

# Programmeren P1

HTML CSS

## Start CSS

Background, Border & Units



WEEK 4



---

# Leerdoelen deze les

- Ik kan een afbeelding toevoegen als achtergrond
- Ik kan verschillende typen borders maken
- Ik kan het verschil uitleggen tussen px, rem en em
- Ik weet wanneer ik vh en vw gebruik

# CSS Borders

- Om elk element in HTML kun je een border plaatsen.
- Een border is een lijn dat om het element zit.
- Deze kun je een bepaalde lijndikte geven, kleur, style en radius.

I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

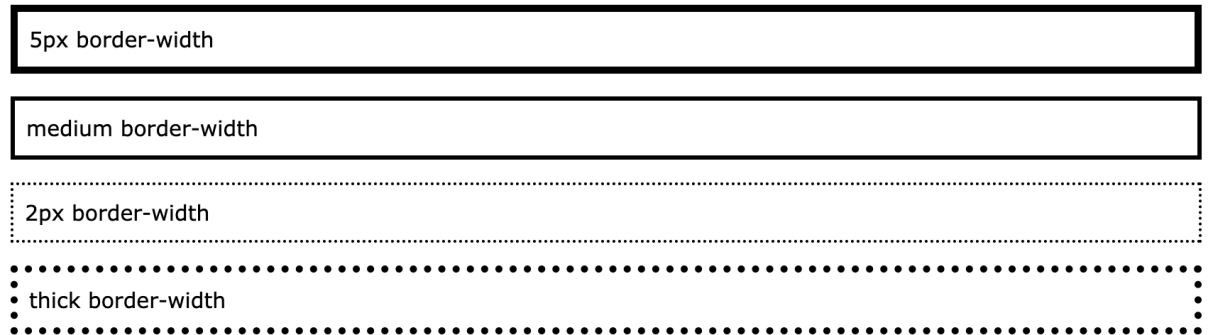
# Border-style

Om een element een border te geven, geef je eerst aan wat voor lijn je wilt hebben. Dit geef je aan met de property: **border-style**

Je kunt een hele lijst aan values kiezen, maar de meest gebruikten zijn: solid, dashed en dotted

```
p {  
  border-style: solid;  
}
```

Solid houdt in dat je een solide lijn trekt, oftewel een normale niet onderbroken lijn.




---

# Border-width

Naast de style, moet je ook aangeven hoe dik je de lijn wilt hebben.  
Dit doe je met de property: **border-width**

```
p {  
    border-width: 5px;  
}
```



# Border-color

En als laatste geef je aan welke kleur de lijn moet zijn.

Dit doe je met de property: **border-color**

```
p {  
  border-color: black;  
}
```

# Border

Alles bij elkaar zijn het drie regels code om één lijntje om een element te trekken. Dat kan ook korter →



```
1
2 p {
3     border-style: solid;
4     border-width: 5px;
5     border-color: black;
6 }
7
8 p {
9     border: 5px solid black;
10 }
```




```
1 p {  
2   border: 5px solid black;  
3   border-radius: 5px;  
4 }
```

This property is used to add rounded borders to an element

## Border-radius

Je kunt de randen van een border ook rond maken. Hiervoor gebruik je de property: **border-radius**





# Border-sides

Uiteraard is het ook mogelijk om alleen aan de linker, rechter, boven of onderkant een lijn te plaatsen. Hoe je dit moet doen kun je vinden op: [Border-sides](#)

---

# CSS Backgrounds

Je kunt in CSS je achtergrond een kleur geven. Hiervoor gebruik je de property: **background-color**:

```
body {  
  background-color: lightblue;  
}
```

Je kunt in CSS op je achtergrond een afbeelding plaatsen. Hiervoor gebruik je de property: **background-image**.

De code ziet er dan als volgt uit:

```
body {  
  background-image: url("hacker.jpg");  
}
```

In de url(" ") plaats de naam van je afbeelding



# Background-repeat

Bij default staat de property `background-image` op `repeat`. Dit betekent wanneer jij een kleine afbeelding hebt als achtergrond, deze zich herhaalt zodat het hele scherm wordt gevuld.

Wil je dit niet (logisch!) dan gebruik je de property: **`background-repeat`**

```
body {  
  background-repeat: no-repeat;  
}
```

# Background-size

De property `background-repeat` zorgt er niet voor dat heel het scherm gevuld is met één afbeelding.

In het vorige voorbeeld hebben we verwijderd dat de afbeelding zich herhaalt, maar nu ziet het scherm eruit als de afbeelding hiernaast.

Om ervoor te zorgen dat de afbeelding het hele scherm vult, gebruik je de property: **`background-size`**

```
body {  
  background-size: cover;  
}
```





---

# Background-position

Als laatst zorgen we er nog voor dat de afbeelding gecentreerd staat, zodat je niet alleen het bovenste gedeelte van de afbeelding ziet maar echt vanuit het midden.

Hiervoor gebruik je de property: **background-position**

```
body {  
  background-position: center;  
}
```





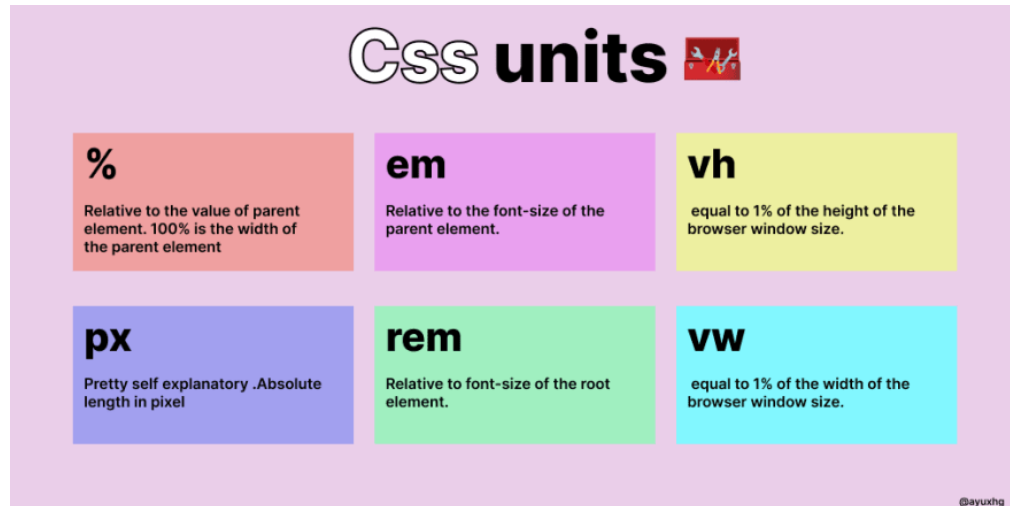
# Background width/height

De uitleg hierboven valt in het niets, als je het element waar je afbeelding in zit geen hoogte en breedte geeft.

Zolang jouw browser niet weet hoe groot het element is, zal hij niets laten zien.

Welke unit je moet gebruiken om de hoogte en breedte aan te geven wordt in de volgende slides uitgelegd.

# CSS Units



In de afgelopen weken heb je al een klein beetje met CSS gewerkt. Dan heb je vast gezien dat overal px (dat staat voor pixels) wordt gebruikt om aan te geven hoe groot iets moet zijn. Dit noemen we: **units**

Naast px hebben we ook de units:

- rem
- em
- vw
- vh



# Units: vw & vh

Wanneer je een element een breedte en een hoogte wilt geven, gebruiken we de vh & vw units.

Vh = view height

Vw = view width

1 view height (vh) staat gelijk aan 1% van de **hoogte** van je browserscherm

1 view width (vw) staat gelijk aan 1% van de **breedte** van je browserscherm

Je gebruikt geen % omdat deze kijkt naar de parent waar je element in zit.

De units vh en vw kijken naar je browserscherm. Dit zorgt ervoor dat het element responsive is, oftewel mee beweegt met het scherm als deze groter of kleiner wordt.

Elk scherm heeft namelijk een ander formaat, gebruik daarom altijd vh en vw!




---

## Units: vw & vh (voorbeeld)

Moet jouw element een breedte krijgen van het hele scherm en de hoogte van de helft van jouw scherm dan ziet de code er als volgt uit →

*Alle breedtes en hoogtes krijgen de units vh & vw*



```
1  section {  
2    width: 100vw;  
3    height: 50vh;  
4  }
```

---

## Unit: px

Px blijven altijd dezelfde grootte,  
wat er ook met je scherm gebeurt.

16px blijft altijd 16px.

**EM** VS **REM** VS **PX**

# Unit: em

Em is relatief aan de **parent** waar je element in zit. Heeft jouw parent een grootte van 16px dan staat 1em gelijk aan 16px.

Is jouw parent 10px dan is 1em gelijk aan 10px, 2em is dan gelijk aan 20px etc.

Als je kijkt in je browser zie je dat de font-size van je <p> aangeeft hoeveel px het is.

```
1 <section>
2   <p>Dit is tekst</p>
3 </section>
```

```
1 section {
2   font-size: 10px;
3 }
4 p {
5   font-size: 1em;
6 }
```

|                       |        |
|-----------------------|--------|
| ▶ display             | block  |
| ▶ font-size           | 10px   |
| height                | 11.5px |
| ▶ margin-block-end    | 10px   |
| ▶ margin-block-start  | 10px   |
| ▶ margin-inline-end   | 0px    |
| ▶ margin-inline-start | 0px    |
| width                 | 786px  |



```
1 p {  
2   font-size: 1rem;  
3 }
```

|                       |        |
|-----------------------|--------|
| ▶ display             | block  |
| ▶ font-size           | 16px   |
| height                | 18.5px |
| ▶ margin-block-end    | 16px   |
| ▶ margin-block-start  | 16px   |
| ▶ margin-inline-end   | 0px    |
| ▶ margin-inline-start | 0px    |
| width                 | 786px  |

## Unit: rem

Rem: rem staat gelijk aan de root. De root is de browser die je gebruikt. Als default heeft jouw browser 16px.

Stel dat iemand de font size in zijn browser (bijvoorbeeld slechtziend) aanpast, dan past de font zich aan en wordt ook groter.

Dit zorgt ervoor dat je website responsive wordt.

Tip: rem wordt het meest gebruikt

Tip 2: geen px meer gebruiken ;)

---

# Tip van de dag!

Geef jouw html een font-size van 62.5% → dan staat 1rem gelijk aan 10px. Zo kun je veel makkelijker rekenen dan met 16px.

Graag gedaan!



```
1  html{  
2      font-size: 62.5%;  
3  }
```



# Opdracht

Voor deze les maak je de opdrachten:

- *Opdracht 4.2 – Background, Font & Borders*
- *Opdracht 4.3 – Uitbreiding opdracht 3.1*

Uiterlijk inleveren: zie Canvas