



# Les 10

Lay-out:  
Grid – deel 2  
Flex

# Content

- Grid: Uitlijnen
- Flex

# Grid - Uitlijnen

# Mogelijke uitlijningen

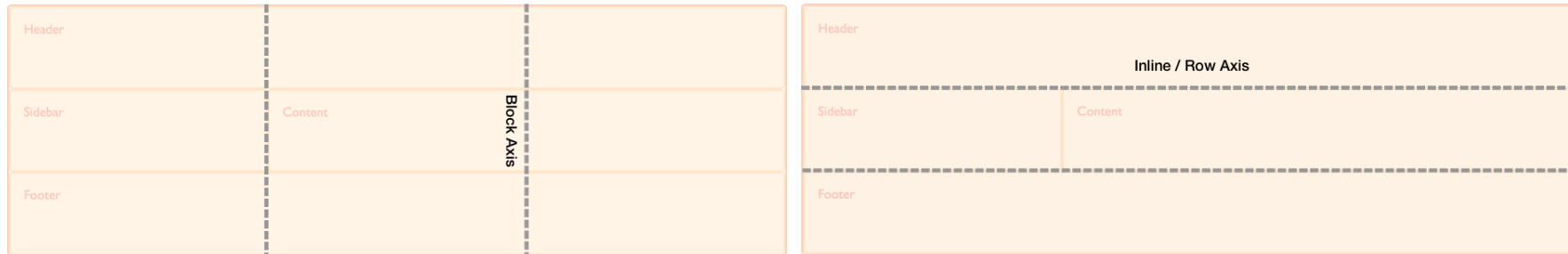
Er zijn twee mogelijke uitlijningen:

- Uitlijnen van grid in zijn container :
  - [justify-content](#): start/center/space-between/space-around/space-evenly
  - [align-content](#): start/center/space-between/space-around
- Uitlijnen van item in een grid-cell/track/area:
  - Alle items in grid (flexbox): [justify-items](#) en [align-items](#) (declaratie in container element.
  - Eén item: [justify-self](#) en [align-self](#) (declaratie in grid-item).

# Uitlijnen van item in een grid-cell/track/area

Om items uit te lijnen langs de **block axis** gebruik je de properties [align-items](#) en [align-self](#).

Items uitlijnen langs de **inline axis** doe je met [justify-items](#) and [justify-self](#).



# Uitlijnen: een voorbeeld

We werken met de voorbeelden van de MDN-website. Deze voorbeelden gebruiken allemaal de volgende basis HTML en CSS ...

## HTML

```
<div class="wrapper">
  <div class="item1">Item 1</div>
  <div class="item2">Item 2</div>
  <div class="item3">Item 3</div>
  <div class="item4">Item 4</div>
</div>
```

## CSS

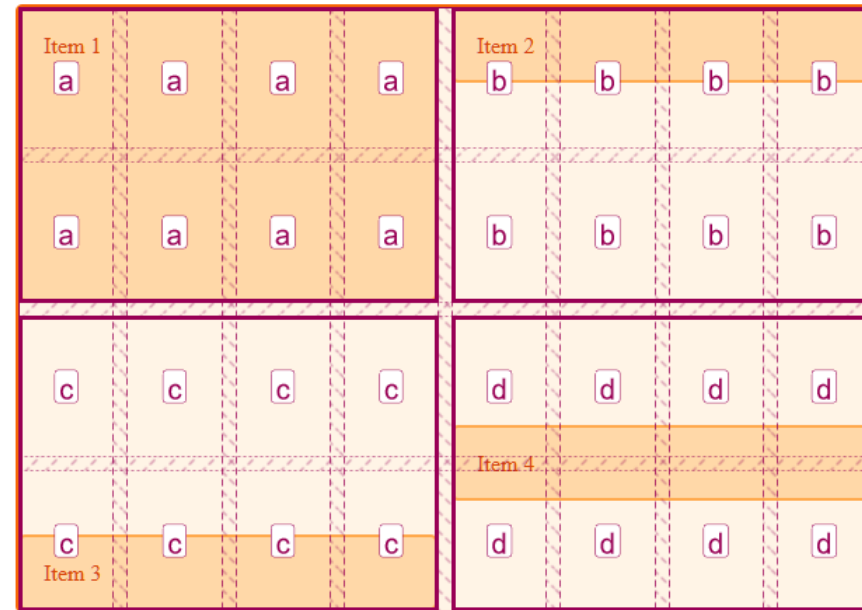
```
.wrapper {
  display: grid;
  grid-template-columns: repeat(8, 1fr);
  grid-gap: 10px;
  grid-auto-rows: 100px;
  grid-template-areas:
    "a a a a b b b b"
    "a a a a b b b b"
    "c c c c d d d d"
    "c c c c d d d d";
}

.item1 {grid-area: a;}
.item2 {grid-area: b;}
.item3 {grid-area: c;}
.item4 {grid-area: d;}
```

# Block axis: align-self en align-items

items uitlijnen langs de block axis met `align-self` en `align-items`

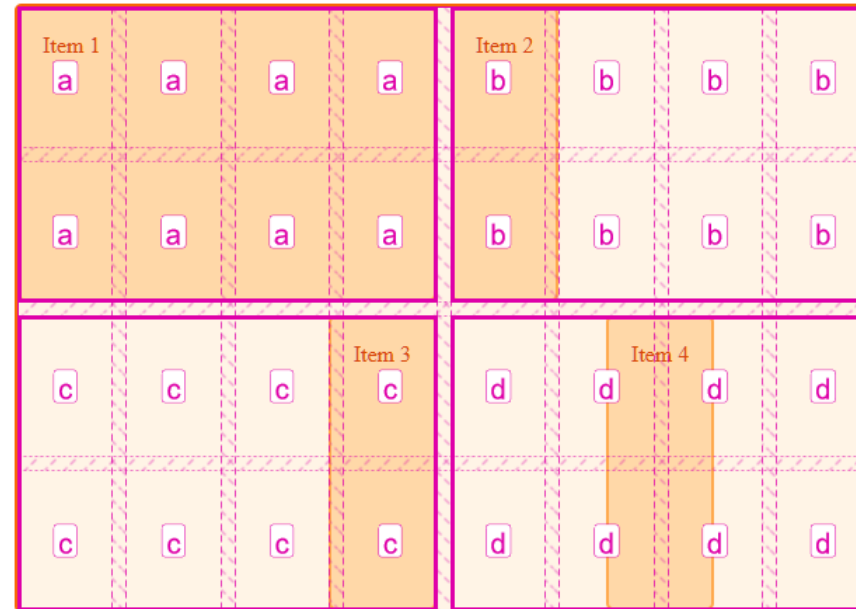
```
.item1 {align-self: stretch;}  
.item2 {align-self: start;}  
.item3 {align-self: end;}  
.item4 {align-self: center;}
```



# Inline axis: justify-self en justify-items

items uitlijnen langs de inline axis met `justify-self` en `justify-items`

```
.item1 {justify-self: stretch;}  
.item2 {justify-self: start;}  
.item3 {justify-self: end;}  
.item4 {justify-self: center;}
```

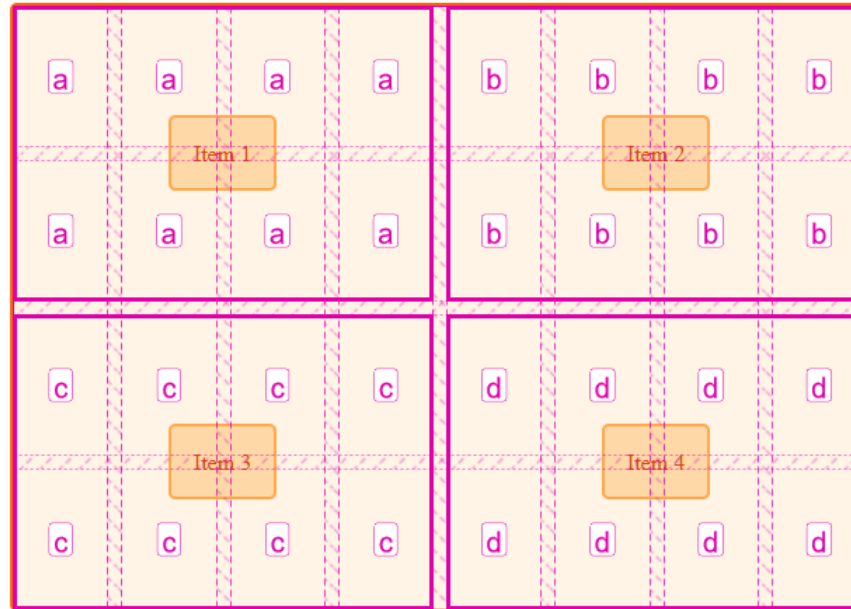




# Inline axis: justify-self en justify-items

items centreren met `align-self/align-items` en `justify-self/justify-items`

```
.wrapper {  
  align-items: center;  
  justify-items: center;  
}
```



**Flex**

**HO  
GENT**

# Flex container

Flex verdeelt de inhoud in 1 richting (één dimensioneel).

Creëer een flex container-element.

Hiermee definieer je impliciet ook de flex items.

Je creëert een *flex container*-element met

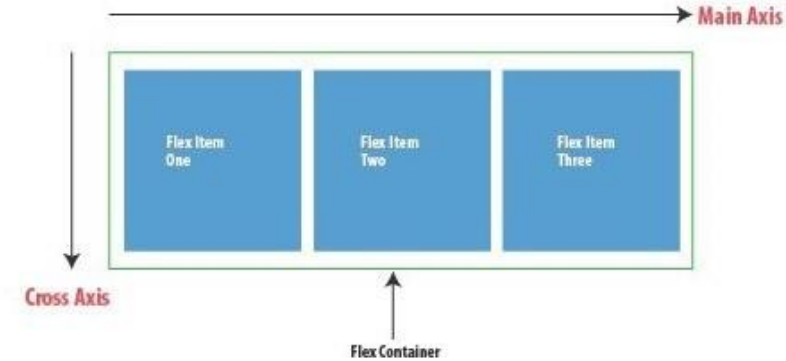
- `display: flex`
- `display: inline-flex`

Het creëren van een *flex container* heeft tevens tot gevolg dat alle (directe) kind-elementen van het *flex container*-element, *flex items* worden.

# Flex containers – richting items

- [Flex containers](#) hebben een **main axis** en een **cross axis**

- Standaard gaat
  - main axis van links naar rechts
  - cross axis van boven naar onder
- kan aangepast worden met [flex-direction](#) property



- De container wordt opgevuld langs de main axis
- Overloop van inhoud wordt geregeld met: [flex-wrap](#)
- Shorthand (direction wrap): [flex-flow](#)

# Flex containers - uitlijnen

## Uitlijnconcepten.

- Uitlijning binnen container:
  - langs de main-axis:  
justify-content: start/center/space-between/space-around/space-evenly
  - Langs de cross-axis:  
align-content: start/center/space-between/space-around/space-evenly

# Flex containers - uitlijnen

- Uitlijning items binnen container:
  - Uitlijnen van alle items langs de cross-axis:  
[align-items](#): stretch/start/center/end/base-line
  - Uitlijnen van individueel flex-item langs cross-axis:  
[align-self](#): stretch/start/center/end/base-line

# Flex – volgorde items - grootte

Volgorde kan gewijzigd worden met eigenschap [order](#).

Initiële waarde van flex-item: [flex-basis](#)

Flex-items kunnen groeien of krimpen als er extra plaats is in de container: [flex-grow](#) - [flex-shrink](#)

Waarde: getal

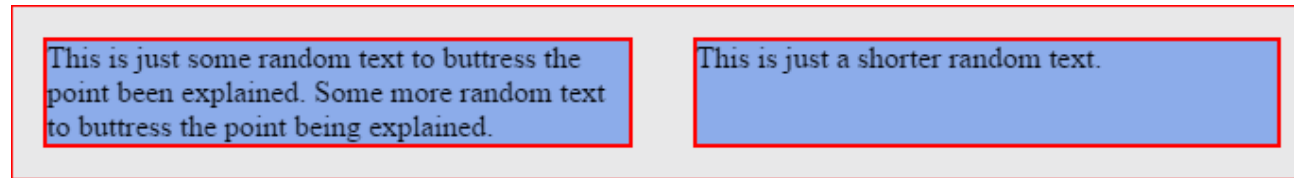
- 0: niet groeien/krimpen
- positief: groei/krimp in verhouding met andere items

Short: [flex](#)

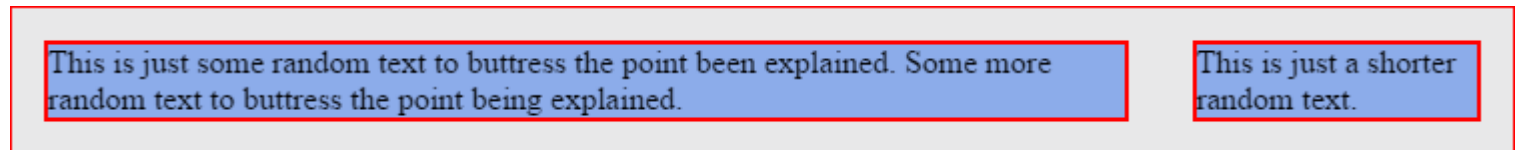
flex-grow: 2;	}	flex: 2 1 auto;
flex-shrink: 1;		
flex-basis: auto;		

# Absolute en relatieve flex

- Absolute flex items: ingenomen ruimte enkel bepaald door Flexbox
  - flex: 1 1 0;
  - item mag groeien, mag verkleinen, en er wordt geen ruimte bepaald op voorhand



- Relative flex items: ingenomen ruimte enkel bepaald door grootte inhoud
  - flex: 1 1 auto;
  - item mag groeien, mag verkleinen, maar ruimte wordt eerst automatisch bepaald door inhoud





# Flexbox en margin: auto;

margin: auto instellen op een item zal vrije ruimte “verplaatsen”

```
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
ul {
  display: flex;
}
li {
  flex: 0 0 auto;
}
```



Branding Home Services About Contact

# Flexbox en margin: auto;

Rechtermarge instellen zorgt dat daar de vrije ruimte geplaatst wordt

```
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
ul {
  display: flex;
}
li {
  flex: 0 0 auto;
}
li:nth-child(1) {
  margin-right: auto;
}
```

Branding

Home

Services

About

Contact

B

About

Contact

Portfolio

Visit

# Flexbox en margin: auto;

Beide margins instellen zorgt dat de vrije ruimte langs beide kanten verspreid wordt

```
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
ul {
  display: flex;
}
li {
  flex: 0 0 auto;
}
li:nth-child(1) {
  margin-left: auto;
  margin-right: auto;
}
```

