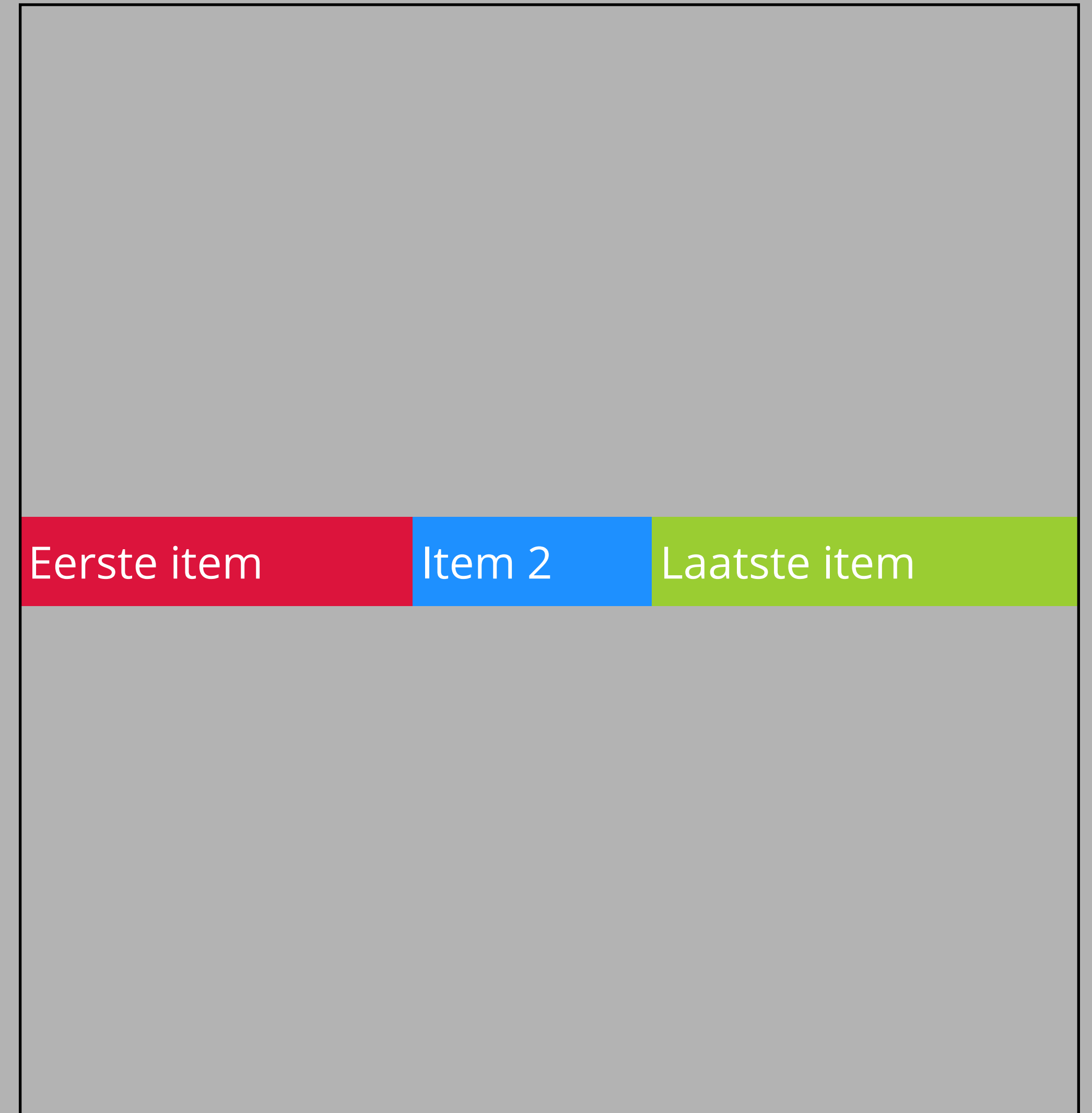
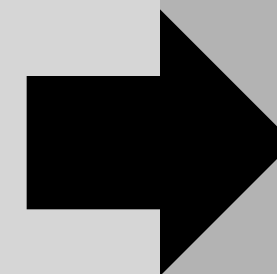
## De HTML

```
<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 5/6

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

## De HTML

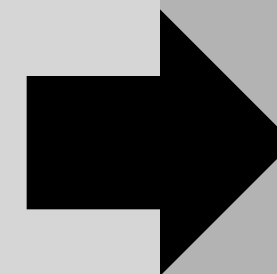
```
<main>
  <article>Eerste item</article>
  <article>Item 2</article>
  <article>Laatste item</article>
</main>
```

## De CSS

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

article { }
```

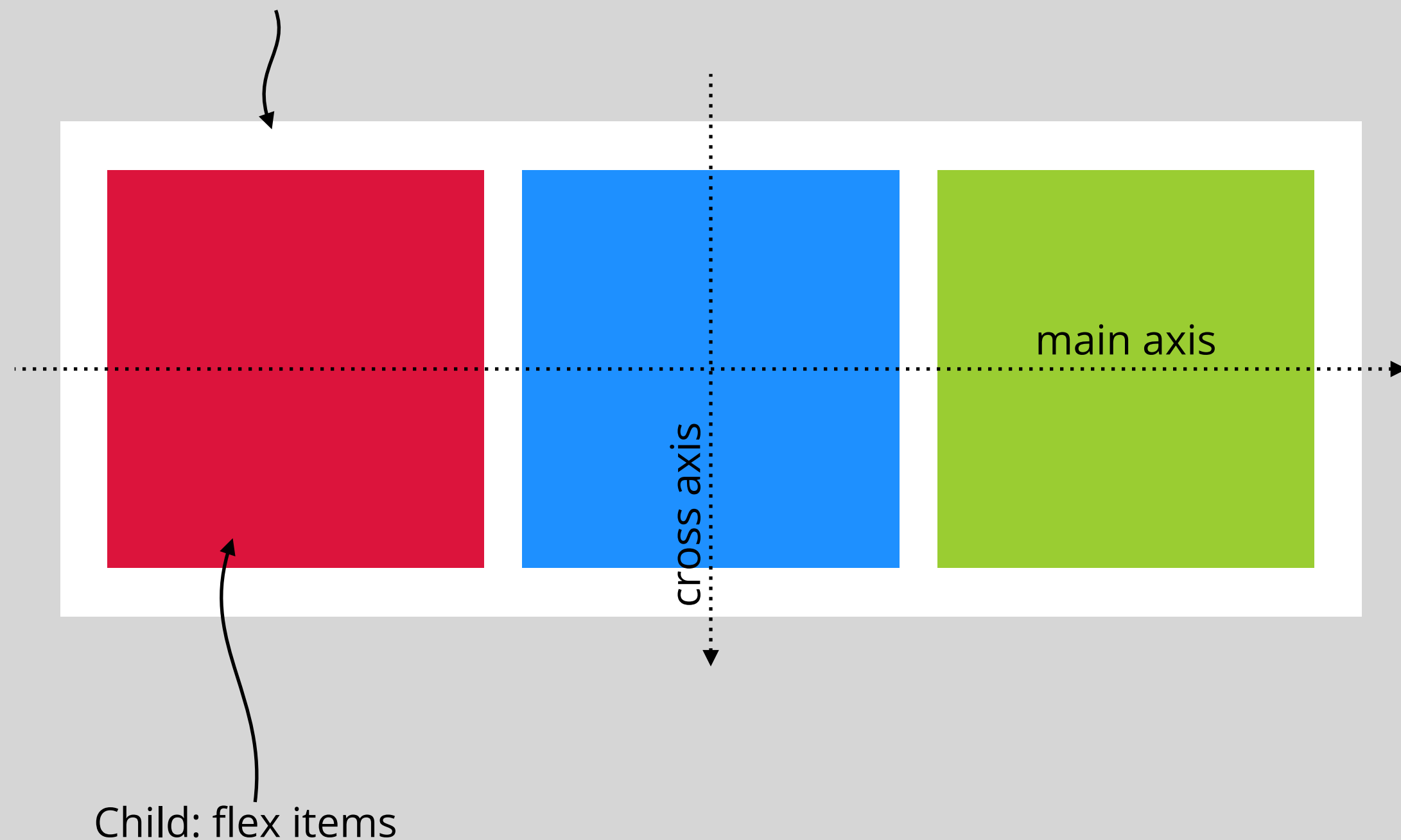
En zo kan er nog veel meer. Daar gaan we deze week mee oefenen.



Eerste item Item 2 Laatste item

# Intro flexbox 6/6

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.

# Dé bron der bronnen

Het naslagwerk met alles over Flexbox:

[css-tricks.com/snippets/css/a-guide-to-flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox)

A screenshot of the CSS-Tricks website showing the article 'A Complete Guide to Flexbox' by Chris Coyier. The page has a dark header with navigation links: \*CSS-TRICKS, ARTICLES, VIDEOS, ALMANAC, NEWSLETTER, GUIDES, BOOKS. The article title is 'A Complete Guide to Flexbox' with a sub-header 'HOME / GUIDES /'. The author is Chris Coyier, dated Apr 8, 2013 (Updated on Aug 16, 2021). The article text states: '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.'

Below the text, there are two diagrams illustrating flexbox concepts. The left diagram shows a 'container' (purple box) containing three 'items' (orange boxes). The right diagram shows a 'container' (purple box) containing three 'items' (orange boxes). The left diagram is labeled 'Parent: flex container' and the right diagram is labeled 'Child: flex items'.

Below the diagrams, there are two sections: 'Properties for the Parent (flex container)' and 'Properties for the Children (flex items)'.

ԼԵՏԳԵ  
ԷԼԵXPOX  
ԷԷԸ ՏՎ\ՏԸ

