



Web Development I

Les 7:
CSS Lay-out – Flex

**HO
GENT**

Inhoud

- ▶ Basisconcepten – CSS Layout
- ▶ Flexbox
 - Flex container
 - Flex items
 - Absolute & relative flex
 - Flexbox en margin: auto;
 - Flex gap

A large, stylized orange cross graphic is positioned on the left side of the slide. The cross is composed of four rectangular arms of equal width and height, meeting at a central white square. The text is overlaid on the horizontal and vertical bars of the cross.

Basisconcepten – CSS

Layout

Lay-out

- ▶ De **normale flow** (**position: static**) van een pagina stapelt alle block elementen op elkaar. Elk block element begint op een nieuwe lijn. Zelfs als de breedte van een element wordt aangepast (verminderd) zal een onderliggend element niet deze ruimte innemen, tenzij men ingrijpt in de normale flow.



Lay-out

- ▶ De **normal flow** (**position: static**) is duidelijk niet de meest sexy layout.
- ▶ Om de normal flow te doorbreken heeft men de volgende mogelijkheden:
 - flexbox lay-out: flex
 - grid lay-out: volgende les
 - float lay-out: float : volgende les
 - relatieve positionering : volgende les
 - absolute positionering : volgende les
 - fixed positionering : volgende les

CSS Flex Layout

- ▶ Flexbox staat voor Flexible Box. Met deze lay-outmodule voor CSS kan je adaptieve ontwerpen (responsive design) maken. Al naargelang de grootte van het scherm en de richting (horizontaal, vertikaal) ervan, worden website-elementen automatisch:
 1. gerangschikt;
 2. een hoogte en een breedte toegekend;

Flexible Box Layout vs Grid Layout

- ▶ In tegenstelling tot 'Flexible Box layout' (kort Flexbox of flex), dat een eendimensioneel lay-outstelsel is, is 'Grid Layout' (kort Grid) een tweedimensioneel lay-outstelsel. Zie afbeelding volgende dia.
- ▶ Grid en flex kunnen samen gebruikt worden om complexe lay-outs te maken.
- ▶ Grid wordt besproken in de volgende les.

Flexible Box Layout vs Grid Layout

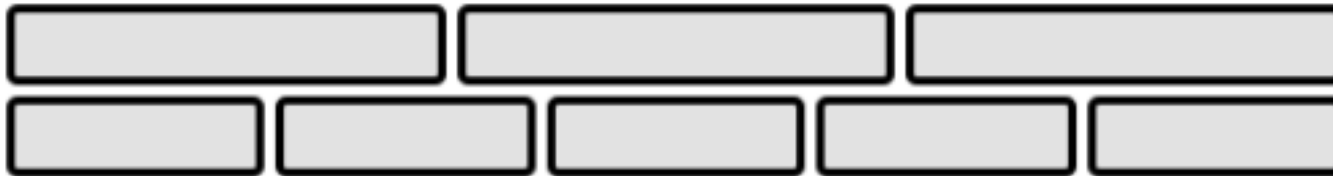


Figure 1 Representative Flex Layout Example



Figure 2 Representative Grid Layout Example

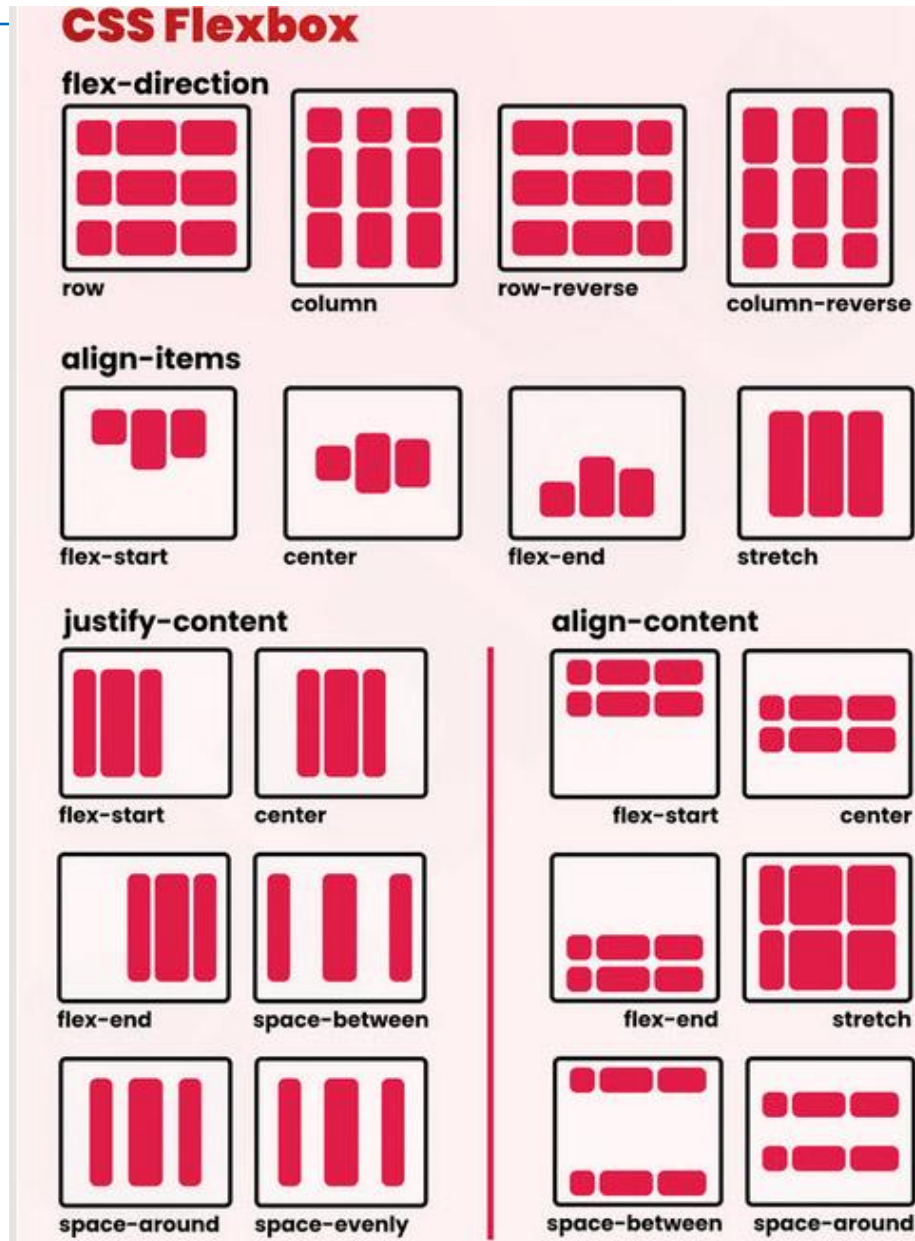
Bron: <https://www.w3.org/TR/css-grid-1/#intro>, laatst geraadpleegd op 2021-10-25

Flexbox

Inleiding

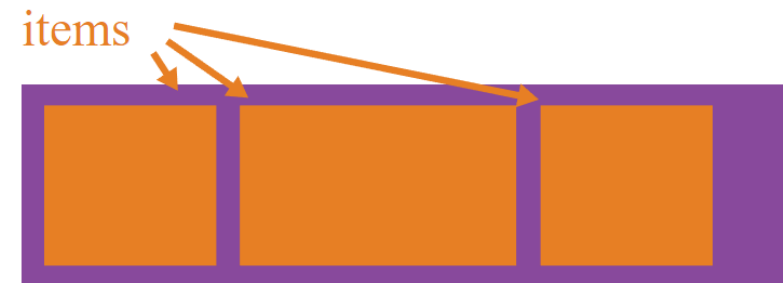
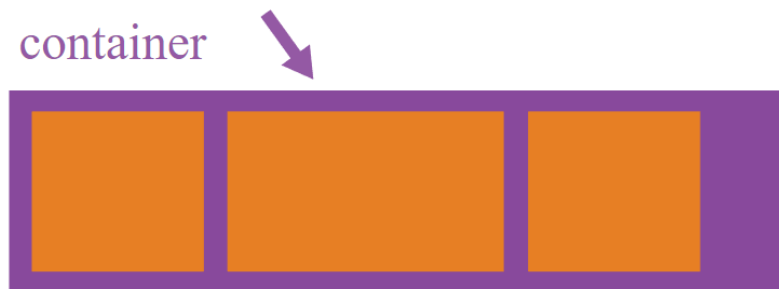
- ▶ [Flexbox](#) is een één dimensionele manier in CSS om **delen van je webpagina** eenvoudig te lay-outen in rijen en/of kolommen
- ▶ Lost moeilijkheden zoals verticaal centreren in “gewone CSS” op.
- ▶ Basisidee: elementen positioneren langs **assen**.
 - Er is een **main axis** en een **cross axis**.
 - We spreken niet meer van **links en rechts** of van **horizontaal en verticaal**.
 - De main axis loopt default horizontaal van links naar rechts en de cross axis verticaal van boven naar onder.

Visuele voorstelling flexbox eigenschappen



Flex container

- ▶ Er is steeds een omvattende container.
- ▶ De rechtstreekse children van deze omvattende container zullen op *flexibele* wijze getoond zullen worden: **flex items**
- ▶ CSS property om van de omvattende container een flexbox te maken is: **display: block flex;** of **display: inline flex;**
- ▶ De eerste waarde bepaalt het outer display type, de omvattende container is dus een block of een inline element
- ▶ De tweede waarde zorgt ervoor dat de rechtstreekse children flex-items worden. Deze volgen niet meer de standaard lay-out.
- ▶ Vroeger was het voldoende 1 waarde mee te geven, dit zal je zeker nog tegenkomen in bestaande sites
 - **display: flex;** = **display: block flex;**
 - **display: inline-flex;** = **display: inline flex;**

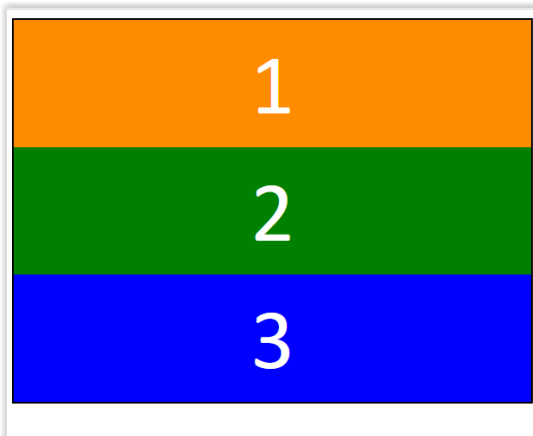


Flex container

► 01-Flexbox/container

```
/* de flexcontainer */  
h1+div {  
  border: 2px solid ■ black;  
  display: block_flex;  
}
```

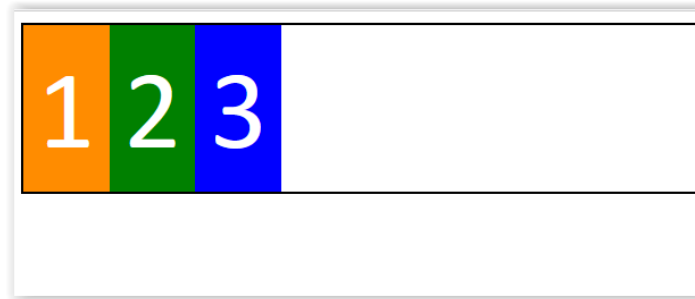
Normal lay-out



Of (oude manier):

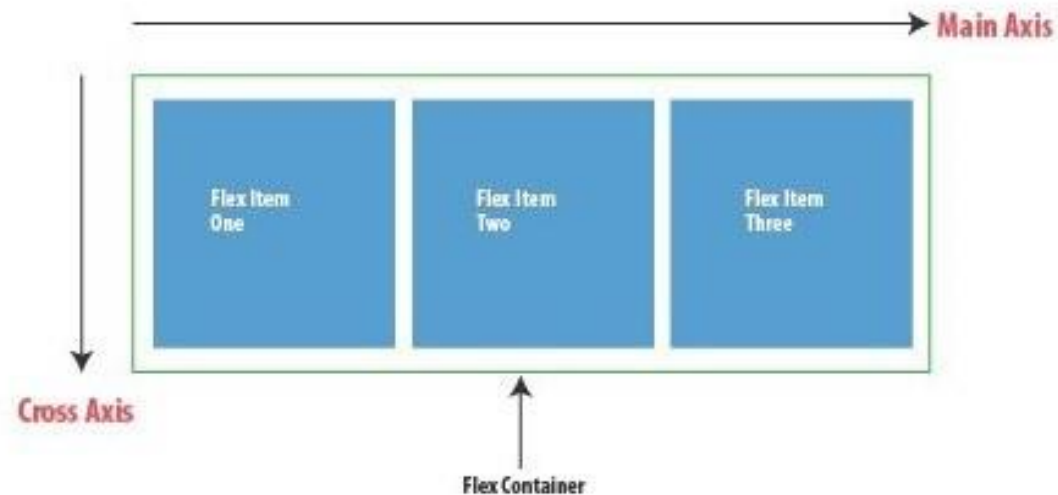
```
/* de flexcontainer */  
h1+div {  
  border: 2px solid ■ black;  
  display: flex;  
}
```

Flexbox lay-out: **display: block flex;**



Flex containers

- ▶ Flex containers hebben een **main axis** en een **cross axis**
 - Standaard gaat de main axis van links naar rechts en de cross axis van boven naar onder
 - Wordt aangepast met *flex-direction* property
- ▶ De container wordt opgevuld langs de main axis.



Flex containers: flex-direction

- ▶ De richting van de **main axis** kan gewijzigd worden door de [flex-direction](#):

- flex-direction: row; (default value)



- flex-direction: column;



- flex-direction: row-reverse;



- flex-direction: column-reverse;



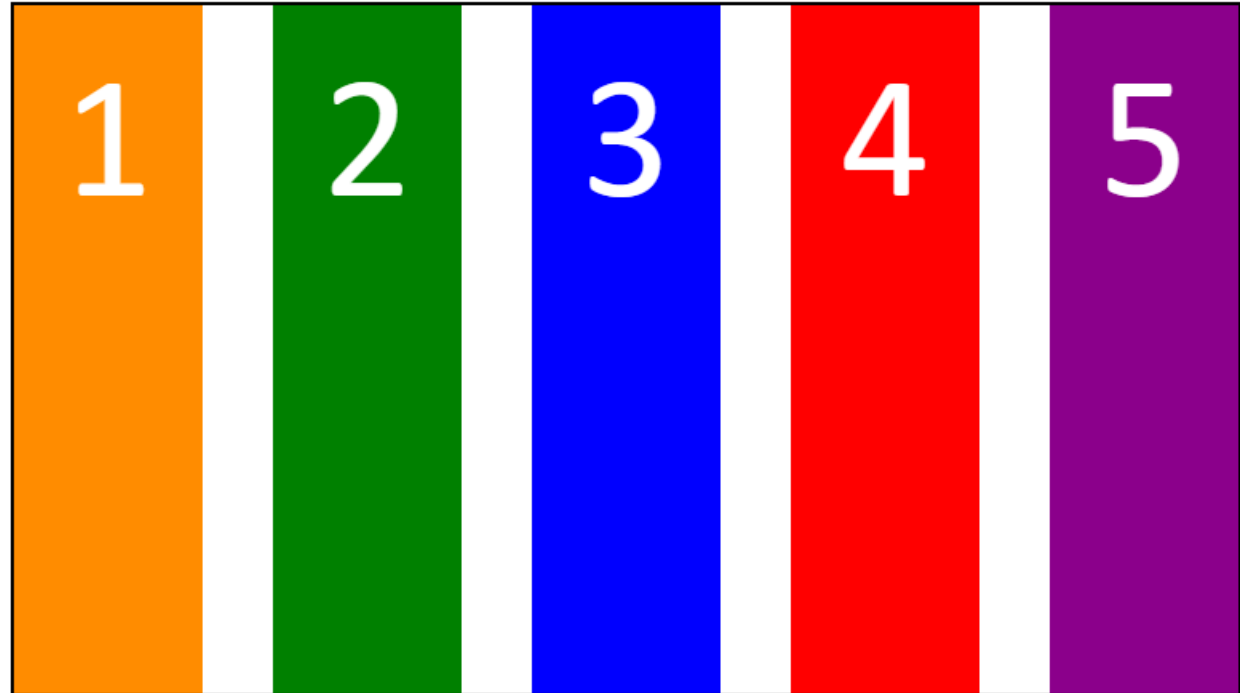
Flex gap

- ▶ Witruimte tussen kolommen en rijen, genoemd ‘Gutters’, creëer je met `column-gap` en `row-gap` of de shorthand `gap`.
- ▶ Als `gap` 2 waarden heeft, is de eerste voor `row-gap`, de tweede voor `column-gap`
- ▶ Mogelijke waarden:
 - length (px, em, rem, ...)
 - percentages

Flex gap

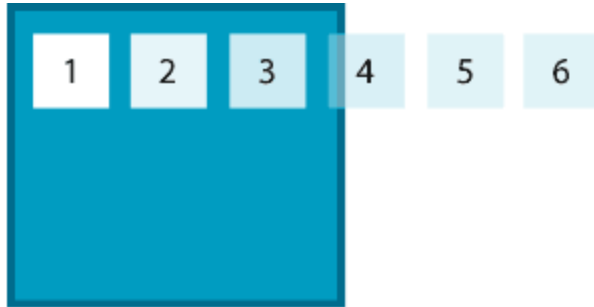
Flexbox - container

```
h1 + div {  
  border: 2px solid ■ black;  
  display: block flex;  
  width: 35rem;  
  height: 50vh;  
  gap: 2rem;  
}
```



Flex containers: flex-wrap

- ▶ Indien de container de flex-items niet meer langs de main-axis kan plaatsen (te weinig ruimte) vallen de flex-items standaard buiten de flex container.



flex-wrap: nowrap;

- ▶ Kan gewijzigd worden met [flex-wrap](#)
 - flex-wrap: nowrap; (default value)
 - flex-wrap: wrap;
 - flex-wrap: wrap-reverse;



flex-wrap: wrap;

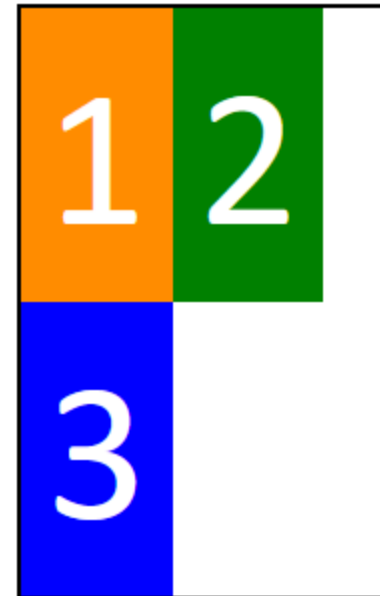
Flex containers: flex-wrap

► 01-Flexbox/container

```
/* de flexcontainer */  
h1+div {  
  border: 2px solid ■black;  
  display: block flex;  
  width: 150px;  
  flex-wrap: no-wrap; /* default value */  
}
```

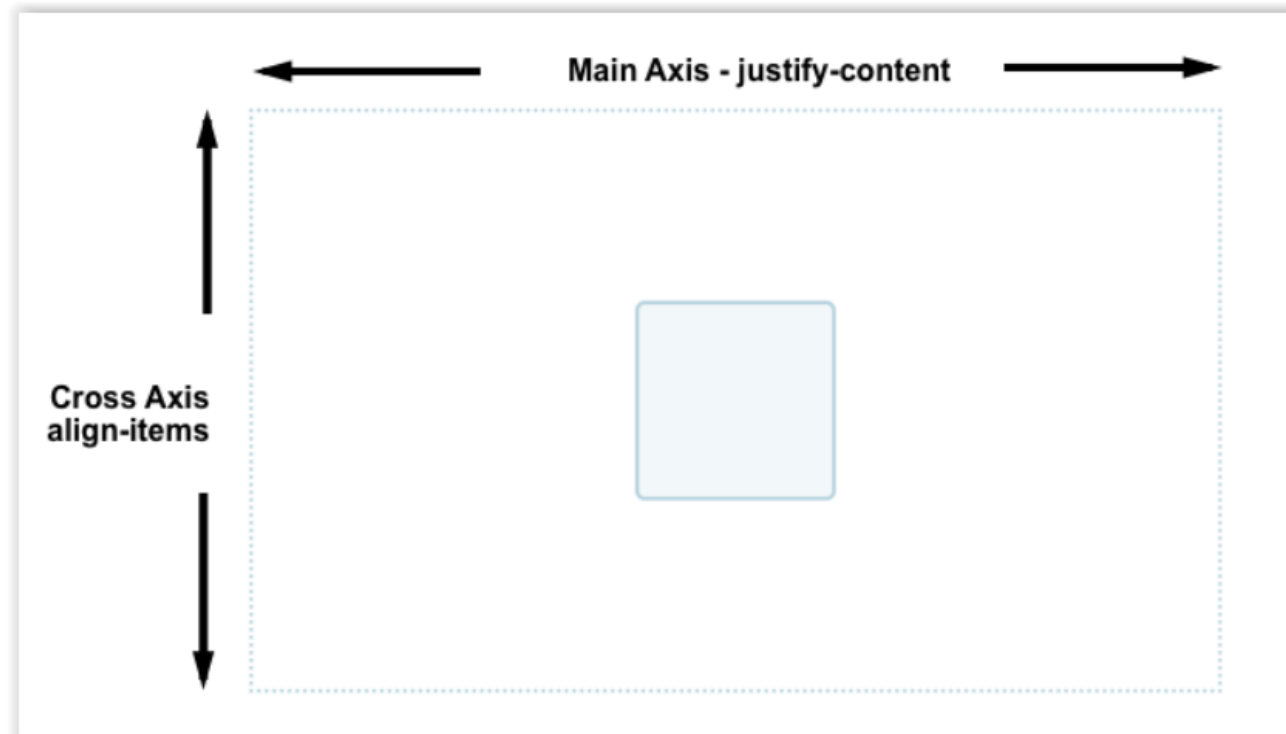


```
/* de flexcontainer */  
h1+div {  
  border: 2px solid ■black;  
  display: block flex;  
  width: 150px;  
  flex-wrap: wrap;  
}
```



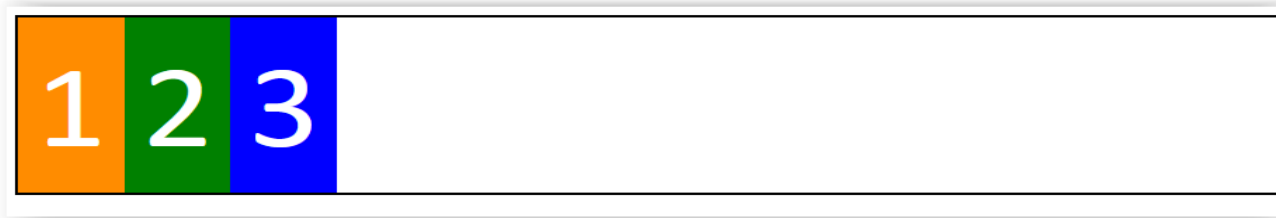
Flex container.

- ▶ Items uitlijnen:
 - langs de main-axis: **justify-content**
 - langs de cross-axis: **align-items**



Flex containers: justify-content

- ▶ Items positioneren langs **main axis** met justify-content
 - justify-content: flex-start; (default)



- justify-content: flex-end;

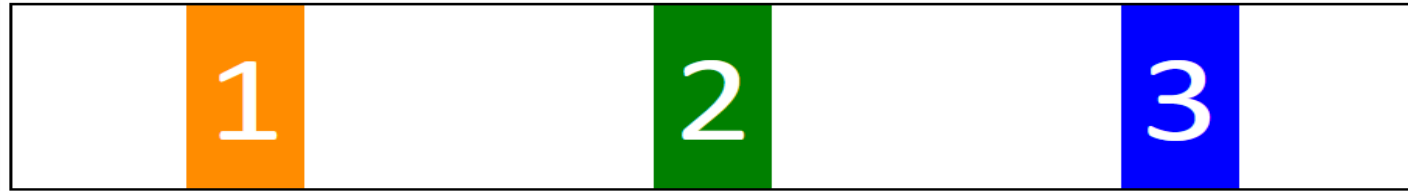


- justify-content: center;



Flex containers: justify-content

- justify-content: space-around; **Rond** elk item evenveel witruimte



- justify-content: space-between; **Tussen** elk item evenveel witruimte



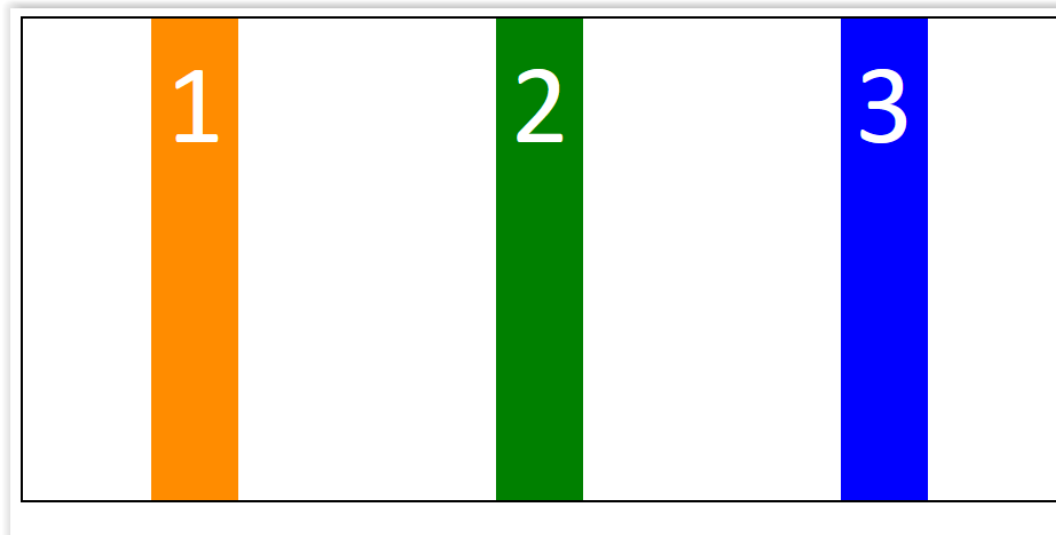
- justify-content: space-evenly; **Voor, na en tussen** elk item evenveel witruimte



Flex containers: align-items

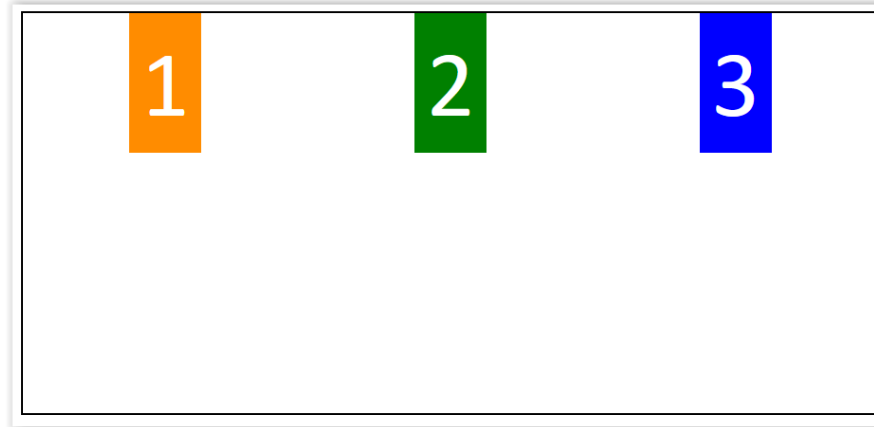
- ▶ **Items** positioneren langs **cross axis** met **align-items**
 - align-items: stretch; (default)

```
/* de flexcontainer */  
h1+div {  
  border: 2px solid ■black;  
  display: block flex;  
  /* width: 150px;  
  flex-wrap: wrap; */  
  justify-content: space-around;  
  height: 50vh;  
  align-items: stretch; /* default value */  
}
```

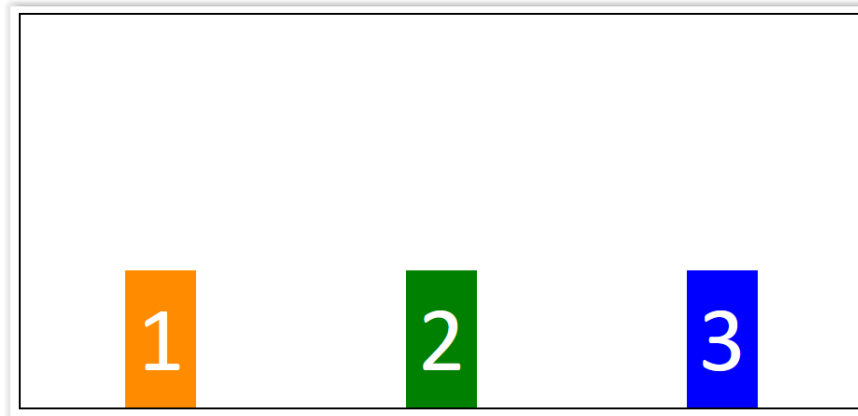


Flex containers: align-items

- ▶ **Items** positioneren langs **cross axis** met **align-items**
 - align-items: flex-start;

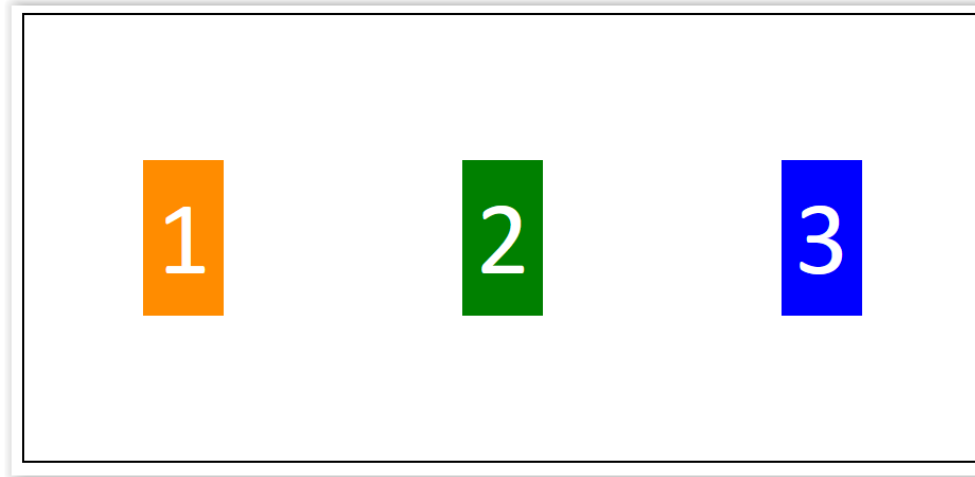


- align-items: flex-end;

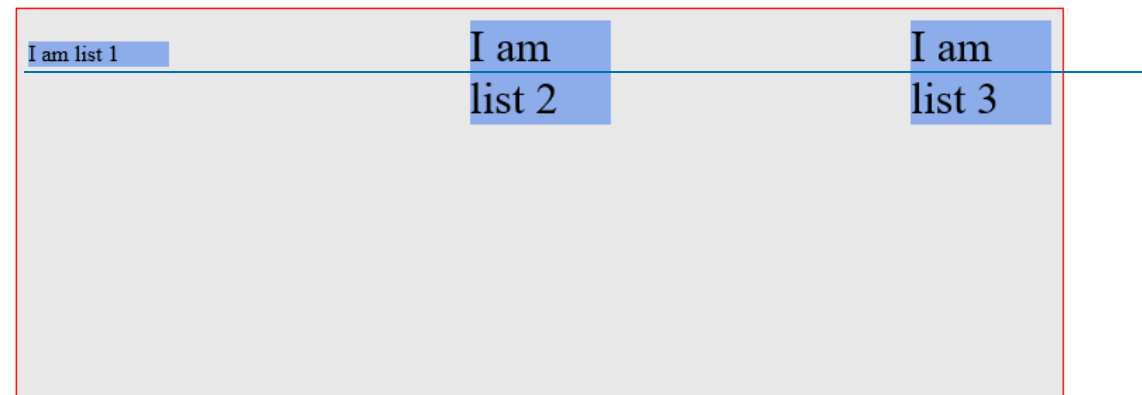


Flex containers: align-items

- align-items: center;



- align-items: baseline;
 - aligneert items volgens “onderkant” tekst.



Flex containers: align-content

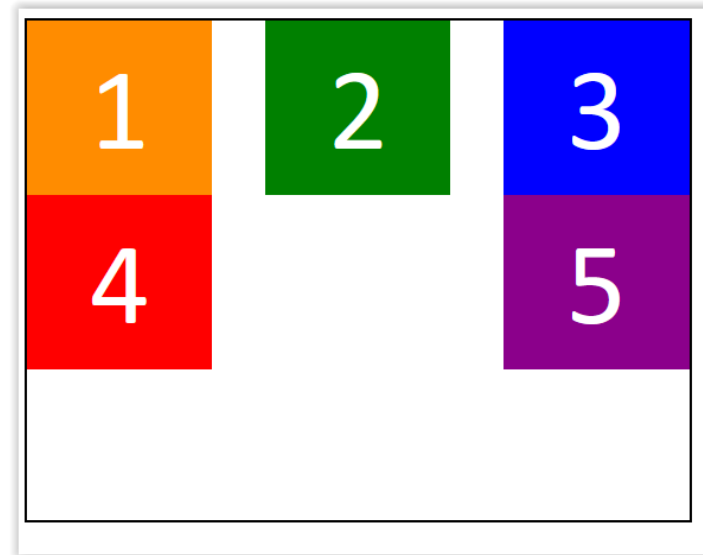
- ▶ Distributie van de ruimte langs de cross-axis met **align-content**. Deze eigenschap heeft geen effect op single-line containers.

```
/* Some default styles to make each box visible */
div > div {
  color: □white;
  font-size: 5em;
  text-align: center;
  padding: 10px;
  width: 10rem;
}

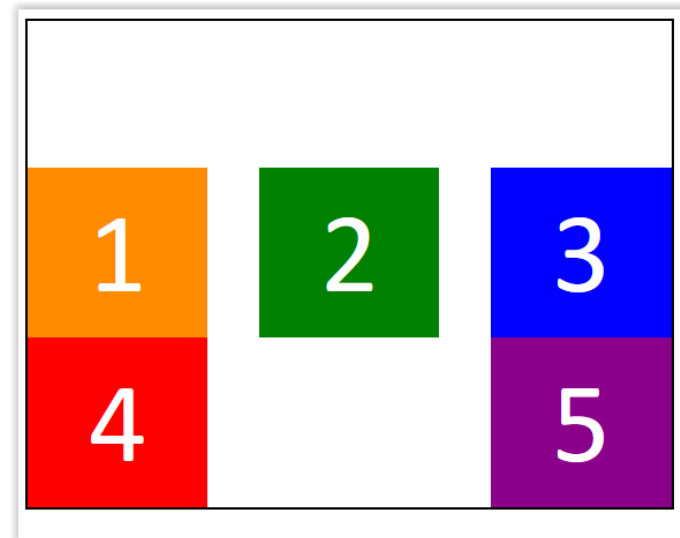
/* de flexcontainer */
h1+div {
  border: 2px solid ■black;
  display: block flex;
  width: 35rem;
  flex-wrap: wrap;
  justify-content: space-between;
  height: 50vh;
  align-content: flex-start; /* default value */
}
```

Flex containers: align-content

- align-content: flex-start;

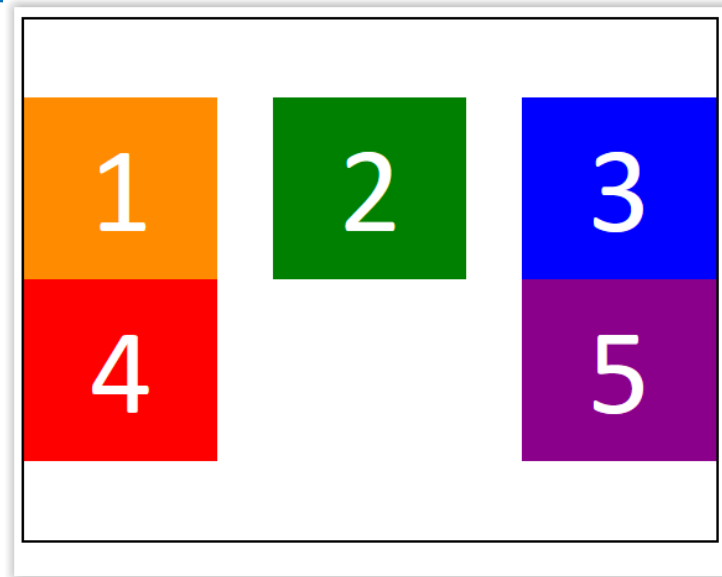


- align-content: flex-end;

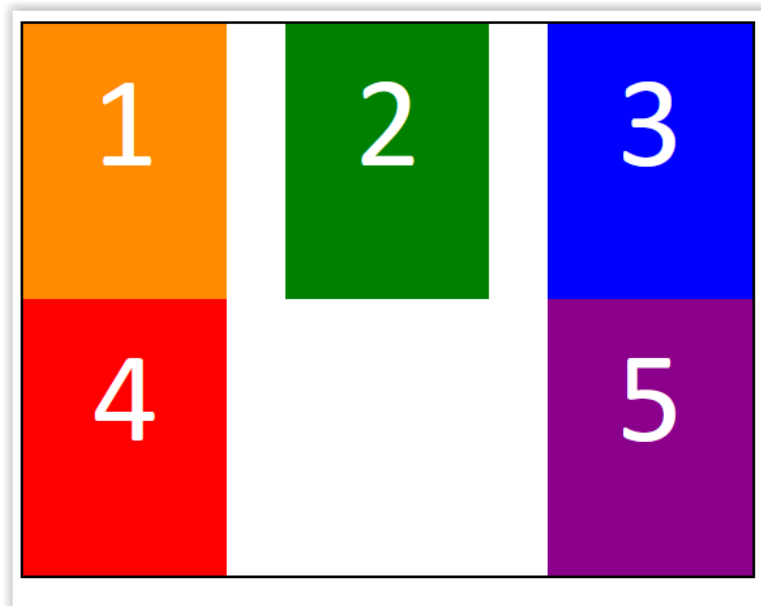


Flex containers: align-content

- align-content: center;



- align-content: stretch;



Flex items: order

- ▶ 01-Flexbox/Items
- ▶ Volgorde waarin de items getoond worden wijzigen, zonder HTML aan te passen
 - Standaardwaarde is 0
 - Items worden geordend van klein naar groot.

```
div > div:nth-child(1) {  
  background-color: darkorange;  
  order: 1;  
}
```



```
/* Some default styles to make each box visible */  
div > div {  
  color: white;  
  font-size: 5em;  
  text-align: center;  
  padding: 10px;  
  width: 10rem;  
}  
  
/* de fl display: block flex;  
h1+div {  
  border: 2px solid black;  
  display: flex;  
}
```

Flex items: flex-grow & flex-shrink

- ▶ *flex-grow* en *flex-shrink* bepalen **hoeveel een item mag groeien/verkleinen** als er extra plaats is in de container
- ▶ Waarde: getal
 - 0: niet groeien
 - positief: groei in verhouding met andere items

```
/* Flex items */
div > div:nth-child(1) {
  background-color: darkorange;
  flex-grow: 0; /* default-value is 0 */
  flex-shrink: 0; /* default-value is 1 */
}
div > div:nth-child(2) {
  background-color: green;
  flex-grow: 2;
  flex-shrink: 2;
}
div > div:nth-child(3) {
  background-color: blue;
  flex-grow: 3;
  flex-shrink: 2;
}
```

```
/* Some default styles to make each box visible */
div > div {
  color: white;
  font-size: 5em;
  text-align: center;
  padding: 10px;
  width: 15rem;
}
```

Flex-items: flex-grow/flex-shrink

The screenshot displays a web browser window with the title "FlexBox" and the URL "127.0.0.1:5501/Completed/01-Flexbox/Items/index.html". The browser shows a page titled "Flexbox - items" with three colored boxes: orange (1), green (2), and blue (3). Below the browser, the developer tools are open, showing the HTML structure and the CSS styles for the selected element.

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <h1>Flexbox - items</h1>
    <div> [flex]
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <!--<div>4</div> <div>5</div>-->
    </div>
    <!--Code injected by live-server-->
    <script type="text/javascript"></script>
  </body>
</html>
```

CSS Styles for the selected element (div):

```
element {
  inline
}
div > div:nth-child(1) {
  style.css:25
  background-color: darkorange;
  flex-grow: 0;
  flex-shrink: 0;
}
div > div {
  style.css:10
  color: white;
  font-size: 5em;
  text-align: center;
  padding: 10px;
  width: 15rem;
}
div {
  style.css:2
  box-sizing: border-box;
}
Overgenomen van body
body {
  style.css:5
  font-family: Calibri, 'Trebuchet MS',
  sans-serif;
```

Inspector Details:

- Flex-item van div
- div
- Basisgrootte (width: 15rem): 220px
- Definitieve grootte: 220px
- Raster: CSS-raster is niet in gebruik op deze pagina
- Vakkenmodel: margin: 0

Flex items: flex-basis

- ▶ *flex-basis* bepaalt de initiële grootte van een item voordat *flex-grow* en *flex-shrink* worden toegepast.
- ▶ Als waarde op auto (initiële waarde) staat
 - Width wordt gebruikt bij horizontale tekst mode
 - Height bij verticale tekst mode
- ▶ merk op: de flex-basis heeft voorrang op de ingestelde width voor het item.

```
/* Flex items */
div > div:nth-child(1) {
  background-color: darkorange;
  flex-basis: 10rem;
  flex-grow: 0; /* default-value */
  flex-shrink: 0; /* default-value */
}
div > div:nth-child(2) {
  background-color: green;
  flex-basis: 15rem;
  flex-grow: 2;
  flex-shrink: 2;
}
div > div:nth-child(3) {
  background-color: blue;
  flex-basis: 12rem;
  flex-grow: 3;
  flex-shrink: 2;
}
```


Flex-items: flex-basis

The image shows a browser window displaying a web page titled "Flexbox - items". The page contains three colored boxes with numbers 1, 2, and 3, representing flex items. Below the browser window, a developer tool is open, showing the HTML structure and the CSS styles for the selected element. The HTML shows a flex container with three items: `<div>1</div>`, `<div>2</div>`, and `<div>3</div>`. The CSS styles for the selected element (the first flex item) are:

```
element {
  background-color: darkorange;
  flex-basis: 10rem;
  flex-grow: 0;
  flex-shrink: 0;
}
```

The developer tool also shows the "Flex-item van div" section, which displays the "Basisgrootte (width: 15rem)" as 140px. The "Definitieve grootte" is also 140px. The "Raster" section shows "CSS-raster is niet in gebruik op deze pagina". The "Vakkenmodel" section shows a margin of 0.

Flex items: flex - shorthand

- ▶ De *flex* property laat je toe flex-grow, flex-shrink en flex-basis in een keer te definiëren
 - flex-grow: 2;
flex-shrink: 1;
flex-basis: 50em;

} flex: 2 1 50em;

Flex items: absolute flex

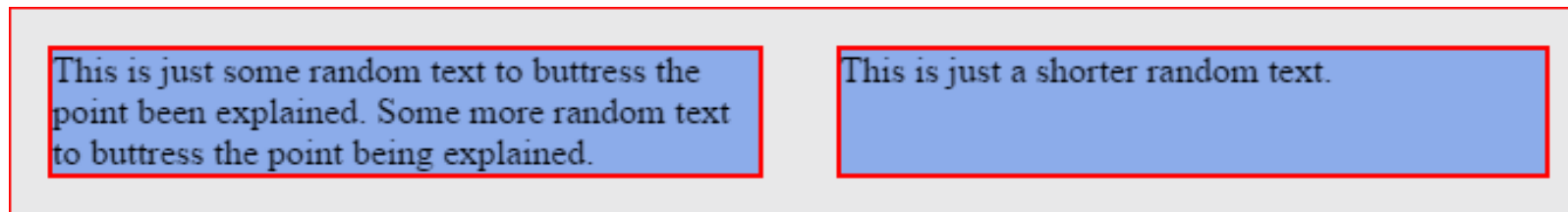
- ▶ absolute flex items

- **flex-basis** krijgt **expliciet een waarde 0**

- voorbeeld:

- **flex: 1 1 0;**

- flex-items kunnen groeien (waarde 1) & kunnen krimpen (waarde 1)
 - startpunt: elk flex-item **neemt geen ruimte in (flex-basis:0)**, dus alle ruimte in de flex-container is vrije ruimte die kan worden verdeeld
 - als resultaat zijn je flex-items even breed



Flex items: relatieve flex

- ▶ relatieve flex items
 - **flex-basis** wordt ingesteld op **content**
 - **flex: 1 1 content;**
 - flex-items kunnen groeien & kunnen krimpen
 - startpunt: ingenomen ruimte wordt eerst bepaald door de **max-content** van de flex-items

This is just some random text to buttress the point been explained. Some more random text to buttress the point being explained.

This is just a shorter random text.

Flex items: relatieve flex

- ▶ relatieve flex items
 - **flex-basis** wordt ingesteld op **auto** (initial value)
 - **flex: 1 1 auto;**
 - flex-items kunnen groeien & kunnen krimpen
 - startpunt: ingenomen ruimte wordt eerst automatisch bepaald door de **oorspronkelijke grootte** van de flex items, i.e.
 - waarde van **width** indien width werd ingesteld op de flex-items
 - **max-content** indien niet expliciet een width werd ingesteld
 - in dit geval is dit hetzelfde als flex-basis **content**

This is just some random text to buttress the point been explained. Some more random text to buttress the point being explained.

This is just a shorter random text.

Flexbox en margin: auto;

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

```
<h1>Flexbox - margin</h1>
<ul>
  <li>Branding</li>
  <li>Home</li>
  <li>Services</li>
  <li>About</li>
  <li>Contact</li>
</ul>
```

```
ul {
  list-style-type: none;
  background-color: lightgray;
  display: block flex;
  padding: 5px;
}
li {
  color: white;
  background-color: deepskyblue;
  border: 1px solid white;
  border-radius: 5px;
  font-size: 1.2rem;
  text-align: center;
  padding: 2px;
  margin: 5px;
  flex: 0 0 auto; /* default value is 0 1 auto */
}
```

Flexbox - margin

Branding Home Services About Contact

Flexbox en margin: auto;

- ▶ Rechtermarge instellen zorgt dat daar de vrije ruimte geplaatst wordt.

```
li:nth-child(1) {  
  margin-inline-end: auto;  
}
```

Flexbox - margin

Branding

Home

Services

About

Contact

Flexbox en margin: auto;

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

```
li:nth-child(1) {  
  margin-inline-end: auto;  
  margin-inline-start: auto;  
}
```

Flexbox - margin

Branding

Home

Services

About

Contact