



# Frontend Foundations

CSS Grid

# Inhoud

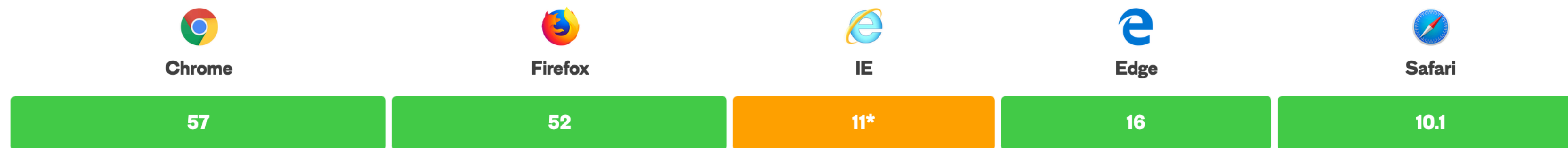
- Basics
- Zelf studie
- <http://cssgridgarden.com/>

# CSS Grid Layout

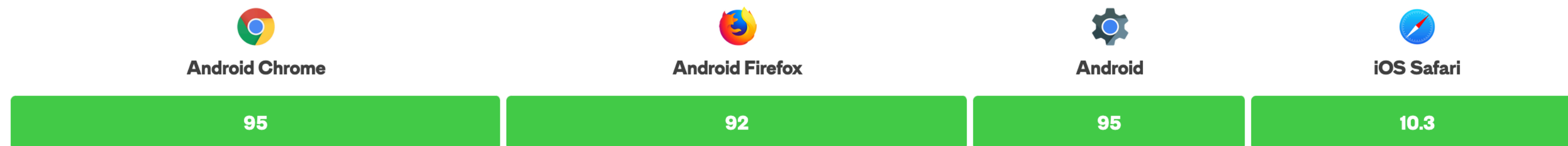
- Flexbox: simple **1-dimensional** layouts
- CSS Grid: complex **2-dimensional** layouts
  - Columns & rows
  - Global layouts, dashboards, ...
  - Krachtig en uitgebreid
- CSS Grid + Flexbox = the 💣 of the 💩
- Browser support: Sinds maart 2017 👍
- CSS tricks: <https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-auto-columns-rows>
- Playground: <https://codepen.io/simoncoudeville-nmct/pen/JeMPPZ>

# Browser support

## Desktop



## Mobile / Tablet



<http://caniuse.com/#feat=css-grid>

# CSS Grid Layout

- Grid container: parent
  - Bepaalt de layout
  - `display: grid`
- Grid items: direct children
  - Volgen de layout
  - Default source order van de children maar laat toe om de order zowel vertical als horizontaal te veranderen.

# Grid container

html

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

Container is de grid container

# Grid item

html

```
<div class="container">  
  <div class="item"></div>  
  <div class="item">  
    <p class="sub-item"></p>  
  </div>  
  <div class="item"></div>  
</div>
```

Item elementen zijn grid items. Sub-item niet.

# Grid Line

- De scheidingslijnen die de structuur van het grid bepalen
- Vertical: column grid line
- Horizontal: Row grid line



Column grid line



# Grid Track

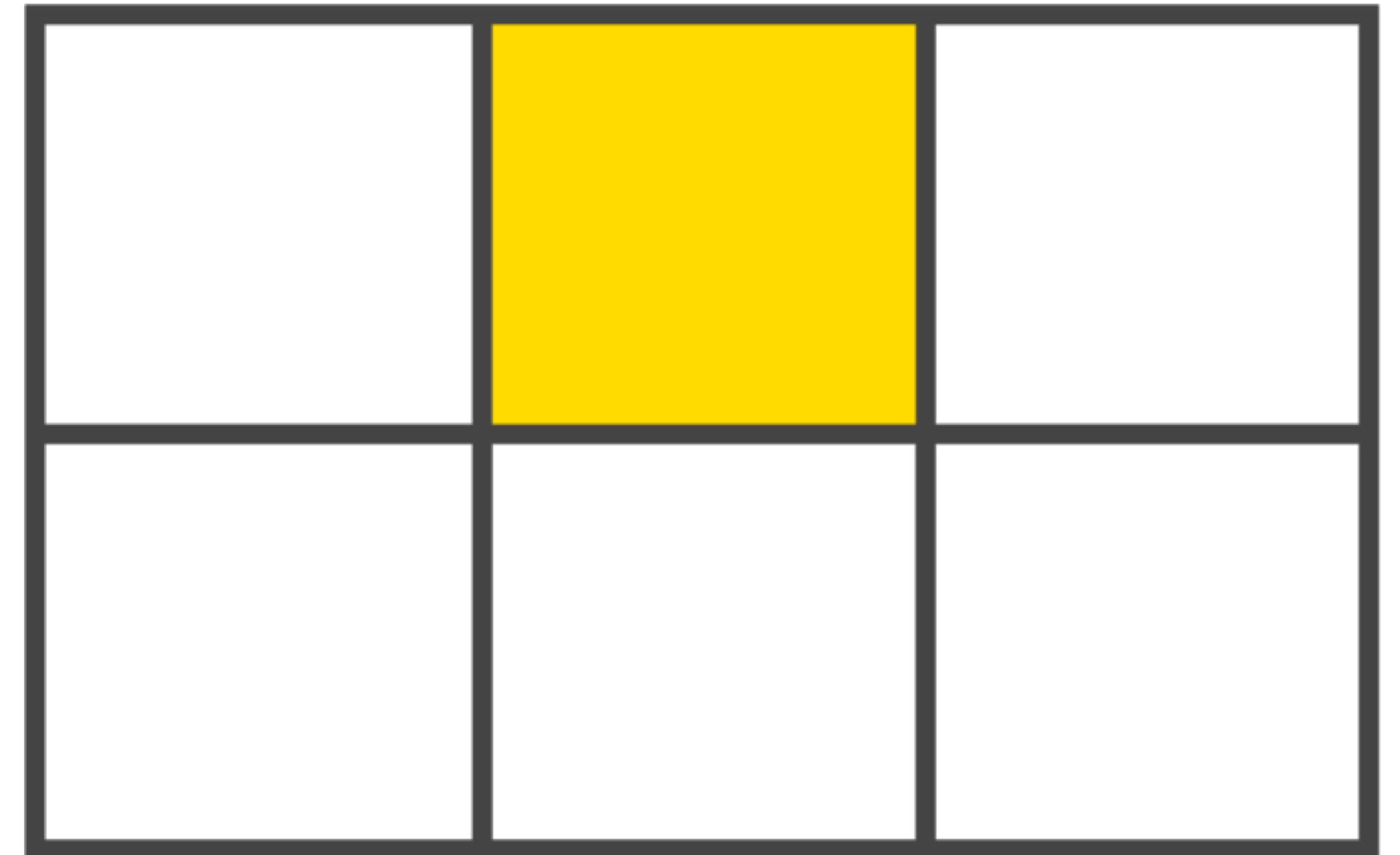
- De ruimte tussen 2 opeenvolgende grid lines.
- Vertical: columns
- Horizontal: Rows



Grid track tussen de 2de en de 3de row grid lines

# Grid Cell

- De ruimte tussen 2 opeenvolgende row & column grid lines.



Grid cell tussen row grid lines 1 & 2,  
en column grid lines 2 & 3.

# Grid Area

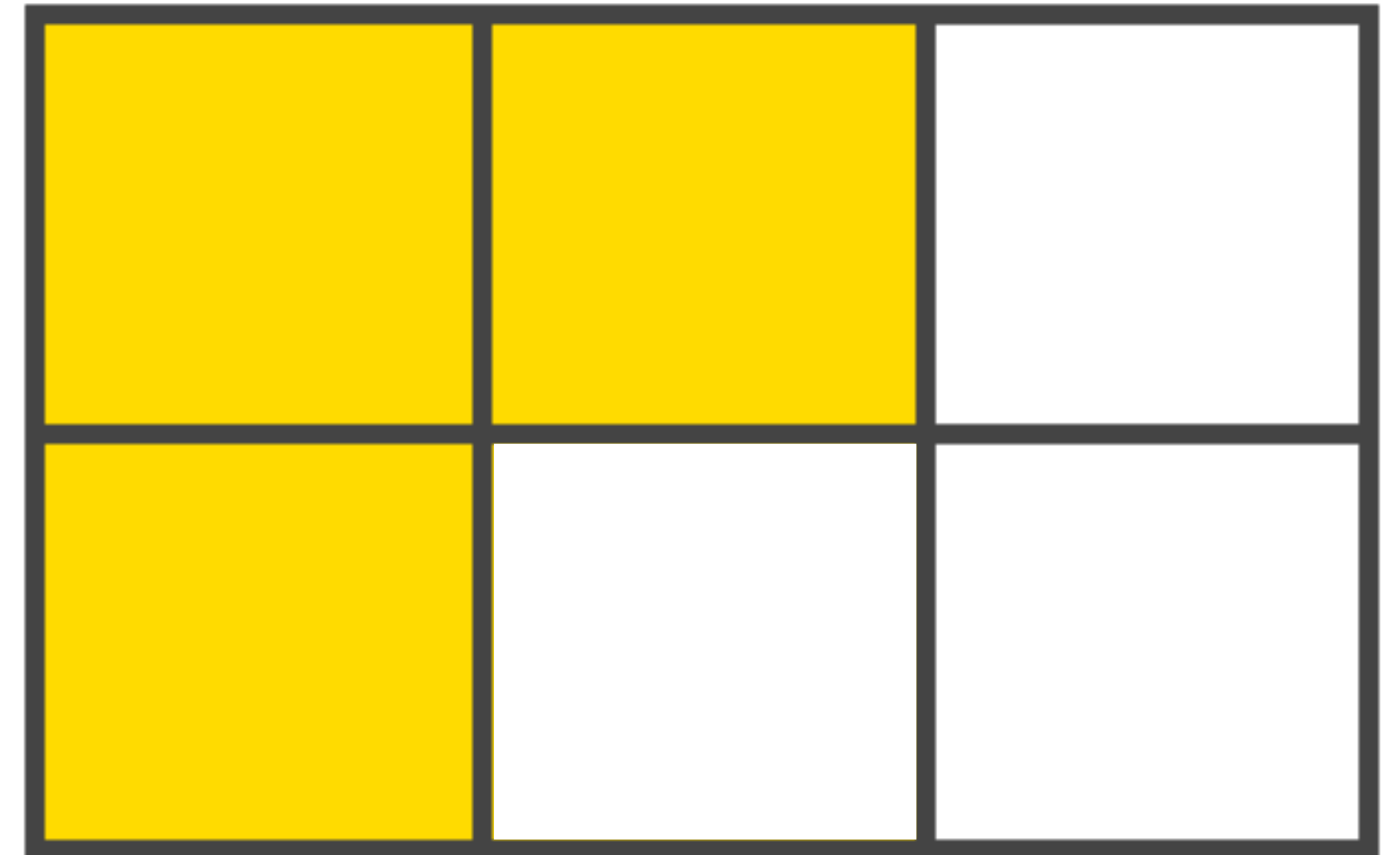
- De ruimte tussen 2 row & column grid lines.
- Kan meerder grid cells bevatten.



Grid area tussen row grid lines 1 & 3,  
en column grid lines 1 & 3.

# Grid Area

- De ruimte tussen 2 row & column grid lines.
- Kan meerder grid cells bevatten.



Dit kan niet. Geen Tetris achtige vormen

# Properties voor de parent (Grid container)

- display: grid | inline-grid
- **Grid tracks, lines en area's definiëren:**
  - grid-template-columns
  - grid-template-rows
  - grid-template-areas
  - grid-template (shorthand)
- **Gap (gutter)**
  - column-gap
  - row-gap
  - gap (shorthand)
- **Alignment**
  - justify-items
  - align-items
  - place-items
  - justify-content
  - align-content
  - place-content
- **auto-generated grid tracks**
  - grid-auto-columns
  - grid-auto-rows
  - grid-auto-flow
- grid: (shorthand)

# Properties voor de child (Grid items)

- **Location binnen de container**
  - grid-column-start
  - grid-column-end
  - grid-row-start
  - grid-row-end
  - grid-column
  - grid-row
  - grid-area
- **Alignment:**
  - justify-self
  - align-self
  - place-self

# gap

- De gutter tussen opeenvolgende items (niet links of rechts boven of onder)
- column-gap
- row-gap
- gap: <row-gap> <column-gap>

# grid-template-columns

## grid-template-rows

- Definiëren de columns en rows
- Meerdere values gescheiden door spaties
- De values staan voor de track size
- De spaties staan voor de grid lines
- **track-size:** auto, px, vh, vw, % of een fraction van de overgebleven ruimte: **fr**
- **line-name:** automatisch positieve en negatieve nummers. Maar je kan die zelf een naam geven om bij een item op te roepen.

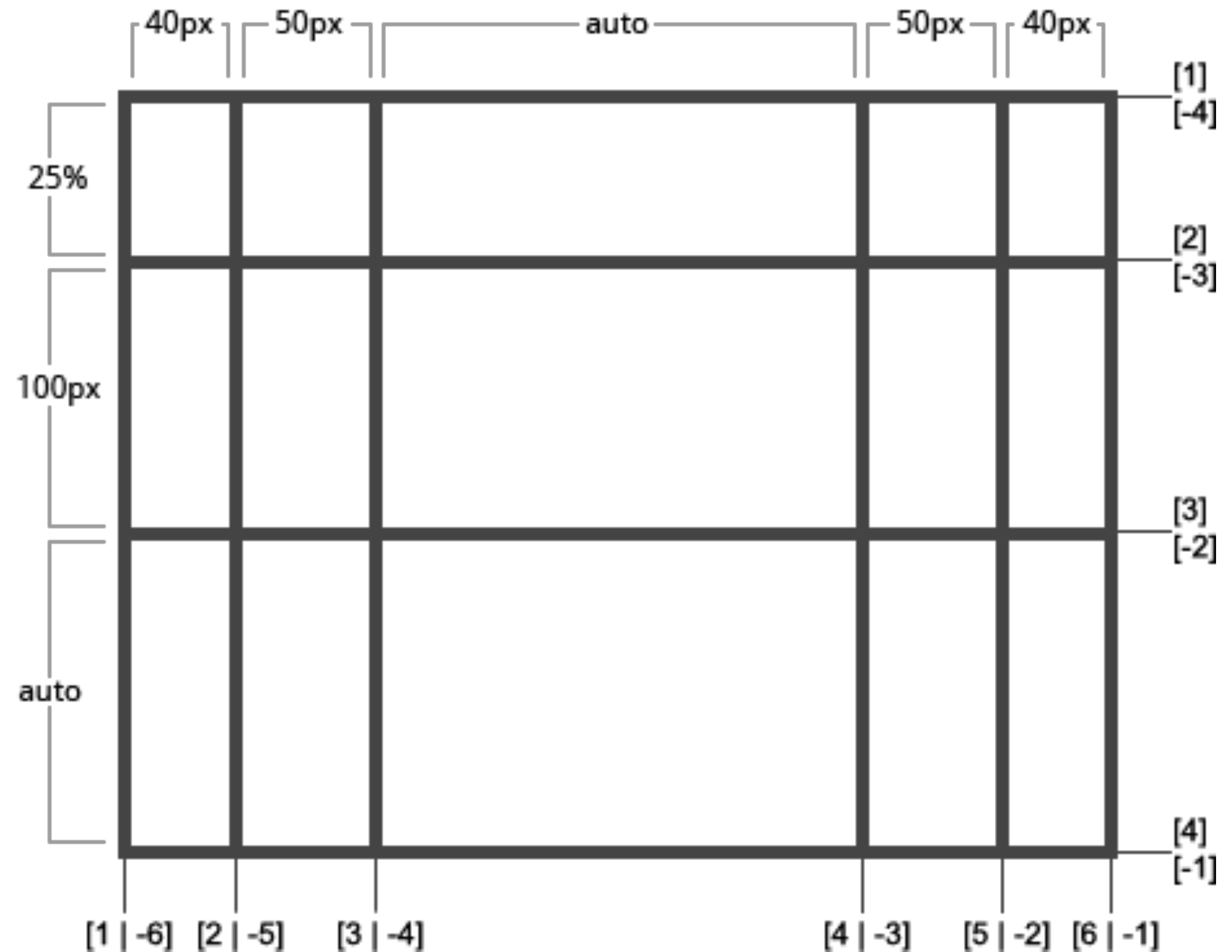
```
.container {  
  grid-template-columns: 4rem 5rem auto 5rem 4rem;  
  grid-template-rows: 25% 10rem auto;  
}
```

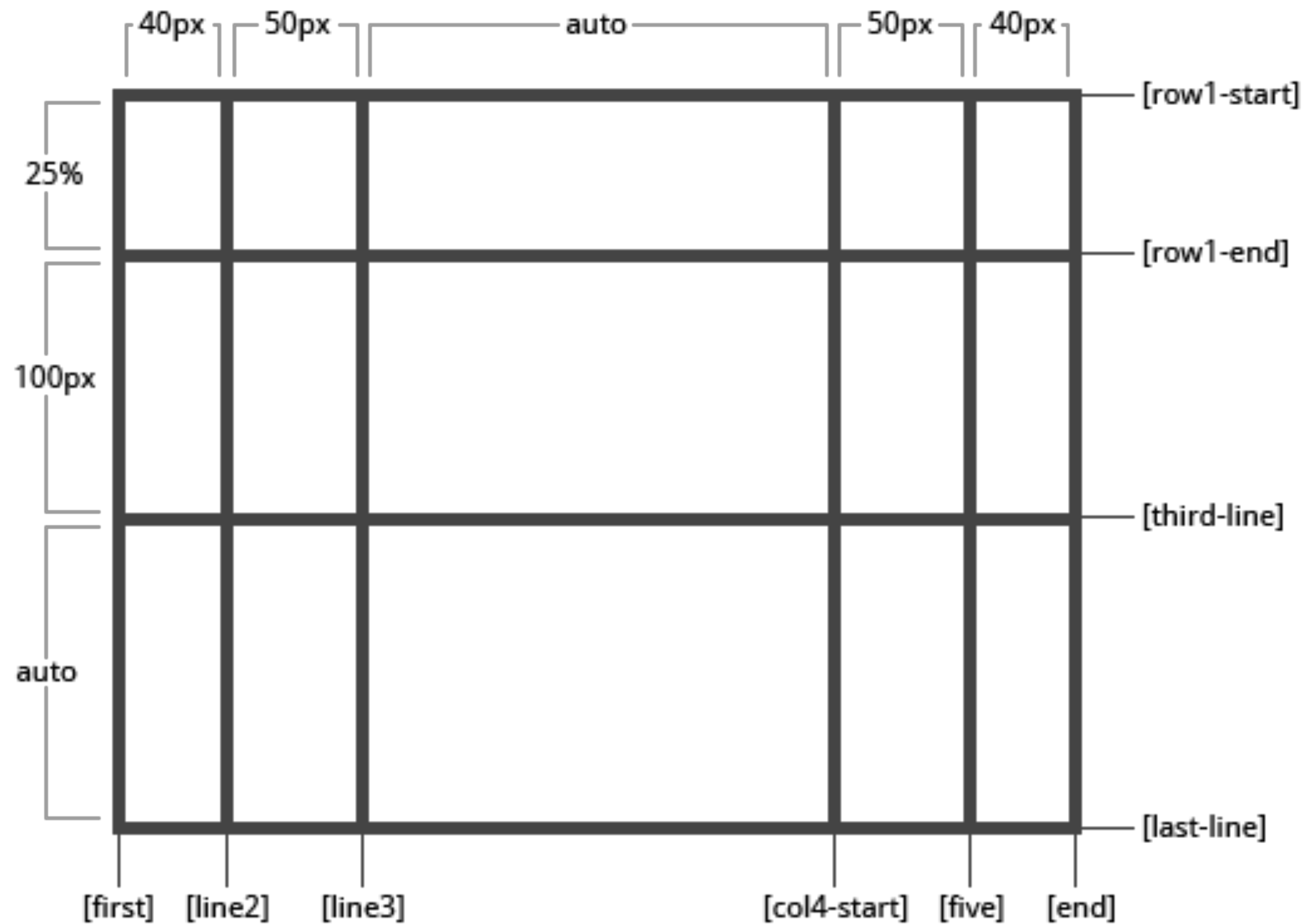


# Automatische line-names

```
grid-template-columns:  
40px 50px auto 50px 40px;
```

```
grid-template-rows:  
25% 100px auto;
```





```
grid-template-columns: [first] 40px [line2] 50px [line3] auto [col4-start] 50px [five] 40px [end];  
grid-template-rows: [row1-start] 25% [row1-end] 100px [third-line] auto [last-line];
```



```
<!DOCTYPE html>
<html lang="en" class>
  <head>...</head>
  <body>
    <div class="c-grid">...</div>
    <script src="https://cpwebassets.codepen.io/assets/stopExecutionOnTimeout-1b93190...js"></script>
    <script src="https://cdpn.io/cp/internal/boombox-key=pen.js-1e24081c-1af4-1d0f-bee7-d39cbd17c709" crossorigin></script>
  </body>
</html>
```

# grid-template-columns

## grid-template-rows

- **Track size special notiations**
  - 1fr: fraction unit: 1 keer de overgebleven ruimte.
  - auto: neemt volledige breedte in die mogelijk is.
  - min-content: kleinst mogelijke grootte voor de content
  - max-content: grootst mogelijke grootte van de content
  - repeat(12, 1fr): maakt 12 kolommen aan van 1 fraction unit.
  - minmax(260px, 1fr): maakt 1 column of row aan van minimum 260px en maximum 1 fraction unit
  - auto-fit en auto-fill keywords: **responsive grid system without media queries!**
  - **Combinaties:** grid-template-columns: minmax(150px, 1fr) auto 1fr;  
Maakt 1 column aan van minimum 150px en maximum 1fr, 1 column die automatisch de breedte van de content krijgt en 1 column die altijd 1 fraction unit is.

**grid-column-start, grid-column-end**  
**grid-row-start, grid-row-end,**  
**grid-column, grid-row**

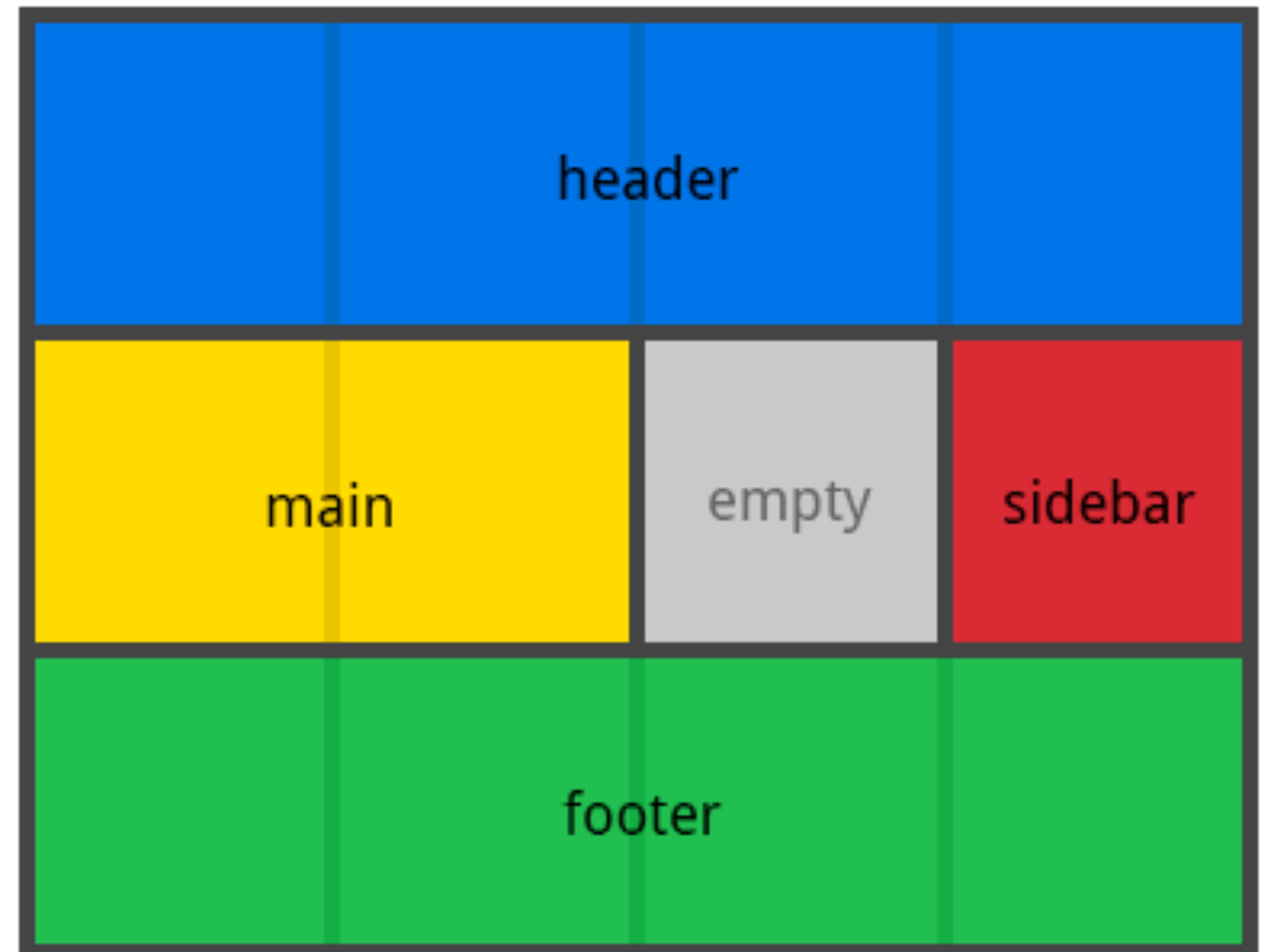
- Bepaalt de locatie van een **grid item** binnen het grid door te refereren naar grid lines
- **Values:**
  - **<line>** positief of negatief nummer of de zelfgekozen grid line name
  - **span <number> of <name>** Hoeveel grid tracks moet het element innemen
- **Bijvoorbeeld:**
  - grid-column-start: 2
  - grid-column-end: span 2;
  - grid-column: 2 / span 2;

# Grid-template-areas

- Laat toe om namen te geven aan grid areas
- Visuele representatie van je grid
- Values
  - **<grid-area-name>** een zelfgekozen naam
  - **.** een lege grid cell
  - **none** geen grid lines

# Grid-template-areas

```
.container {  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```



# Grid-area

- Laat toe om een grid item op een **voorgedefinieerde** grid area te plaatsen.
- (Nog) een kortere manier dus om de volgorde te bepalen

<https://codepen.io/simoncoudeville-nmct/pen/PoVqGrJ>



# Align items in a cell

- **Horizontal**
  - justify-items: for each item in the container
  - justify-self: for one specific item in the container
  - Common values: start, end, center and stretch
- **Vertical \***
  - align-items: for each item in the container
  - align-self: for one specific item in the container
  - Common values: start, end, center and stretch
- **Shorthand**
  - place-items: Shorthand for justify-items and align-items
  - Center 1 item horizontally and vertically with 2 lines of css

# Align content in a grid

- **Horizontal**
  - justify-content: for each item in the container
  - Common values: start, end, center and stretch
- **Vertical \***
  - align-content: for each item in the container
  - Common values: start, end, center and stretch
- **Shorthand**
  - place-content: Shorthand for justify-content and align-content
  - Center multiple items horizontally and vertically with 3 lines of css

\* container needs a height

<https://codepen.io/simoncoudeville-nmct/pen/MWLwbaR>



# Voorbeelden

- <https://gridbyexample.com/patterns/>
- <https://css-tricks.com/snippets/css/css-grid-starter-layouts/>
- <https://css-tricks.com/things-ive-learned-css-grid-layout/>
- Image gallery met CSS grid: <https://codepen.io/zuca/pen/wmzBrx>
- Heel mooi voorbeeld van verschillende technieken: <https://codepen.io/mxbck/pen/ERNwBy?editors=0100>

# Zelfstudie

- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://dev.to/perborgen/want-to-learn-css-grid-here-s-my-free-full-length-course-31ln>
- <https://css-tricks.com/look-ma-no-media-queries-responsive-layouts-using-css-grid/>
- Auto-fill vs auto-fit: <https://css-tricks.com/auto-sizing-columns-css-grid-auto-fill-vs-auto-fit/>
- Minmax function: <https://bitsofco.de/how-the-minmax-function-works/>