

Statische websites

CSS LAYOUT, CSS GRID & CSS FLEXBOX

Overzicht

↘ CSS Layout

- × position: static;
- × position: relative;
- × position: fixed;
- × position: absolute;
- × position: sticky;
- × Z-index: value;

↘ CSS Grid

- × Display property
- × Items
- × Columns
- × Rows
- × Gaps
- × Lines
- × grid-template-columns
- × grid-template-columns – the fr unit
- × grid-template-rows

↘ CSS Flexbox

- × Parent element (container)
- × Flex-direction
- × Flex-wrap
- × Justify-content

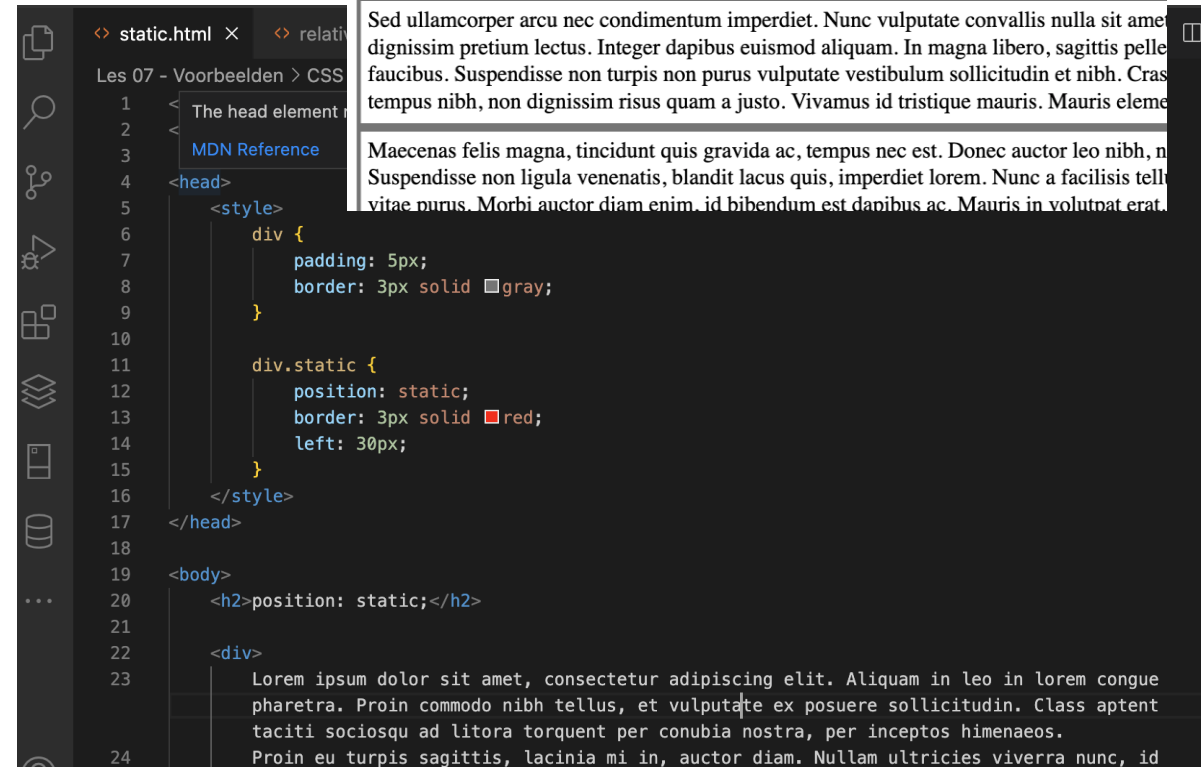
CSS LAYOUT

Position property

- Specificeert **hoe** een element **gepositioneerd** zal worden
- Vijf verschillende positie waarden
 - × Static
 - × Relative
 - × Fixed
 - × Absolute
 - × Sticky

position: static;

- HTML elementen zijn **standaard** static **gepositioneerd**
- Static elementen worden **niet beïnvloed** door de properties **top**, **bottom**, **left** en **right**



The screenshot shows a code editor with a dark theme. The file is named 'static.html'. The code is as follows:

```
1 <!-- The head element -->
2 <!-- MDN Reference -->
3
4 <head>
5   <style>
6     div {
7       padding: 5px;
8       border: 3px solid gray;
9     }
10
11     div.static {
12       position: static;
13       border: 3px solid red;
14       left: 30px;
15     }
16   </style>
17 </head>
18
19 <body>
20   <h2>position: static;</h2>
21
22   <div>
23     Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam in leo in lorem congue p
24     pharetra. Proin commodo nibh tellus, et vulputate ex posuere sollicitudin. Class aptent
25     taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.
26     Proin eu turpis sagittis, lacinia mi in, auctor diam. Nullam ultricies viverra nunc, id
```

On the right side of the editor, there are two panels. The top panel shows a snippet of Lorem Ipsum text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam in leo in lorem congue p litora torquent per conubia nostra, per inceptos himenaeos. Proin eu turpis sagittis, lacinia ultrices pharetra dui. Phasellus sed feugiat ligula. Vivamus sit amet sapien eu sem euismo luctus. Ut fringilla eget augue nec fringilla." The bottom panel shows a snippet of text: "Div element met position: static;".

position: relative;

- Een element met **position: relative;** is **gepositioneerd** ten opzichte van zijn **normale positie**
- Merk op dat je een horizontale scrollbar zal krijgen omdat **left: 20px;** is gebruikt
- De plaats waar dit element zou staan wordt **vrij gehouden**

The image shows a development environment with VS Code and a web browser. In VS Code, the `index.html` file contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5   div.relative {
6     position: relative;
7     left: 30px;
8     border: 3px solid #73AD21;
9   }
10 </style>
11 </head>
12 <body>
13
14 <h2>position: relative;</h2>
15
16 <p>An element with position: relative; is positioned relative to its normal position:</p>
17
18 <div class="relative">
19   This div element has position: relative;
20 </div>
21
22 </body>
23 </html>
```

The browser (Incognito) displays the rendered page. The heading `position: relative;` is shown. Below it, a paragraph states: "An element with position: relative; is positioned relative to its normal position:". Then, a green-bordered box contains the text "This div element has position: relative;". The browser's address bar shows the URL `127.0.0.1:5500/index.html`.

position: absolute;

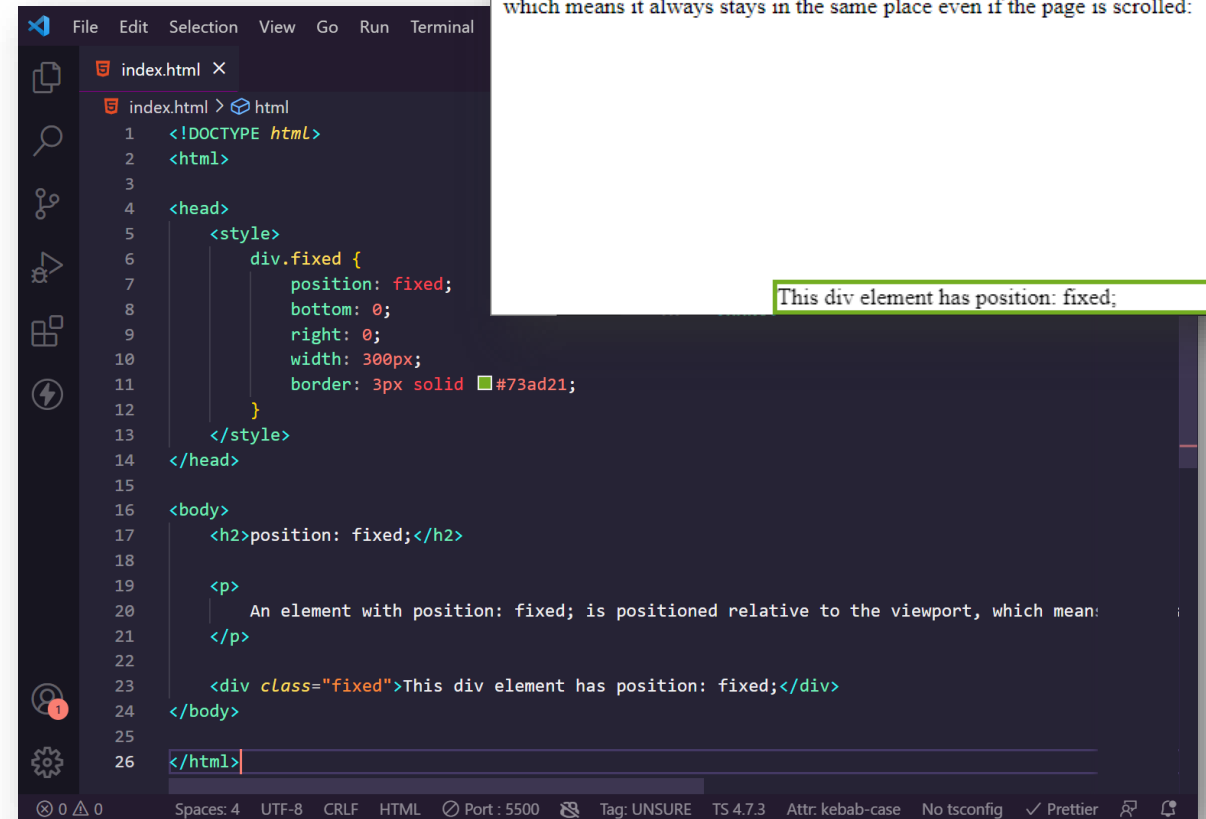
- Een element met **position: absolute;** wordt **gepositioneerd** ten opzichte van zijn **parent**
- Als er geen parent is zal de **<body>** gebruikt worden als parent
- De plaats waar dit element zou staan wordt **niet vrij gehouden**
- Zal wel mee scrollen integenstelling tot **position: fixed;**

The image shows a VS Code editor with an HTML file named 'index.html'. The code defines two CSS classes: 'div.relative' and 'div.absolute'. The 'div.relative' class has a width of 400px, height of 200px, and a green border. The 'div.absolute' class has a width of 200px, height of 100px, and a green border. The HTML body contains a heading, a paragraph, and two nested divs: one with class 'relative' containing the text 'This div element has position: relative;', and another with class 'absolute' containing the text 'This div element has position: absolute;'. To the right, a browser window at 127.0.0.1:5500/index.html displays the rendered page. It shows the heading, the paragraph, and the two nested divs. The 'div.relative' is a large green box, and the 'div.absolute' is a smaller green box positioned inside the relative one. The browser's developer tools are open, showing the CSS for the selected element.

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <style>
6     div.relative {
7       position: relative;
8       width: 400px;
9       height: 200px;
10      border: 3px solid #73ad21;
11    }
12
13    div.absolute {
14      position: absolute;
15      top: 80px;
16      right: 0;
17      width: 200px;
18      height: 100px;
19      border: 3px solid #73ad21;
20    }
21  </style>
22 </head>
23
24 <body>
25   <h2>position: absolute;</h2>
26   <p>
27     An element with position: absolute; is positioned relative to the nearest pos.
28   </p>
29   <div class="relative">
30     This div element has position: relative;
31     <div class="absolute">This div element has position: absolute;</div>
32   </div>
33 </body>
34
35 </html>
```

position: fixed;

- Het element wordt **uit** de normale **document flow** gehaald
- Een element met **position: fixed;** is **gepositioneerd** ten opzichte van de **viewport**, wat betekent dat deze altijd op dezelfde plaats blijft, zelfs als de pagina wordt gescrold
- De properties **top**, **right**, **bottom** en **left** worden gebruikt om het element te positioneren



position: sticky;

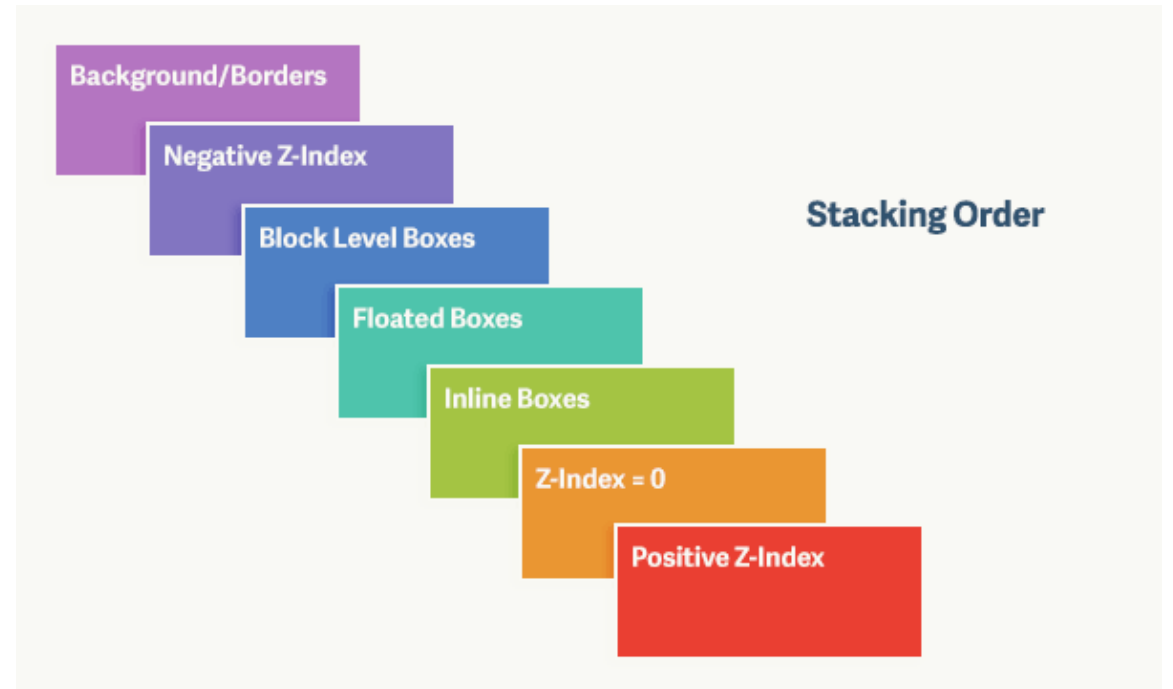
- Een element met **position: sticky;** wordt **gepositioneerd** op basis van de **schuifpositie van de gebruiker**
- Wanneer de sticky div de positie **top: 0;** bereikt zal deze blijven staan

The image shows a web browser window at the top right and a code editor at the bottom. The browser window displays a green box with the text "I am sticky!" at the top of the page. Below the box, there is explanatory text: "In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position. Scroll back up to remove the stickyness." and two paragraphs of Lorem Ipsum text. The code editor shows the HTML and CSS for this example. The CSS defines a class ".sticky" with the following properties: position: -webkit-sticky; position: sticky; top: 0; padding: 5px; background-color: #cae8ca; border: 2px solid #4caf50. The HTML includes a <div class="sticky">I am sticky!</div> and a <div style="padding-bottom: 2000px"> containing the explanatory text and Lorem Ipsum paragraphs.

```
index.html •
index.html > html > body > div.sticky
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <style>
6     div.sticky {
7       position: -webkit-sticky;
8       position: sticky;
9       top: 0;
10      padding: 5px;
11      background-color: #cae8ca;
12      border: 2px solid #4caf50;
13    }
14  </style>
15 </head>
16
17 <body>
18   <p>
19     Try to <b>scroll</b> inside this frame to understand how sticky positioning w
20   </p>
21   <div class="sticky">I am sticky!</div>
22   <div style="padding-bottom: 2000px">
23     <p>
24       In this example, the sticky element sticks to the top of the page (top: 0
25     </p>
26     <p>Scroll back up to remove the stickyness.</p>
27     <p>
28       Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definit
29       efficiantur his ad. Eum no molestiae voluptatibus.
30     </p>
31     <p>
32       Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definit
33       efficiantur his ad. Eum no molestiae voluptatibus.
34     </p>
35   </div>
36 </body>
37
38 </html>
```

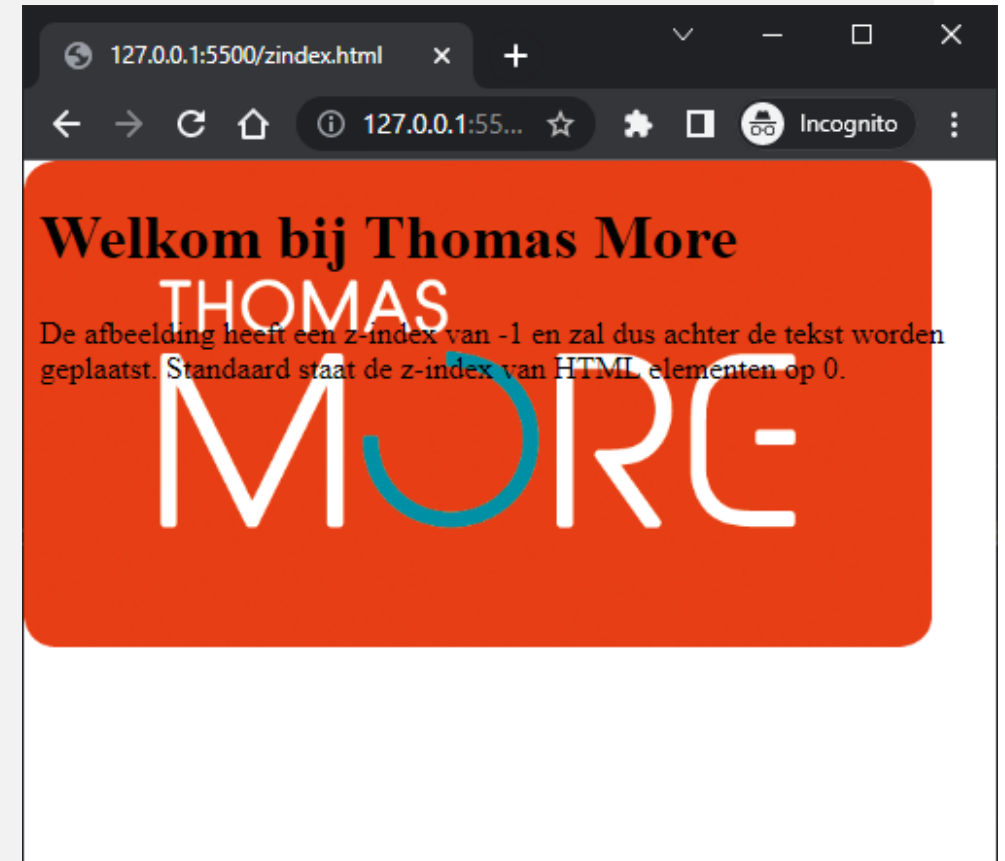
Z-index: value;

- Wanneer elementen worden gepositioneerd, kunnen ze andere **elementen overlappen**
- De property **z-index** geeft de **stapelvolgorde** van een element aan (welk element voor of achter het andere moet worden geplaatst)
- Een element kan een **positieve** of **negatieve** stapelvolgorde hebben



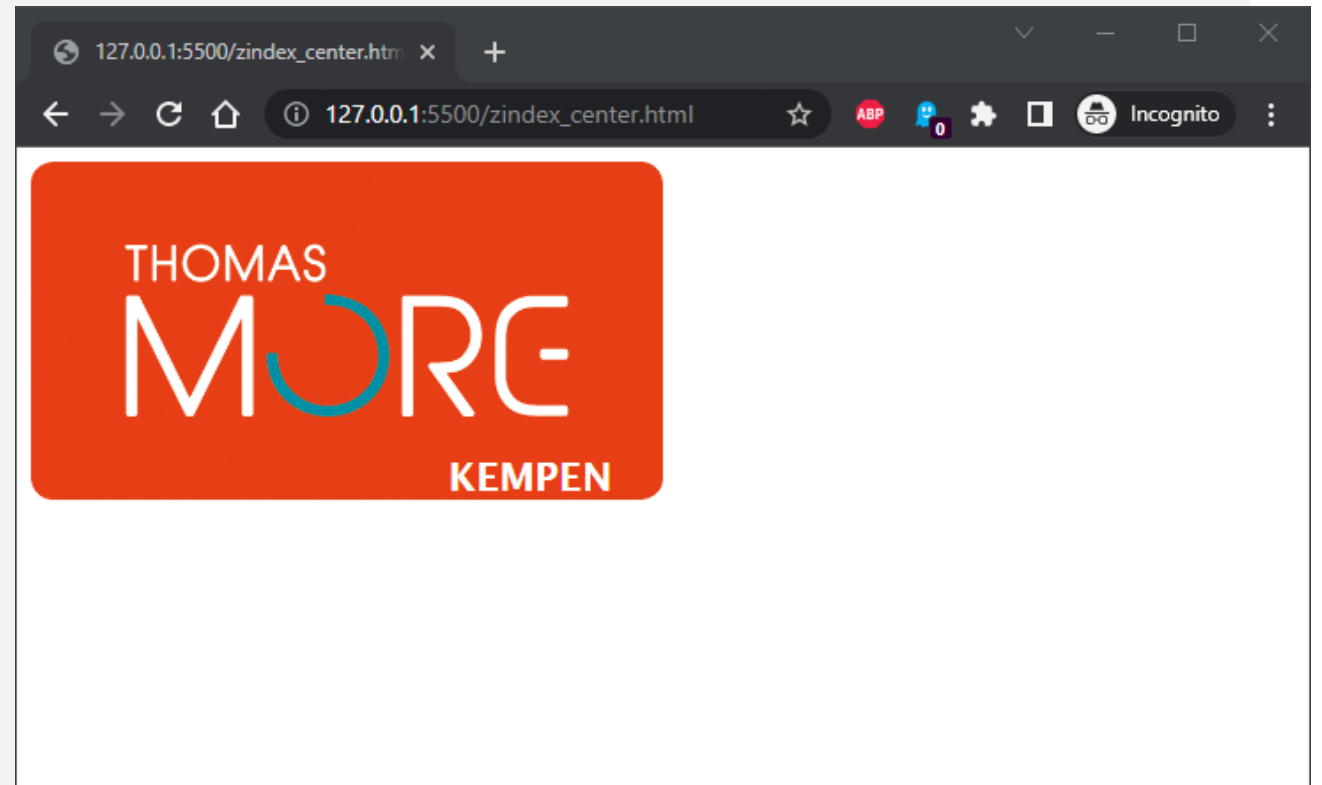
Voorbeeld

```
zindex.html - Position - Visual Studio Code
zindex.html x
zindex.html > html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <style>
6     img {
7       position: absolute;
8       left: 0px;
9       top: 0px;
10      z-index: -1;
11    }
12  </style>
13 </head>
14
15 <body>
16   <h1>Welkom bij Thomas More</h1>
17   
18   <p>
19     De afbeelding heeft een z-index van -1 en zal dus achter de tekst worden geplaatst.
20   </p>
21 </body>
22
23 </html>
```



Voorbeeld

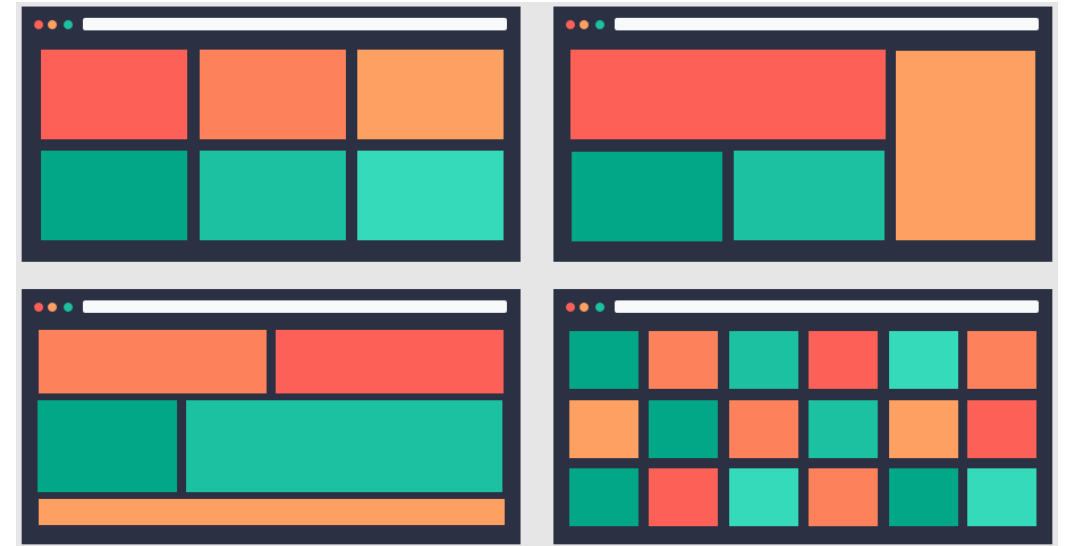
```
zindex_center.html > html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <style>
6     .container {
7       position: relative;
8     }
9
10    .kempen {
11      position: absolute;
12      right: 75px;
13      top: 85px;
14      width: 100%;
15      text-align: center;
16      font-size: 22px;
17      font-weight: 700;
18      color: white;
19      text-transform: uppercase;
20      font-family: "Lucida Sans", Verdana, sans-serif;
21    }
22
23    img {
24      width: 50%;
25      height: auto;
26    }
27  </style>
28 </head>
29
30 <body>
31   <div class="container">
32     
33     <div class="kempen">Kempen</div>
34   </div>
35 </body>
36
37 </html>
```



CSS Grid

CSS Grid

- Lay-out met **rijen** en **kolommen**
- Ondersteund door alle moderne browsers



Display property

- ↘ Een HTML element wordt een **grid container** wanneer we het display property instellen op **grid** of **inline-grid**
- ↘ Alle **child** elementen van de **grid container** worden **grid items**

https://www.w3schools.com/css/tryit.asp?filename=trycss_grid

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  background-color: #2196F3;  
  padding: 10px;  
}
```

```
.grid-container {  
  display: inline-grid;  
  grid-template-columns: auto auto auto;  
  background-color: #2196F3;  
  padding: 10px;  
}
```

Items

- ↘ Een grid bestaat uit **één parent** element met **één of meerdere child** elementen

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  background-color: #2196F3;  
  padding: 10px;  
}  
.grid-item {  
  background-color: rgba(255, 255, 255, 0.8);  
  border: 1px solid rgba(0, 0, 0, 0.8);  
  padding: 20px;  
  font-size: 30px;  
  text-align: center;  
}
```

1	2	3
4	5	6
7	8	9

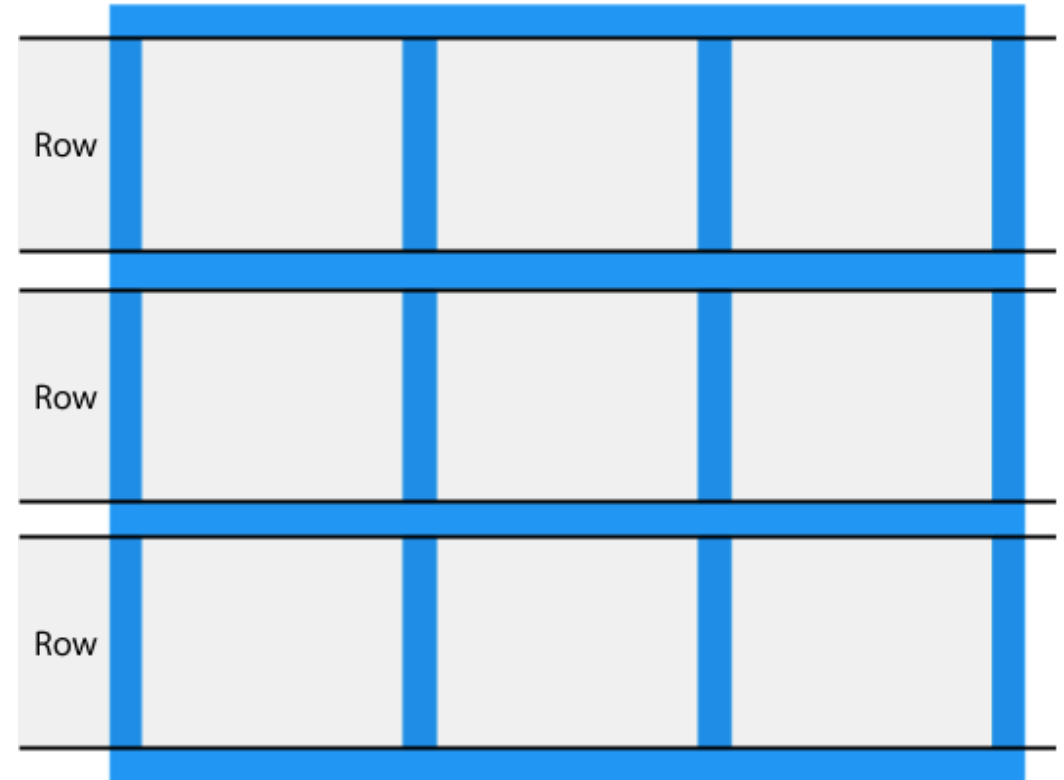
Columns

- De verticale lijnen van de **grid items** zijn de **kolommen**

Column	Column	Column

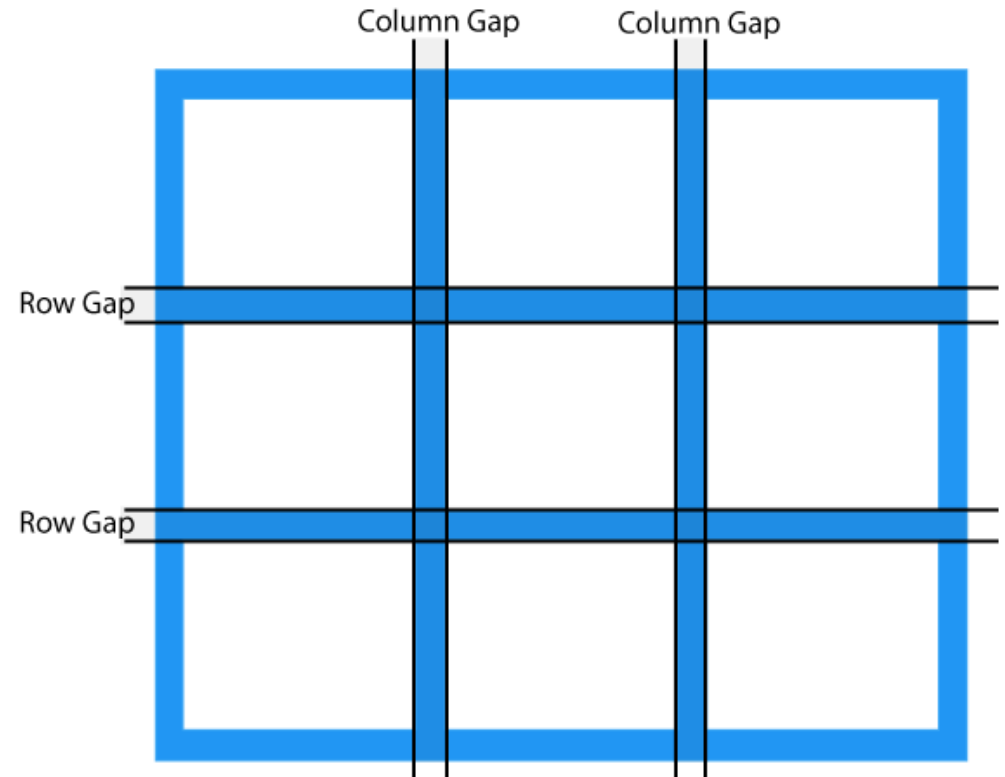
Rows

- ↘ De horizontale lijnen van de **grid items** zijn de **rijen**



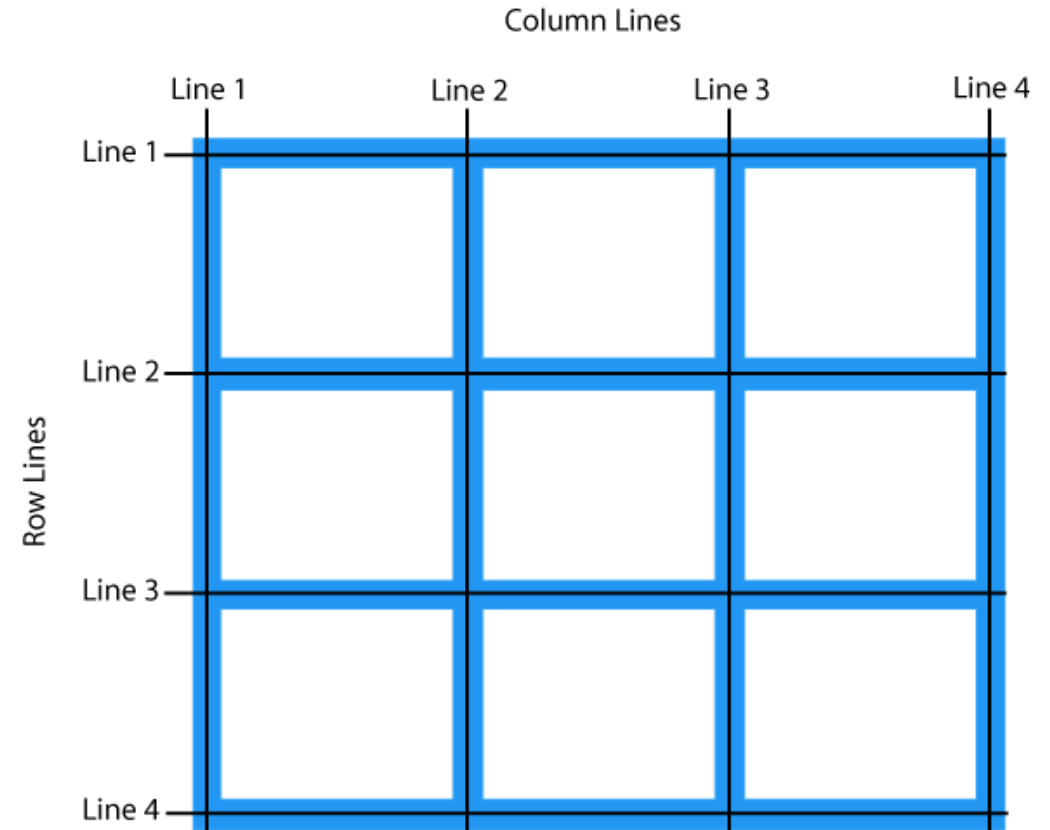
Gaps

- De spatie tussen lijnen van de grid items noemen we het **gap**
- Dit kunnen we instellen via CSS
 - × `column-gap: 50px;`
 - × `row-gap: 30px;`
 - × `gap: 50px;`



Lines

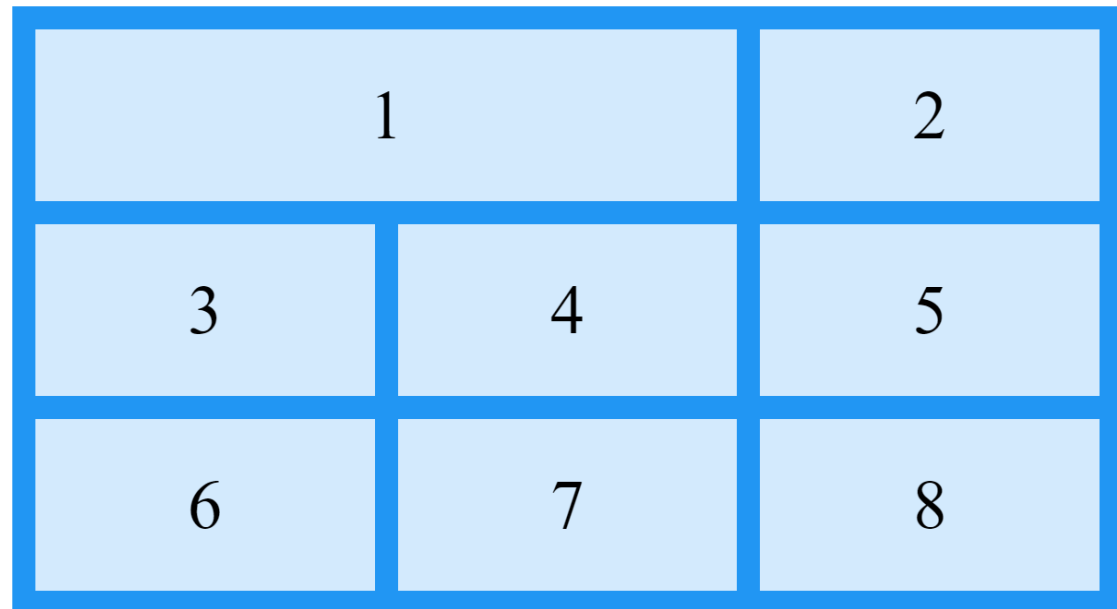
- De lijnen tussen de rijen noemen we **row lines**
- De lijnen tussen de kolommen noemen we de **column lines**



Lines

- ↘ Laat een grid item beginnen op column line 1 en eindigen op column line 3

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  gap: 10px;  
  background-color: #2196F3;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
  
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```



grid-template-columns

- Property geeft aan hoeveel **kolommen** het grid heeft
- Wanneer we **4 kolommen** willen met allemaal dezelfde breedte gebruiken we **auto**
- Als het grid **meer dan 4** grid items heeft, zal er automatisch een **nieuwe rij** worden toegevoegd

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto auto;  
}
```

grid-template-columns – the fr unit

- Elke **lengte-eenheid** kan gebruikt worden (procent/pixels/...)
- Grid introduceert een extra lengte-eenheid om flexibele rasters te creëren
- De **fr unit** vertegenwoordigt een **fractie** van de beschikbare ruimte in de rastercontainer
- Een **mix** van **eenheden** is toegestaan (procent/pixels/fr...)

grid-template-rows

- Property geeft de **hoogte** van elke rij aan

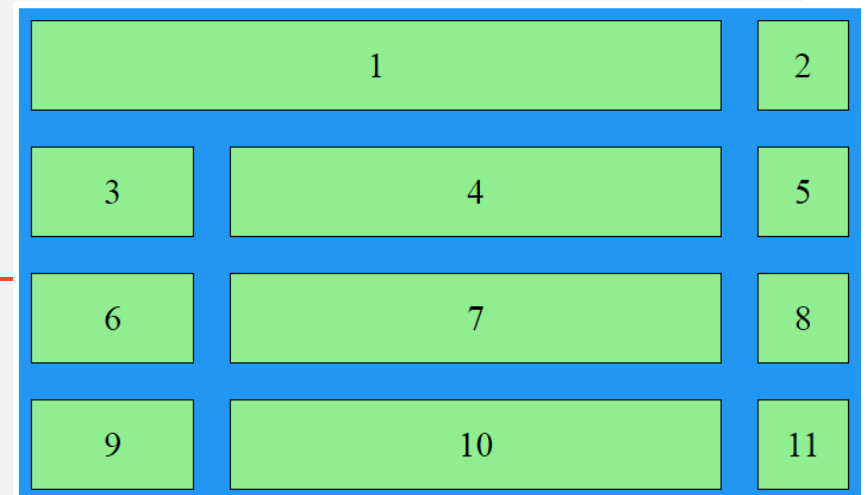
```
.grid-container {  
  display: grid;  
  grid-template-rows: 80px 200px;  
}
```

1	2	3	4
5	6	7	8

Voorbeeld

```
.grid-container {  
  display: grid;  
  grid-gap: 30px 30px;  
  grid-template-columns: 20% 60% auto;  
  background-color: #2196f3;  
  padding: 10px;  
}  
  
.grid-container>div {  
  background-color: lightgreen;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 30px;  
  text-align: center;  
}  
  
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```






```
<div class="grid-container">  
  <div class="grid-item item1">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
  <div class="grid-item">8</div>  
  <div class="grid-item">9</div>  
  <div class="grid-item">10</div>  
  <div class="grid-item">11</div>  
</div>
```



CSS FLEXBOX

CSS Flexbox

- Lay-out met responsive **rijen** en **kolommen**
- Ongeveer hetzelfde als CSS Grid, maar werkt anders
- **Focus** ligt **meer** op **rijen**, **niet** op **kolommen**
- Ondersteund door alle moderne browsers

				
29.0	11.0	22.0	10	48

Parent element (container)

- Flex container wordt flexibel als we de **display** property op flex zetten

```
.flex-container {  
  display: flex;  
  background-color: DodgerBlue;  
}
```

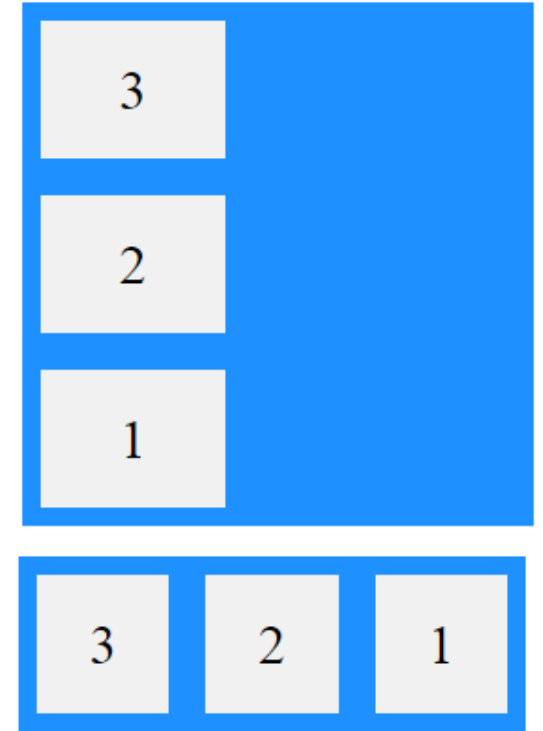
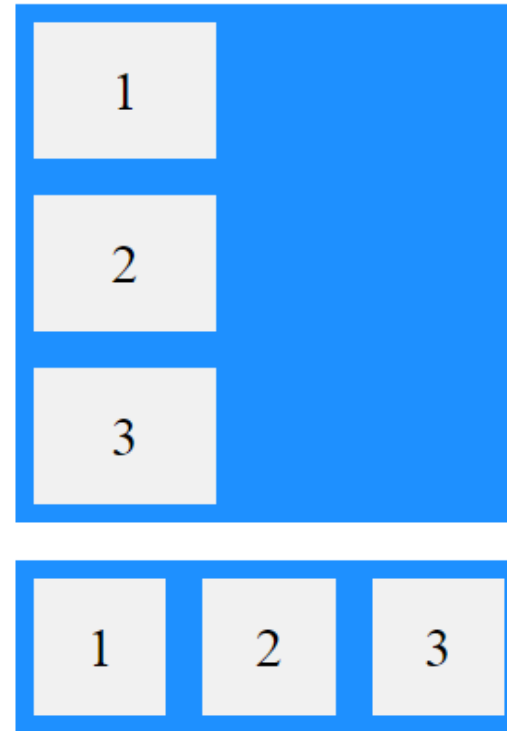
```
.flex-container > div {  
  background-color: #f1f1f1;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}
```



```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
</div>
```

Flex-direction

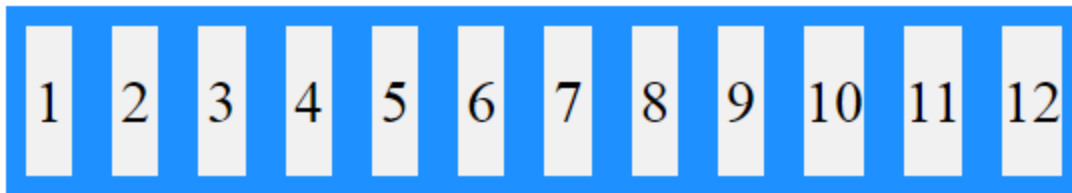
- Geeft aan **welke richting** de container de flex items zal stapelen
 - × `flex-direction: column;`
 - × `flex-direction: column-reverse;`
 - × `flex-direction: row;`
 - × `flex-direction: row-reverse;`



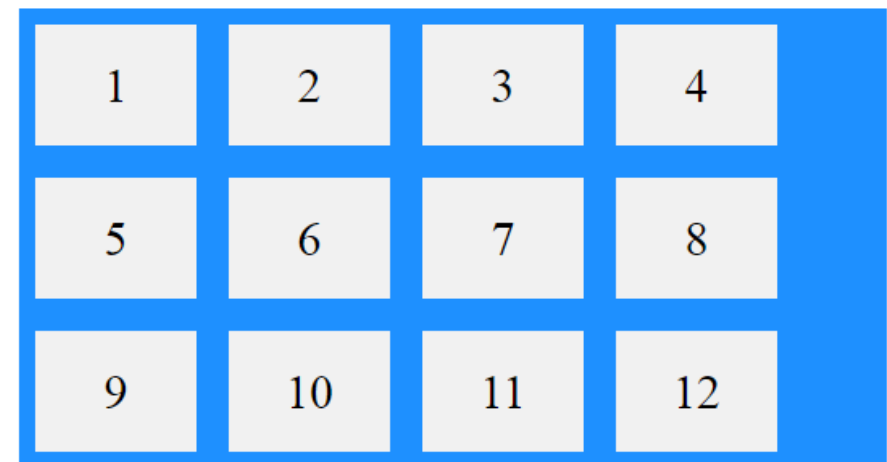
Flex-wrap

➤ **Flex-wrap** property definieert of de flex items zullen wrappen of niet

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
  background-color: DodgerBlue;  
}
```



```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
  background-color: DodgerBlue;  
}
```

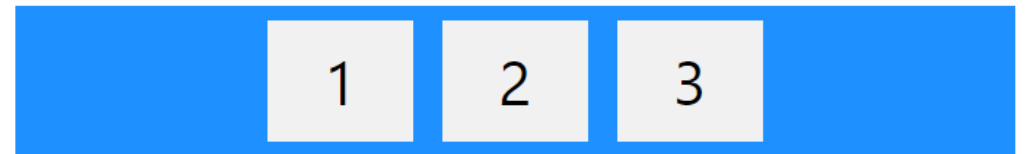


Justify-content

- **Justify-content** property wordt gebruikt om de flex items uit te lijnen
 - × **justify-content: flex-start;**
 - × **justify-content: flex-end;**
 - × **justify-content: space-around;**
 - × **justify-content: space-between;**

```
.flex-container {  
  display: flex;  
  justify-content: center;  
  background-color: DodgerBlue;  
}
```

```
.flex-container > div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;  
  font-size: 30px;  
}
```



Bronnen

Positioning

- https://www.w3schools.com/css/css_positioning.asp
- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Positioning

Grid

- https://www.w3schools.com/css/css_grid.asp
- https://www.w3schools.com/css/css_grid_container.asp
- <https://developer.mozilla.org/en-US/docs/Glossary/Grid>
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout
- <https://github.com/fireship-io/224-animated-css-grid>

Flexbox

- https://www.w3schools.com/css/css3_flexbox.asp
- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
- <https://www.youtube.com/watch?v=K74l26pE4YA>

Oefeningen

INDIVIDUEEL OF PAIR PROGRAMMING