

# **Development 1 : Labo CSS Grid**

studiegebied HWB

bachelor in het **toegepaste Informatica**

campus **Kortrijk**

academiejaar **2022-2023**



## Inhoudsopgave

Inhoudsopgave.....	Fout! Bladwijzer niet gedefinieerd.
<b>1</b>	<b>Inleiding.....3</b>
1.1	Verslag .....3
<b>2</b>	<b>Labo.....4</b>
2.1	Opdracht 1 .....4
2.2	Opdracht 2 .....4
2.3	Opdracht 3 .....5
2.4	Opdracht 4 .....5
2.5	Opdracht 5 .....5
2.6	Opdracht 6 .....6
2.7	Extra oefening.....7

# 1 Inleiding

Aan de hand van Grid kunnen we op een eenvoudige manier complexe design layouts ontwerpen die toch responsive zijn. Dit onderwerp wordt **niet** behandeld in de CSS cursus en wordt ondersteund door extra materiaal door middel van artikels en filmpjes.

## 1.1 Verslag

In dit deel van de cursus staan verschillende vragen die je moet beantwoorden en opdrachten om iets te maken of uit te proberen. Het is belangrijk dat je alle opdrachten zorgvuldig uitvoert!

Documenteer je werk in een verslag document (pdf of docx) “**verslag CSS deel 7**” waarin je:

- Voor elke uitprobeeropdracht een entry maakt met screenshots ter staving van wat je deed.
- Je antwoorden op de gestelde vragen neerschrijft.

Oplossingen van ‘grotere’ opdrachten (met veel code) bewaar je in een Webstorm project “**verslag CSS deel 7**”. Per opdracht maak je in dit project een aparte folder waarin je de bestanden (en subfolders) plaatst.

## 2 Labo

Het is je waarschijnlijk al opgevallen dat, als je complexe layouts wil maken met floating elements, je soms wel extra geneste blokken nodig hebt om het design te laten werken. Wanneer een element vast zit in de structuur van de DOM, kan het enkel maar via JavaScript een nieuwe positie krijgen. Bijvoorbeeld: Het is niet eenvoudig om de footer bovenaan te plaatsen wanneer je scherm minder dan 700px breed is.

Door middel van CSS Grid kunnen we veel vrijer omspringen met de positie van elementen. Bij Grid definiëren we eerst de grid structuur van onze layout, het aantal rijen en het aantal kolommen. Daarna kunnen we aangeven welk blok er op welke positie moet komen in de Grid en hoeveel grid blokjes er ingenomen moeten worden.

In de marge: CSS Grid is niet hetzelfde als Flexbox

Via de Developer Console van Chrome kan je eenvoudig de structuur van een Grid debuggen.



- [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://www.youtube.com/watch?v=AqwPrR7hklE>
- <https://www.youtube.com/watch?v=7kVeCgQCxIk>

### 2.1 Opdracht 1

- Open de Grid exercises en open exercise 1.
- Pas het document `exercise1.css` aan zodat de negen blokken in een Grid met 3 kolommen getoond worden die even breed zijn.
- Hoeveel rijen worden er gegenereerd? Hoe komt dit?
- Pas enkel de css file `exercise1.css` aan.

### 2.2 Opdracht 2

- Open de Grid exercises en open exercise 2.
- Pas het document `exercise2.css` aan zodat de negen blokken automatisch in een Grid van 4 x 3 getoond worden.
- Hoe worden de blokken nu getoond?
- Pas enkel de css file `exercise2.css` aan.

## 2.3 Opdracht 3

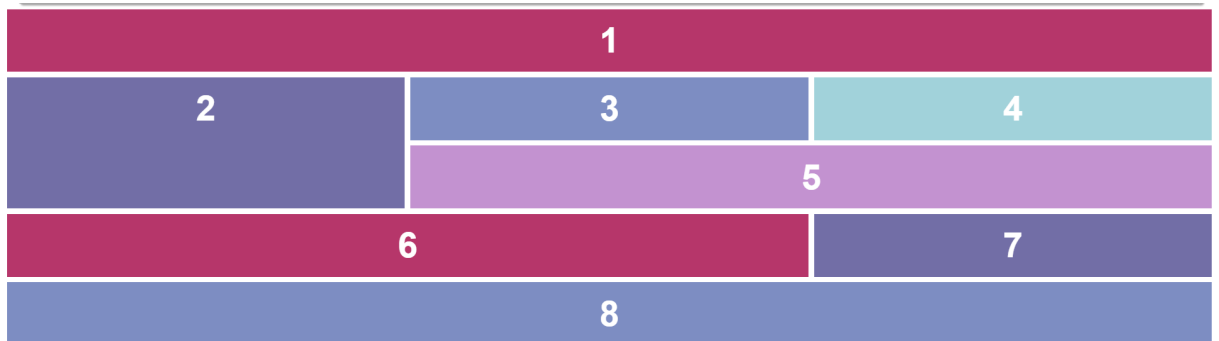
- Open de Grid exercises en open exercise 3.
- Pas het document exercise3.css aan zodat er een 5x5 Grid gemaakt wordt.
- Zorg ervoor dat de blokken in een kruis weergegeven worden waarbij 4 blokken in elke hoek, één blok in het midden en de andere 4 blokken het kruis vervolledigen.
- Kies zelf welke blokken je op welke positie zet.
- Pas enkel de CSS-file exercise3.css aan.

## 2.4 Opdracht 4

- Open de Grid exercises en open exercise 4.
- Start vanaf de oplossing van Opdracht 3.
- Op groot scherm moeten de blokken in een 5 x 5 kruis staan. Wanneer het scherm kleiner wordt dan 700px moeten de blokken in een 3 x 3 grid getoond worden.
- Pas enkel de css file exercise4.css aan.

## 2.5 Opdracht 5

- Open de Grid exercises en open exercise 5.
- Maak zelf een Grid zoals onderstaande afbeelding.

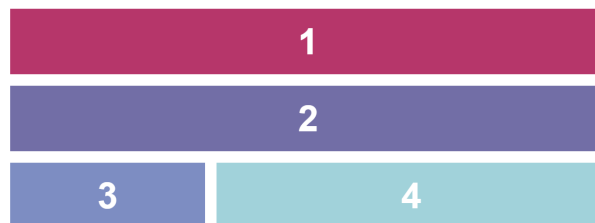


- Positioneer elk element in de juiste cellen en zorg voor de juiste span.
- Geef de blokken geen margins, maar zoek een andere manier om de witruimte tussen de blokken te krijgen.
- Kies zelf een gepaste achtergrondkleur voor ieder blok, zodat je de witruimte goed kan zien.
- Pas enkel de css file exercise5.css aan.

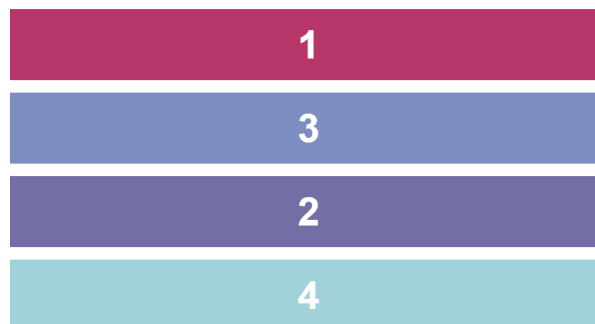
## 2.6 Opdracht 6

- Open de Grid exercises en open exercise 6.
- Maak een website met Grid die zich aanpast aan verschillende groottes zoals op onderstaande afbeeldingen.
- Let op de positie van de verschillende blokken op iedere schermgrootte.
- Maak geen gebruik van JavaScript.
- Pas enkel de css file `exercise6.css` aan.

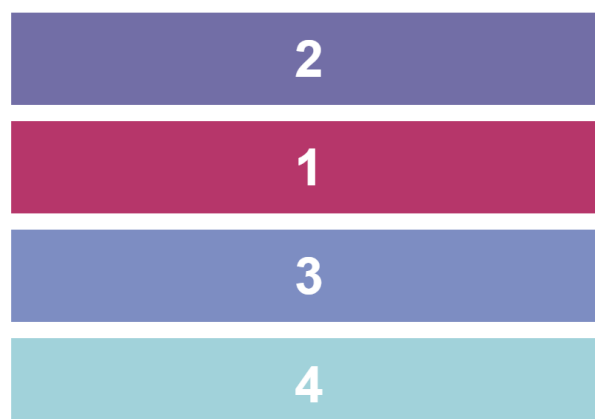
**Breder dan 992 pixels.** Blok 4 is dubbel zo groot als blok 3. Gebruik hiervoor *fractional units*.



**Breder dan 772 pixels.** Let op de volgorde van de blokken.



**Breder dan 320 pixels.** Let op de volgorde van de blokken.



## 2.7 Extra oefening

Open de oefening `Beginsituatie.zip`. Gebruik CSS Grid om alle blokken in een matrix van 3x3 voor te stellen waarbij de middelste kolom dubbel zo breed is als de eerste en laatste kolom. Zorg ervoor dat het volgende patroon verschijnt als je scherm kleiner wordt dan 760px. Let op de volgorde van de getallen!

