THOMAS MORE | University of Applied Sciences

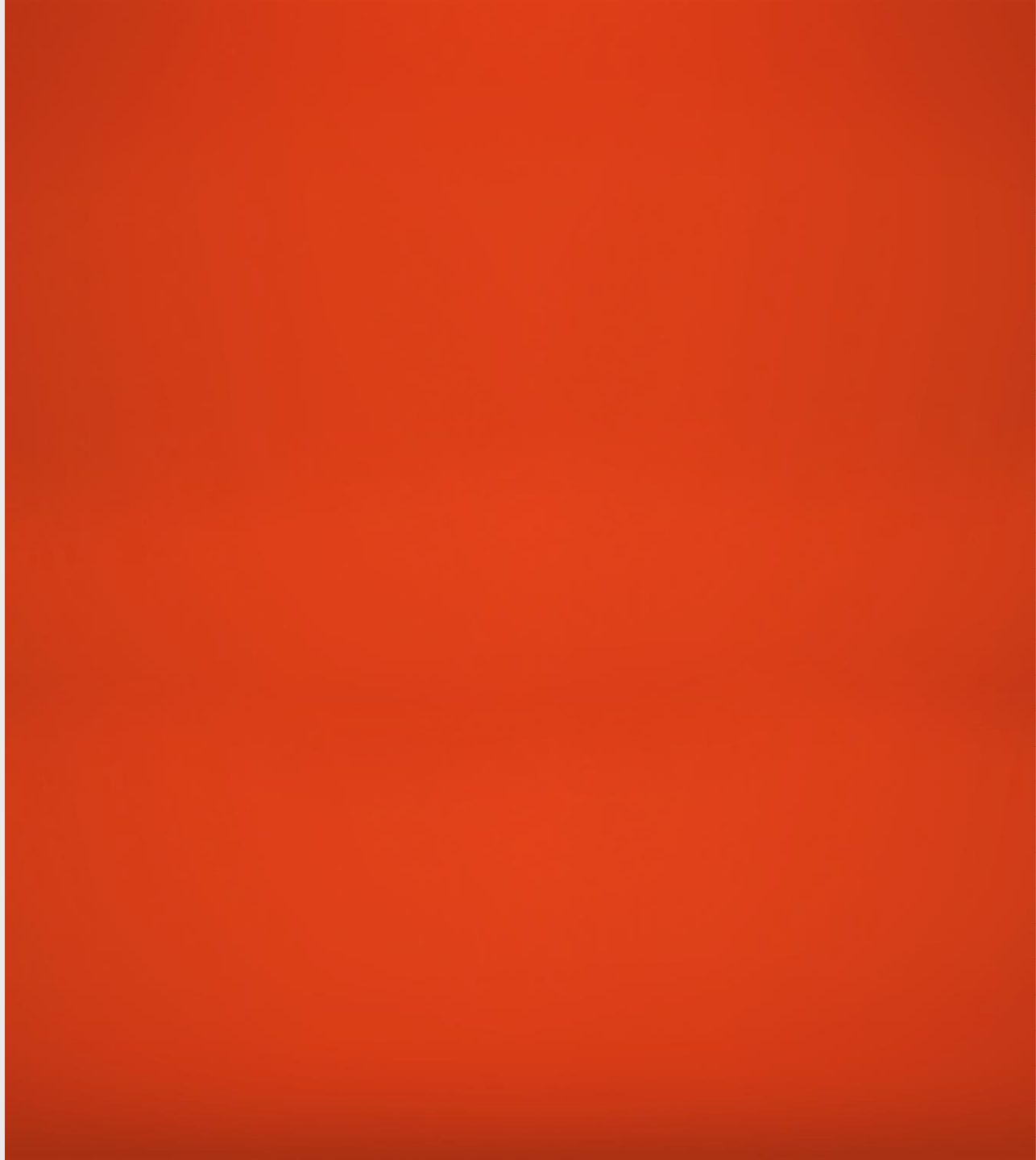
Statische websites

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
  - Colums
  - 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

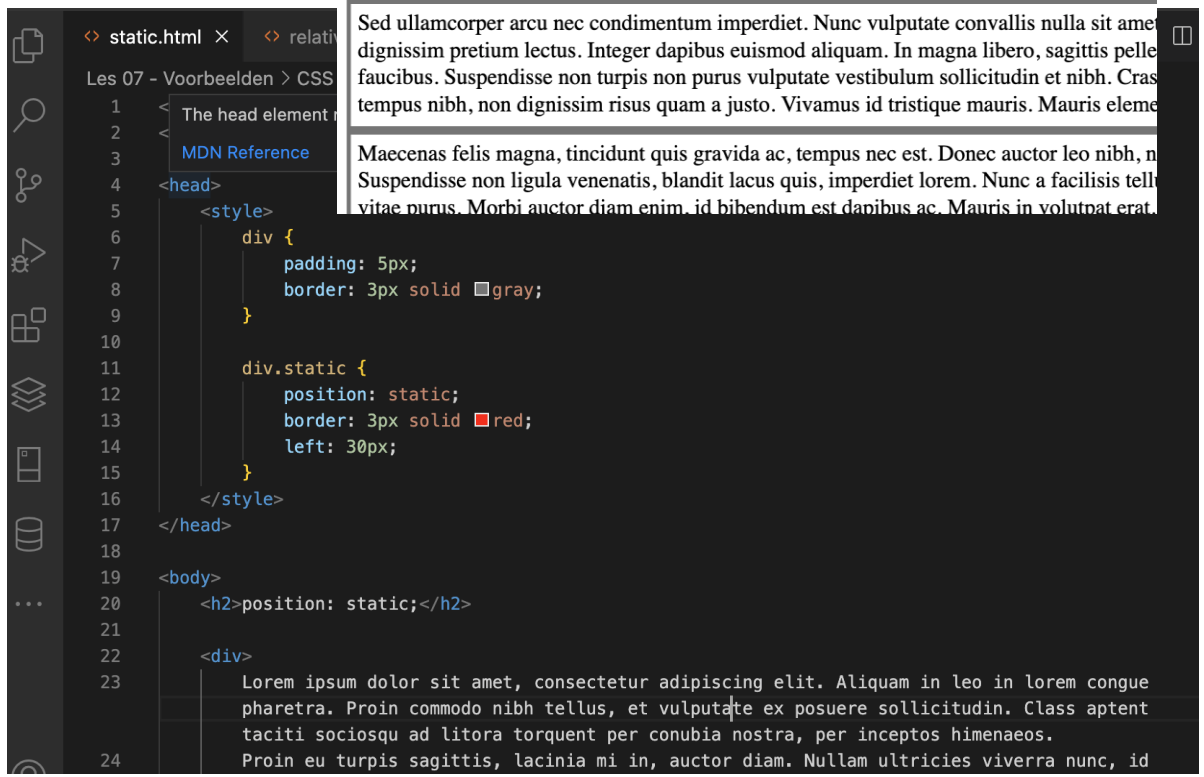


Position property

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

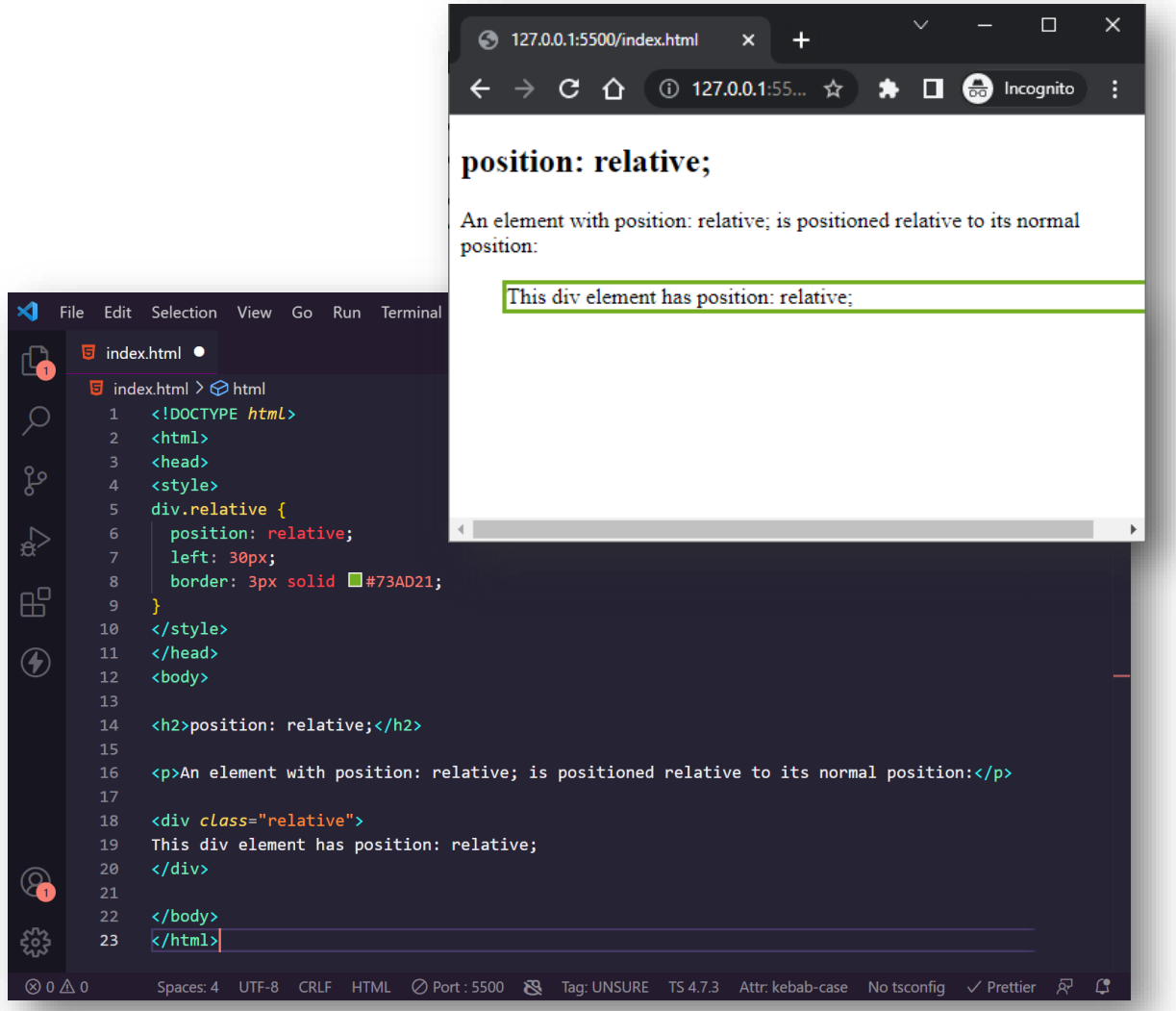
position: static;

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



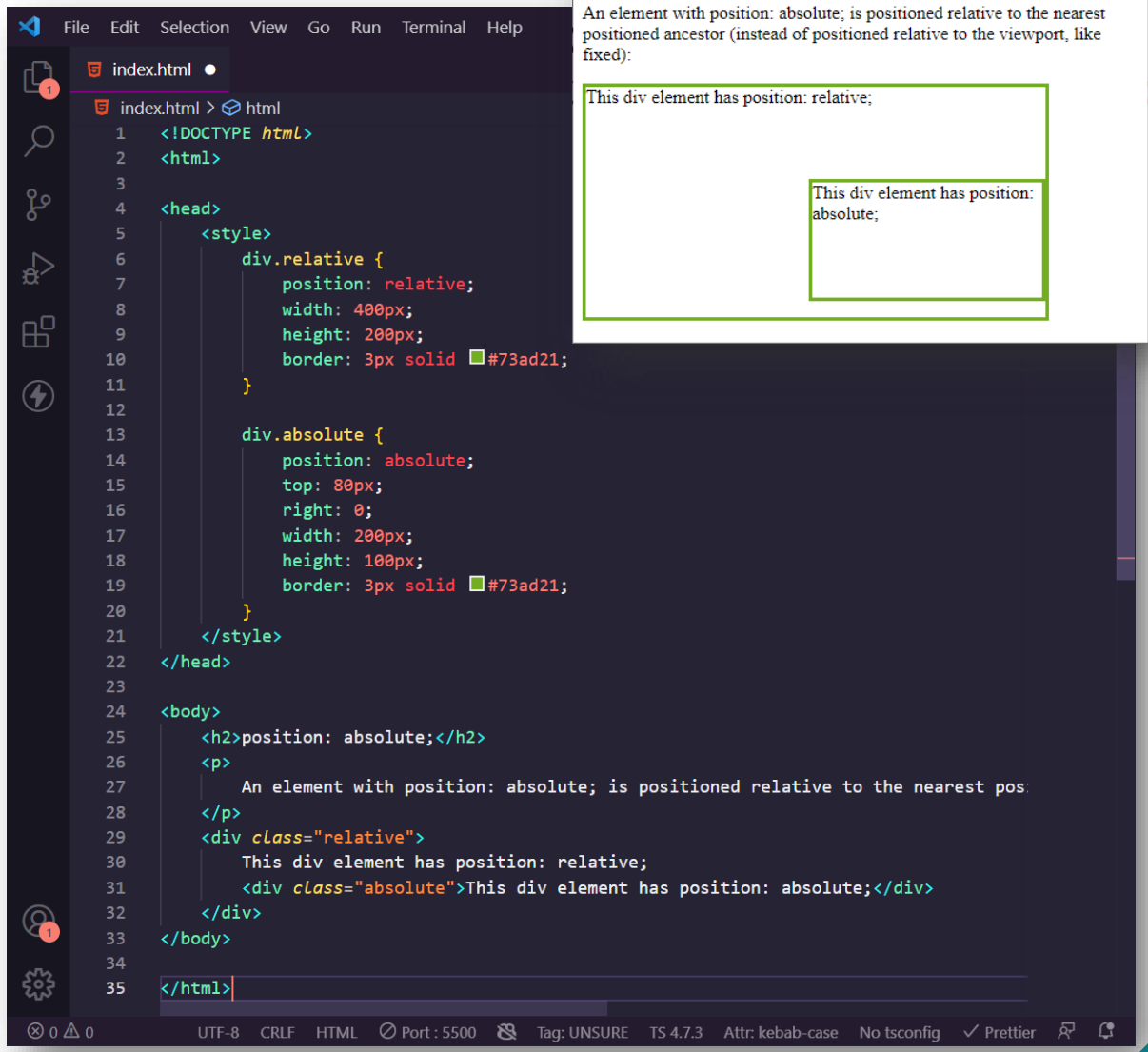
position: relative;

- 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



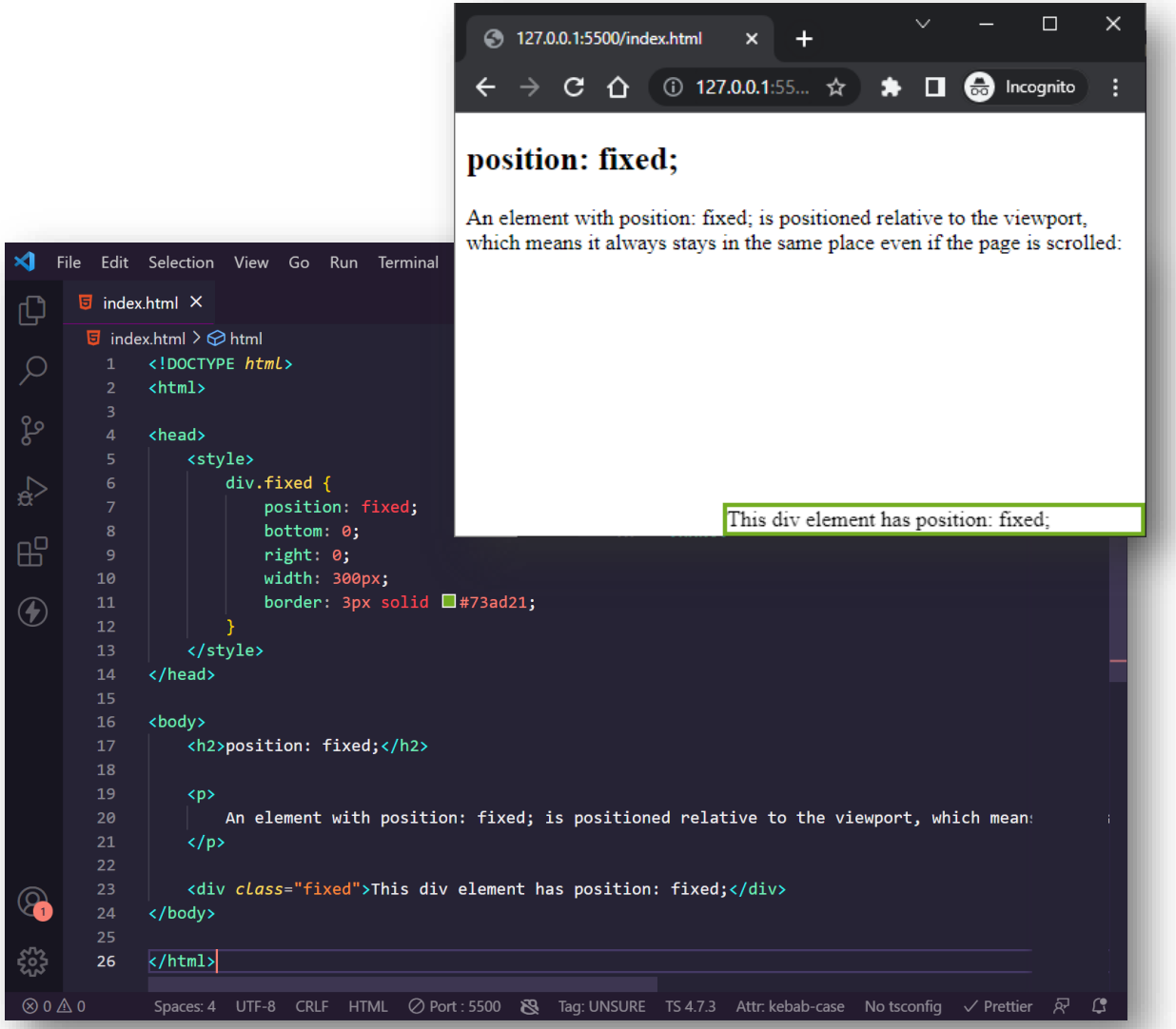
position: absolute;

- 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;



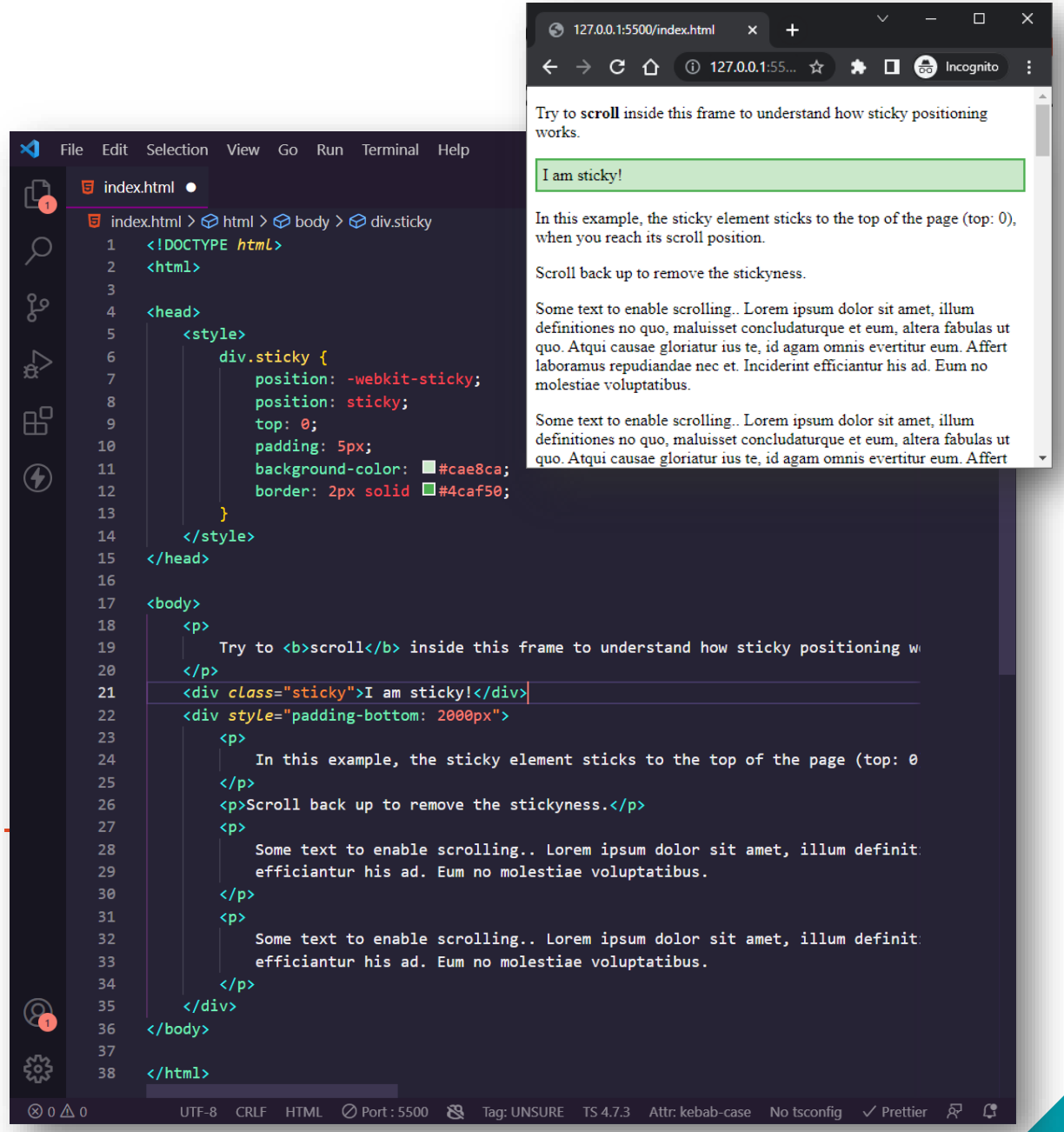
position: fixed;

- 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



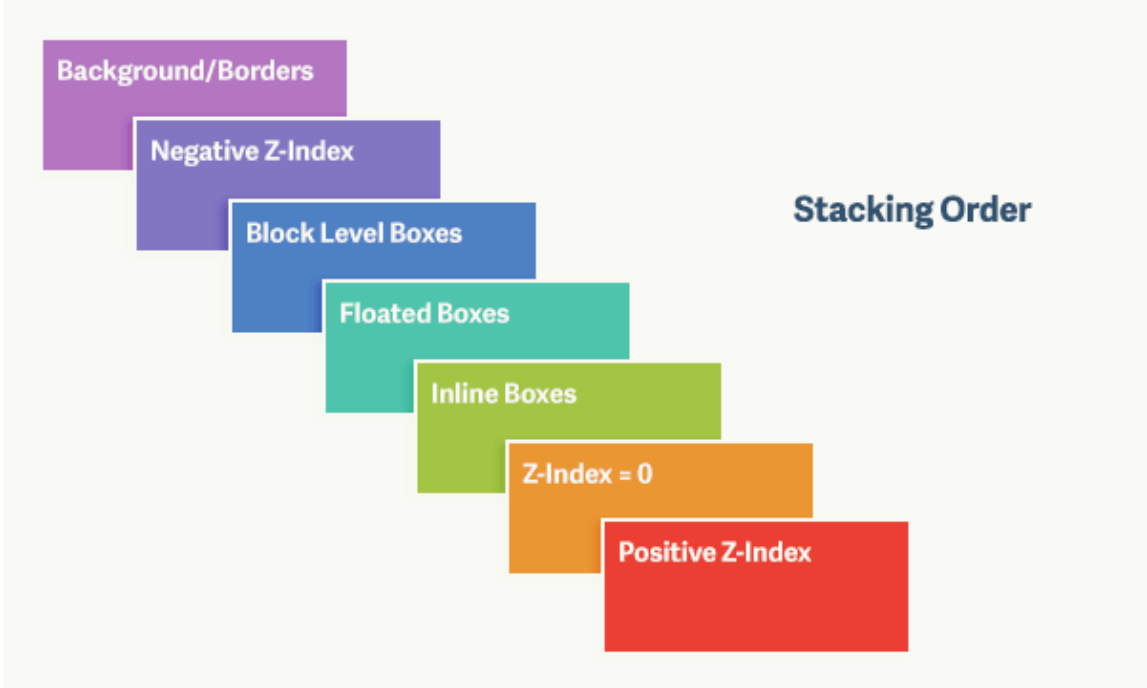
position: sticky;

- 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



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

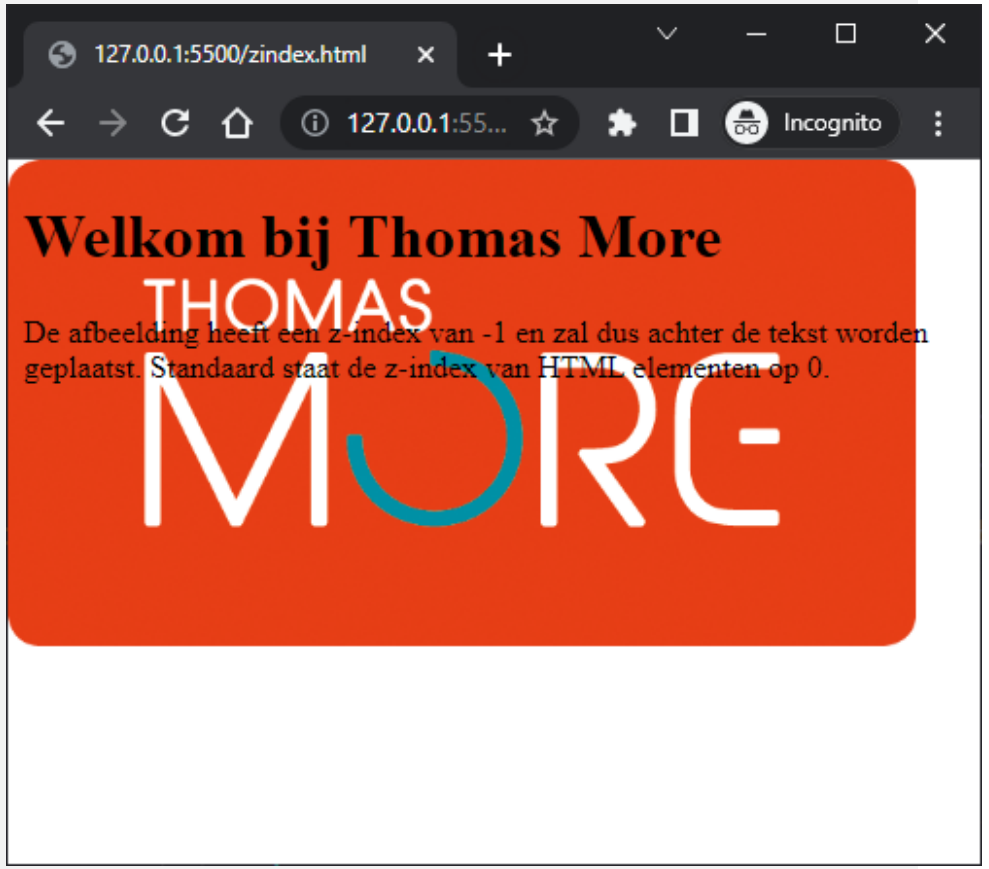


Voorbeeld

File Edit Selection View Go Run Terminal Help zindex.html - Position - Visual Studio Code

zindex.html

```
<!DOCTYPE html>
<html>

<head>
    <style>
        img {
            position: absolute;
            left: 0px;
            top: 0px;
            z-index: -1;
        }
    </style>
</head>

<body>
    <h1>Welkom bij Thomas More</h1>
    <img src="tm.png" height="250" />
    <p>
        De afbeelding heeft een z-index van -1 en zal dus achter de tekst worden geplaatst.
    </p>
</body>

</html>
```



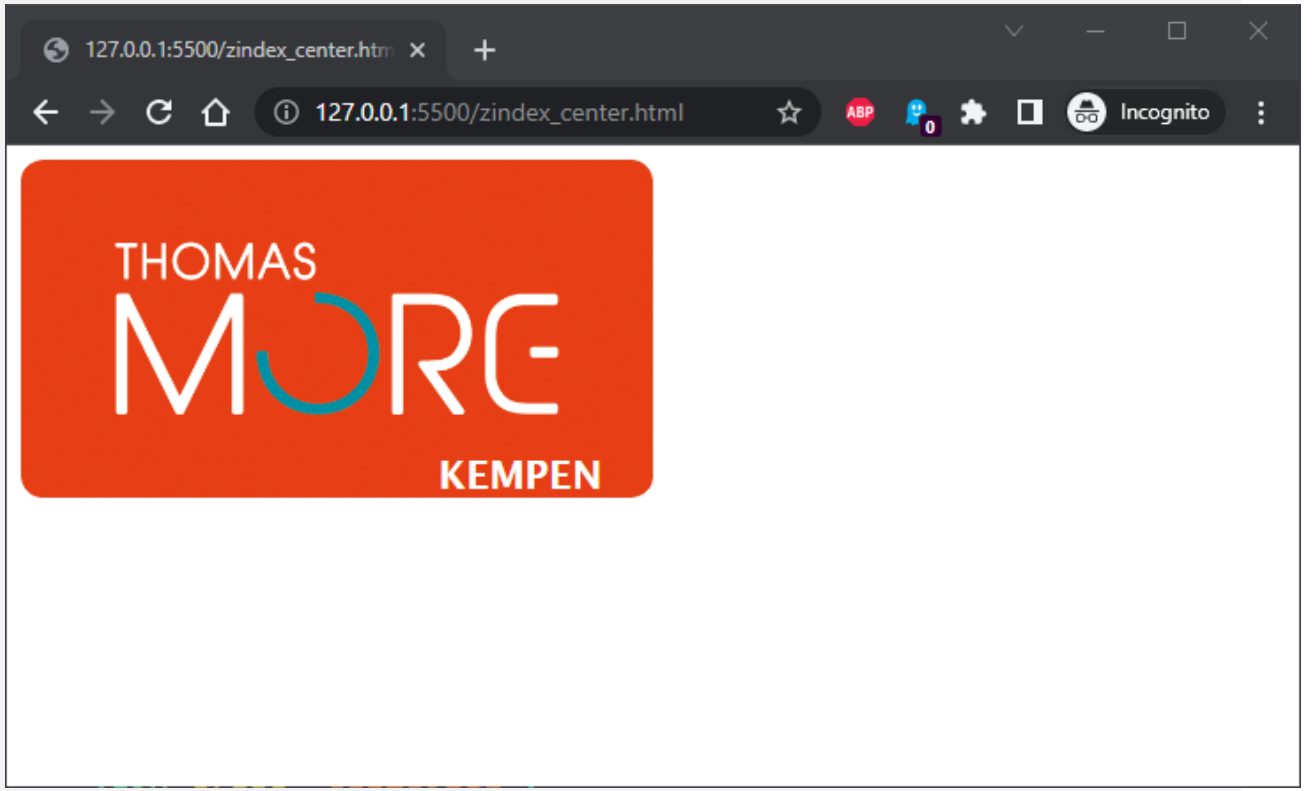
Voorbeeld

File Edit Selection View Go Run Terminal Help zindex_center.html - Position - Visual Studio Code

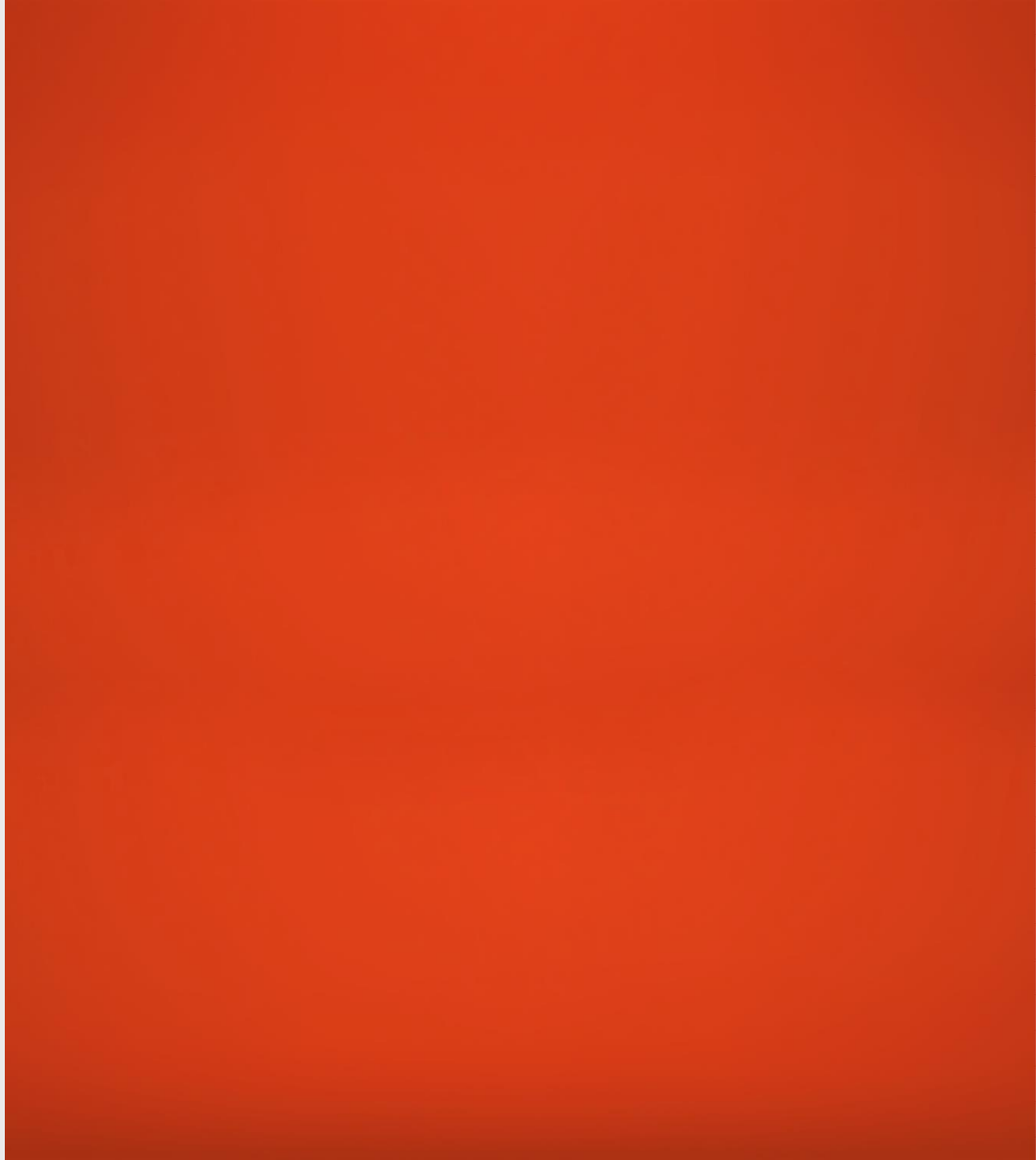
zindex_center.html

```
<!DOCTYPE html>
<html>

<head>
    <style>
        .container {
            position: relative;
        }

        .kempen {
            position: absolute;
            right: 75px;
            top: 85%;
            width: 100%;
            text-align: center;
            font-size: 22px;
            font-weight: 700;
            color: white;
            text-transform: uppercase;
            font-family: "Lucida Sans", Verdana, sans-serif;
        }

        img {
            width: 50%;
            height: auto;
        }
    </style>
</head>

<body>
    <div class="container">
        <img src="tm.png" alt="TM logo" />
        <div class="kempen">Kempen</div>
    </div>
</body>

</html>
```

Spaces: 4 UTF-8 CRLF HTML Port : 5500 TS 4.7.3 Tag: UNSURE No tsconfig Attr: kebab-case Prettier

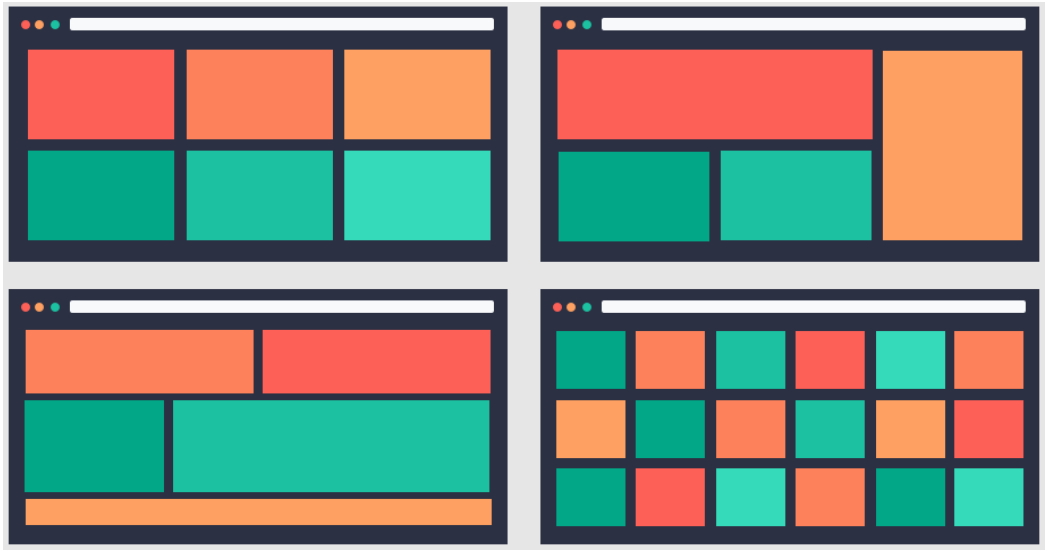


CSS Grid



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**

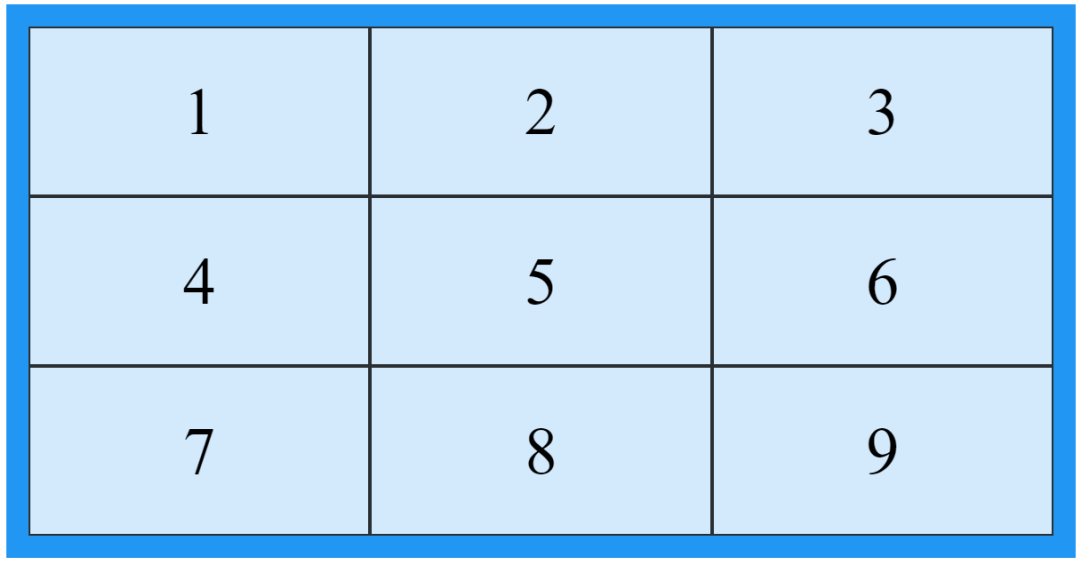
https://www.w3schools.com/css/tryit.asp?filename=trycss_grid

```
.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

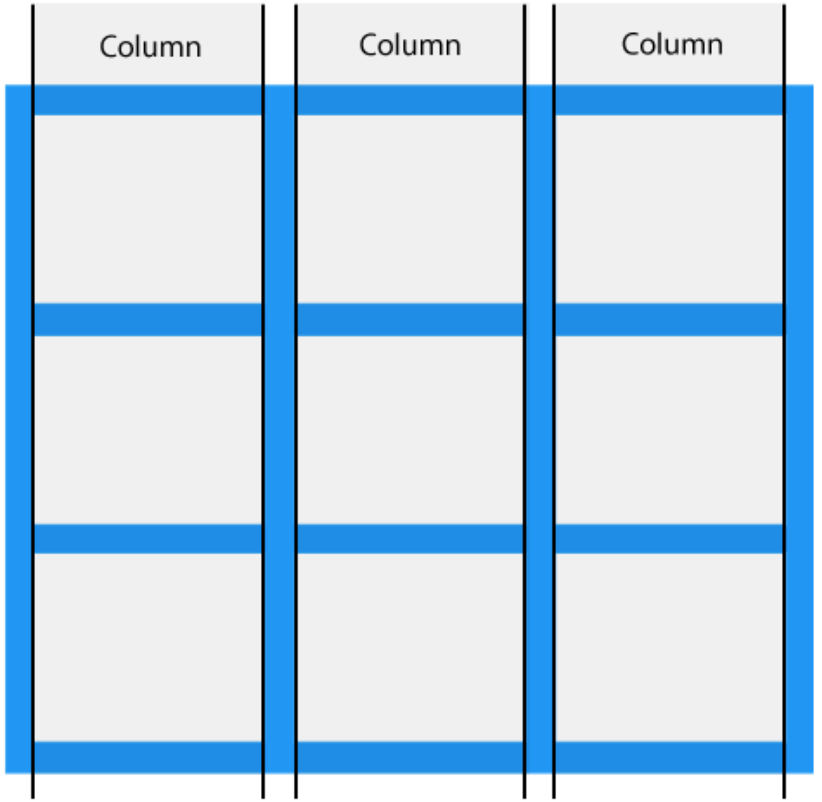
- Een grid bestaat uit één parent element met één of meerdere child elementen

```
.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;
}
```



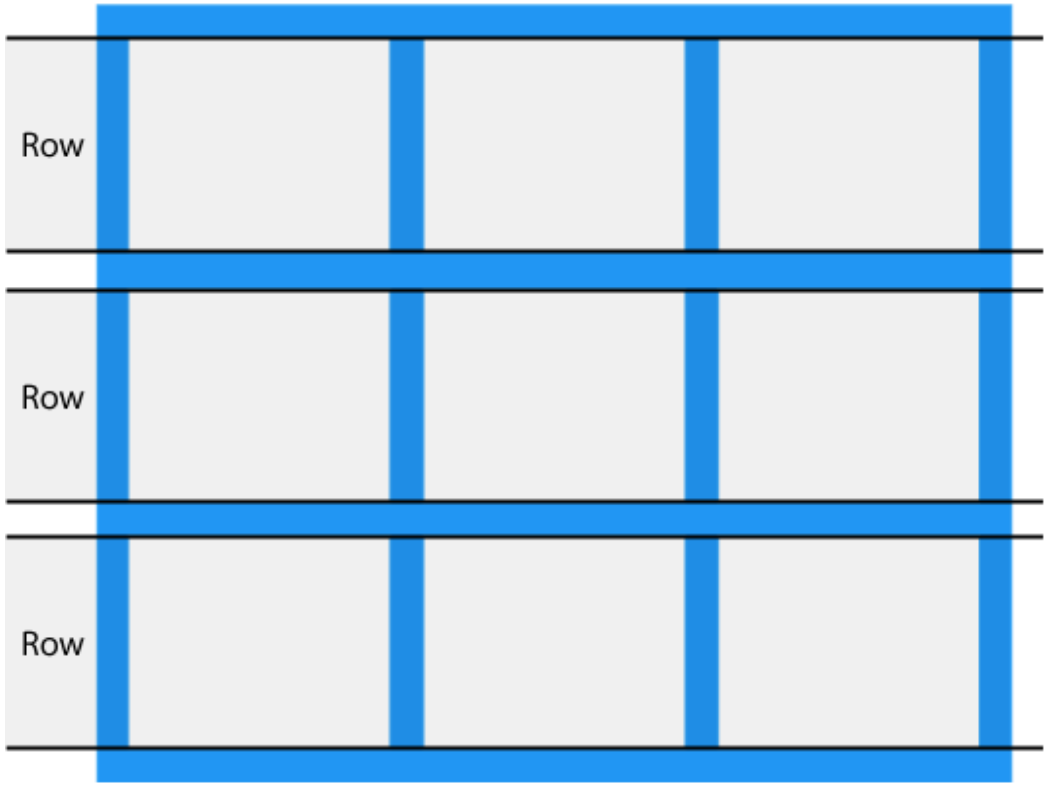
Columns

- De verticale lijnen van de grid items zijn de kolommen



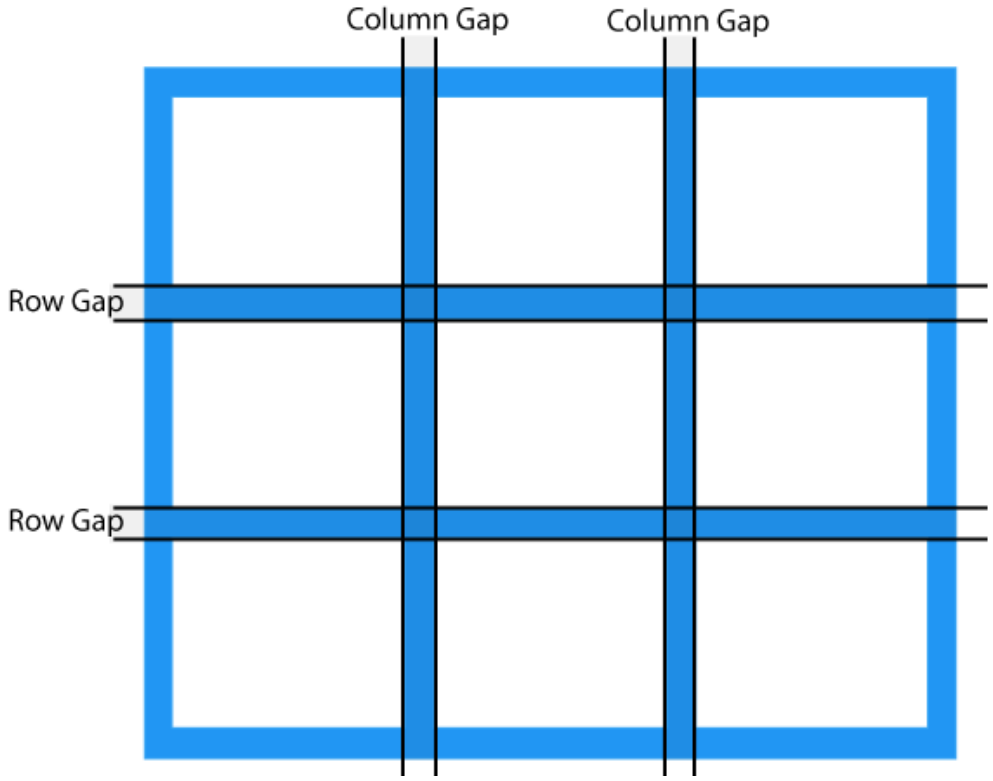
Rows

- De horizontale lijnen van de grid items zijn de rijen



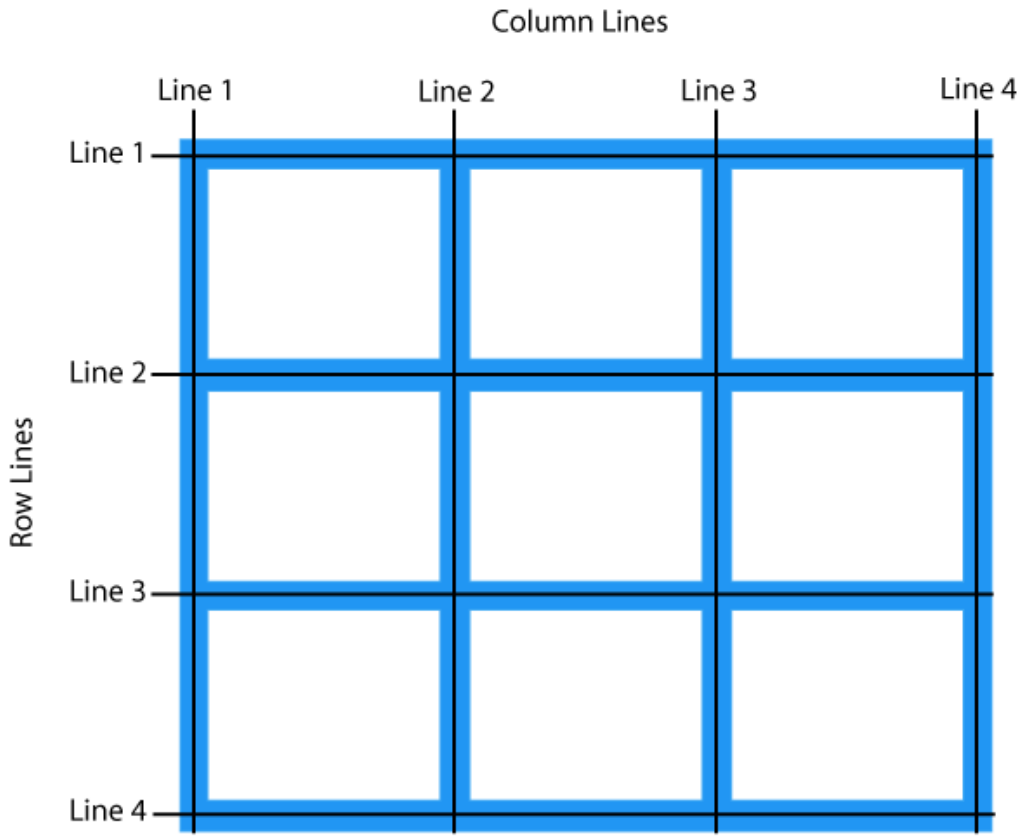
Gaps

- 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

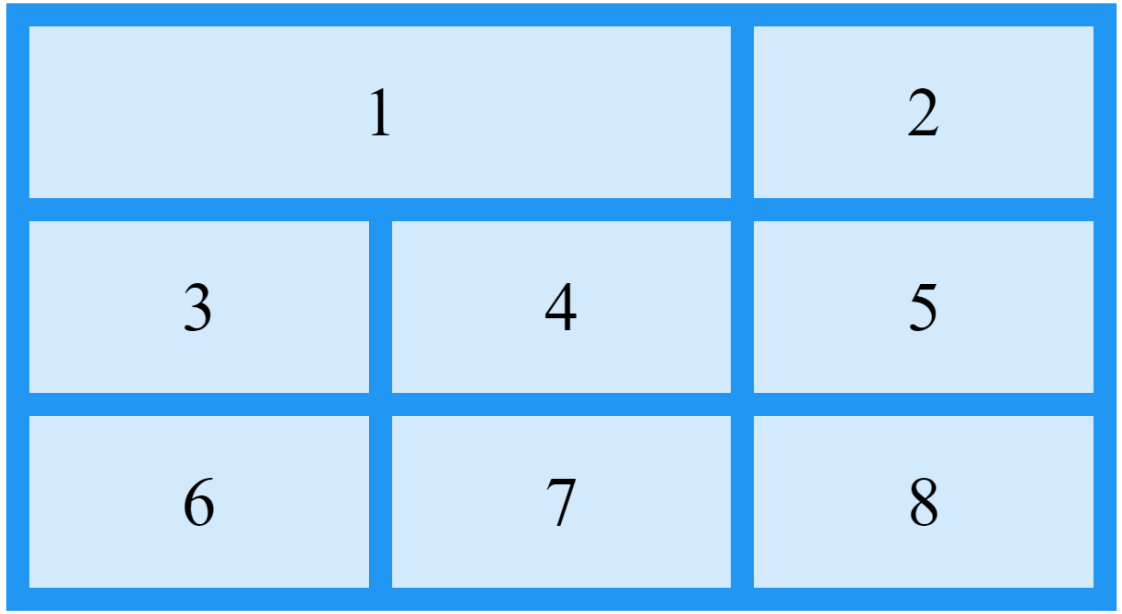
- De lijnen tussen de rijen noemen we **row lines**
- De lijnen tussen de kolommen noemen we de **column lines**



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;
}
```

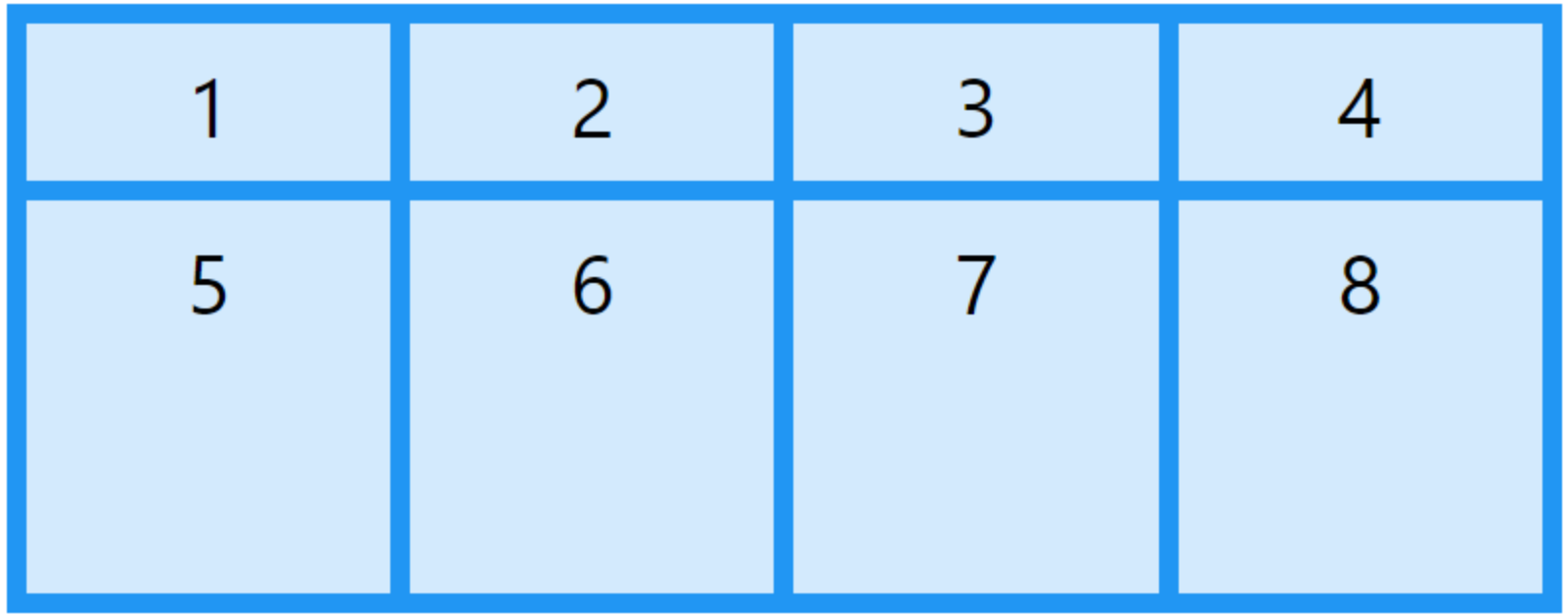
grid-template-columns – the fr unit

- 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

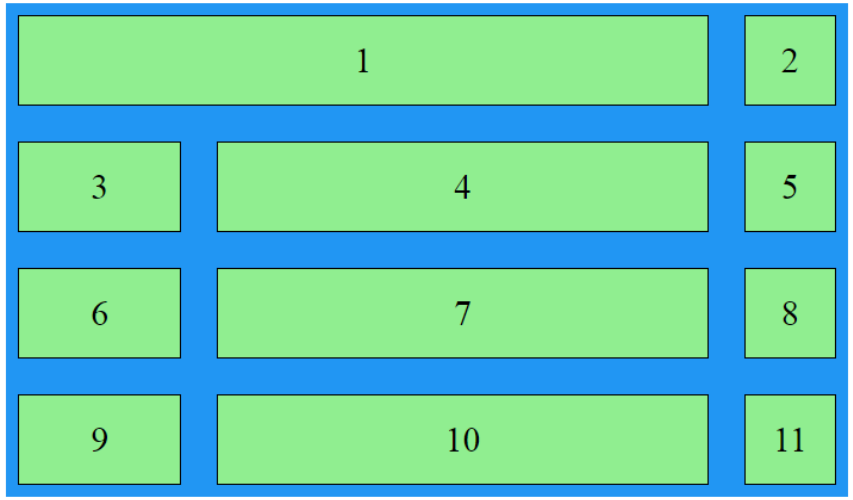
```
.grid-container {
  display: grid;
  grid-template-rows: 80px 200px;
}
```



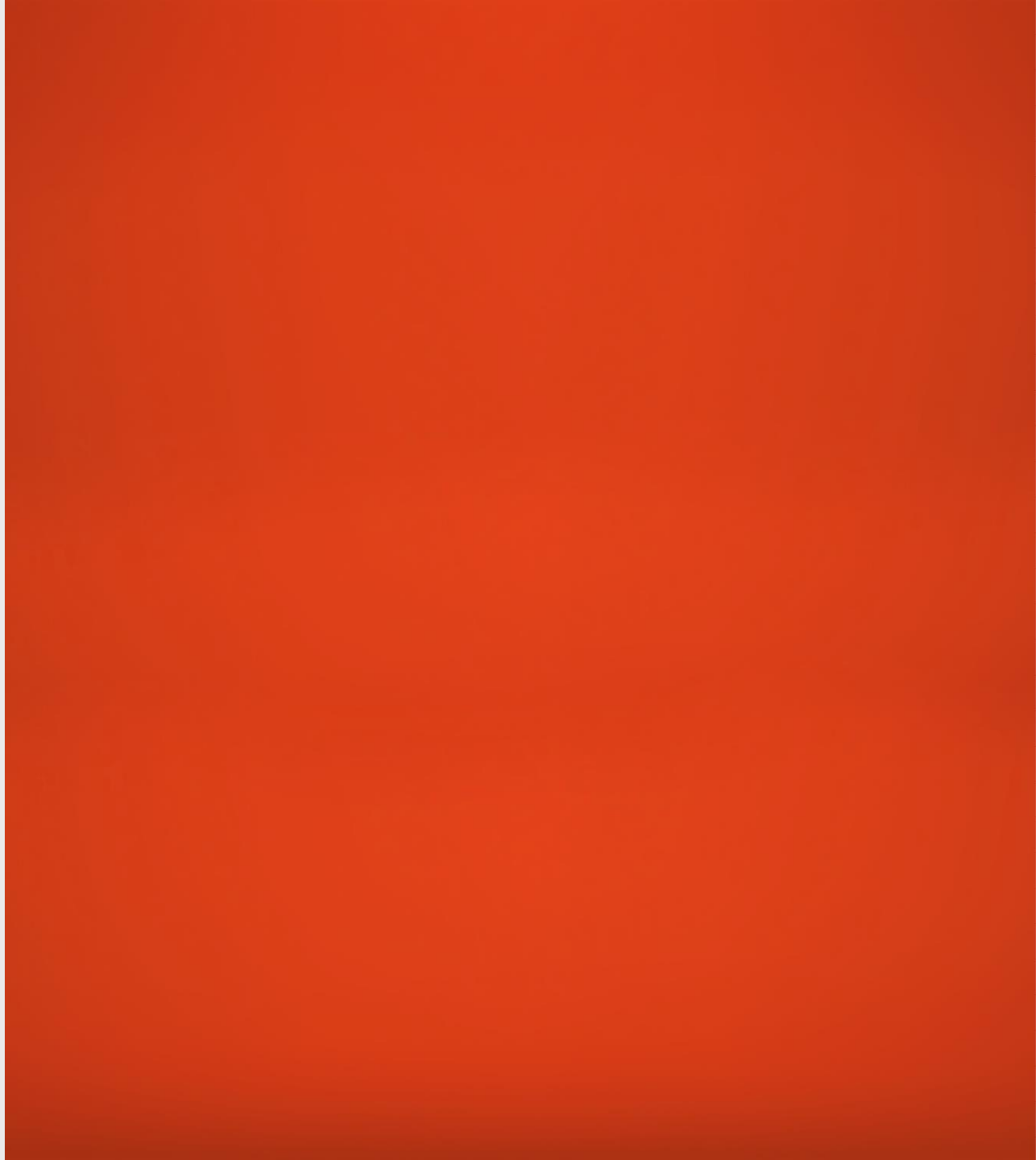
Voorbeeld

```css
.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;
}
```

```html
<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)

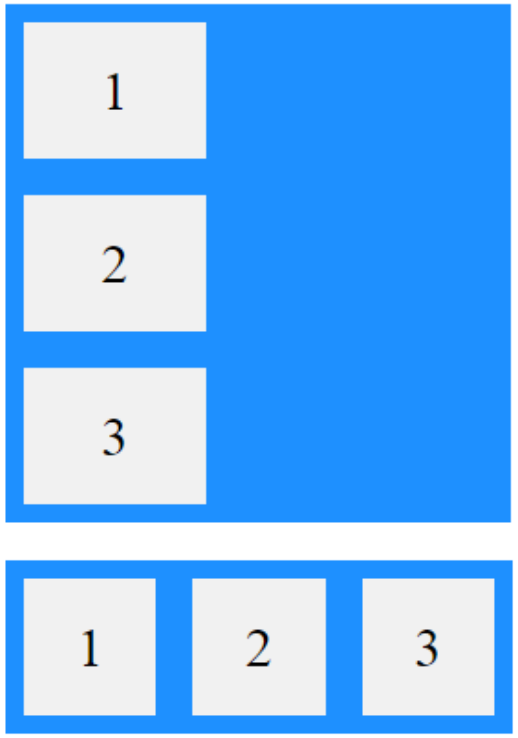
- Flex container wordt flexibel als we de display property op flex zetten

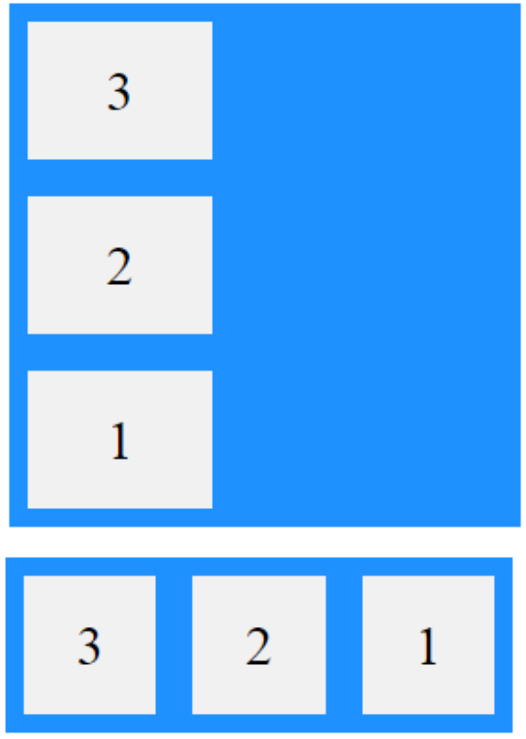
```
.flex-container {
  display: flex;
  background-color: DodgerBlue;
}

.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;

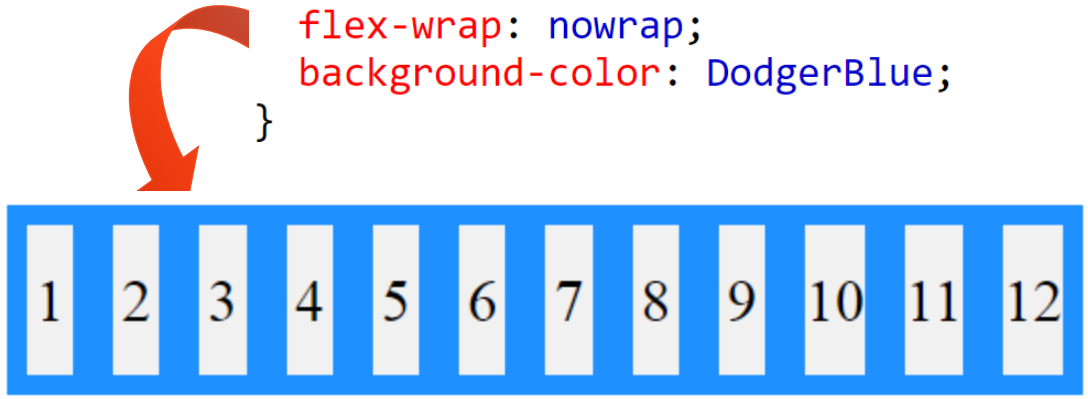




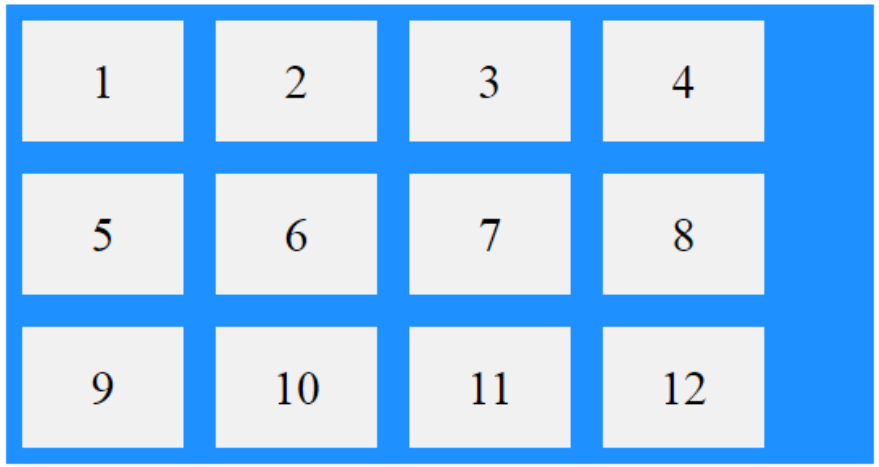
Flex-wrap

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

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



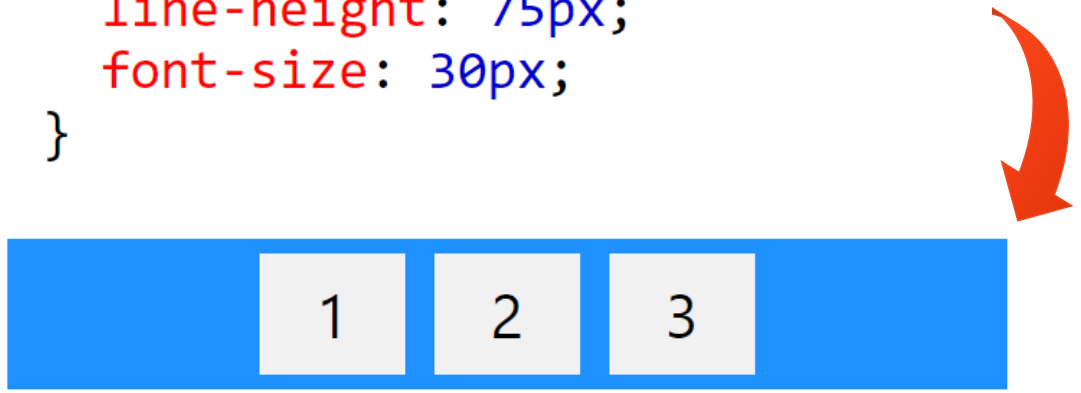

```
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: DodgerBlue;
}
```



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

THOMAS MORE | University of Applied Sciences

Oefeningen

INDIVIDUEEL OF PAIR PROGRAMMING