

**Informatietechnologie 1**

**Cascading Style Sheets**

**Het box model**

**Kristof Michiels**

# In deze presentatie

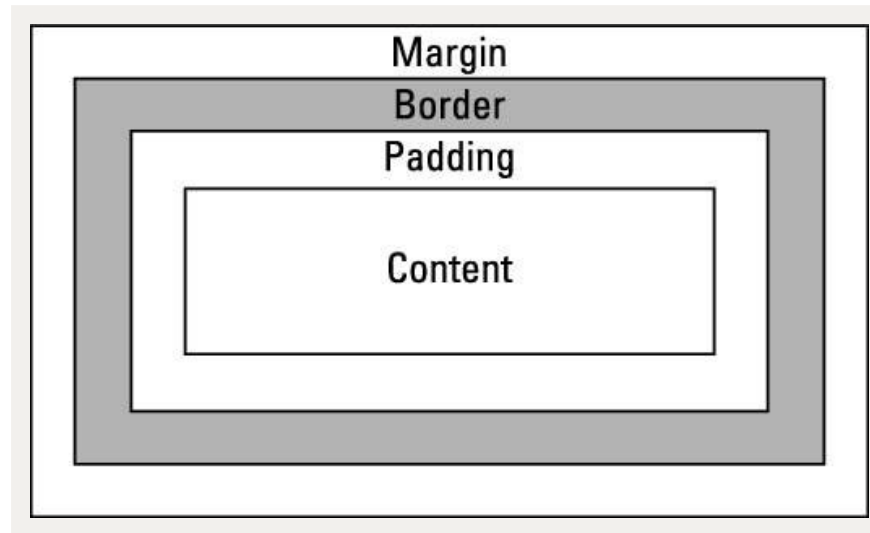
- De delen van een element box
- De afmetingen van een box
- Padding
- Borders
- Outlines
- Margins
- Display rollen
- Schaduw toevoegen

# Het box model: één van de fundamentele concepten van CSS

# Box model

- Elk element in een document genereert een rechthoekige element box
- Is het geval voor zowel block elementen als voor inline elementen
- Je kan er eigenschappen zoals width, height, padding, border en margin op gaan zetten

# De box



# Onderdelen van de box

- Content:
  - De informatie die zich in het element bevindt
- Padding:
  - Dit is het gebied tussen de content en de optionele border. Is optioneel.
- Border:
  - Dit is een lijn die het element en zijn padding omhult. Is optioneel.
- Margin:
  - Is een optionele hoeveelheid ruimte die kan toegevoegd worden aan de buitenzijde van de border van een element

# Afmetingen van de box

- Standaard worden breedte en hoogte van een element automatisch door de browser uitgerekend
  - De box zal zo breed zijn als het moederelement ruimte biedt...
  - ... en zo hoog als de informatie die ze bevat nodig heeft (box zonder content heeft als hoogte 0)
  - Met width en height kan je dit evenwel vanuit CSS aanpassen!
- Een specifieke breedte aangeven doe je met width
  - Waarden: lengte eenheid | percentage | auto (standaardwaarde)
- Een specifieke hoogte aangeven doe je met height
  - Waarden: lengte eenheid | percentage | auto (standaardwaarde)

# Afmetingen van de box

- Standaard is de totale visuele breedte van een element de som van width + padding + border
- Standaard is de totale visuele hoogte van een element de som van height + padding + border
- Voor de totale element breedte+hoogte kan je uiteraard ook de margin meerekenen

```
main {  
  width: 100rem;  
  height: 50rem;  
  border: 0.1rem solid black;  
  padding: 1rem;  
  margin: 2rem;  
  /* Totale visuele breedte: 0.1rem+1rem+100rem+1rem+0.1rem = 102.2rem */  
  /* Totale visuele hoogte = 0.1rem+1rem+50rem+1rem+0.1rem = 52.2rem */  
}
```



# Box-sizing

- Met box-sizing kan je ervoor kiezen om padding + border deel te laten uitmaken van de width en de height. Width en height staan dan effectief voor de totale visuele breedte en hoogte van het element
- Waarden: content-box (standaard) | border-box

```
main {  
  width: 100rem;  
  height: 50rem;  
  border: 0.1rem solid black;  
  padding: 1rem;  
  box-sizing: border-box;  
  /* hierdoor totale breedte = 100rem, totale hoogte = 50rem */  
}
```

# Box-sizing

- Het is eigenlijk aan te bevelen voor elk element box-sizing: border-box; te gebruiken. Daarom gaan we het in onze oefeningen toekennen op onderstaande manier
- Op deze manier zet je met width de volledige visuele breedte van je element. Eventuele border en padding worden dan afgetrokken van de width
- Handig wanneer je bvb de width in % of vw wil weergeven...

```
html {  
    box-sizing: border-box;}  
  
*, *:before, *:after {  
    box-sizing: inherit;}
```

# Block vs Inline

- Je kan width en height enkel speciëren voor block elementen en geen tekst bevattende inline elementen zoals afbeeldingen
- De width en height eigenschappen voor inline tekstelementen worden door de browser genegeerd
- Je kan dus geen width en height zetten op een a of em element

# Hoogte weergeven

- *Height* zetten werkt op dezelfde manier als *width*
- Je gaat het in de praktijk wel minder vaak zetten
- Je laat de browser dan de *height* automatisch bepalen als gevolg van de content binnen dit element
- Als je toch een height zet: weet dat het meer content kan bevatten dan jouw *height*
- Je kan in dat geval remediëren met de *overflow* eigenschap (zie verder)

# Maximum- en minimumafmetingen

- Als je limieten wil zetten op block elementen, gebruik dan max- en min- width of height
- Eigenschap: max-width, min-width, max-height, min-height
- Waarden: lengte eenheid | percentage | none

# Maximum- en minimumafmetingen

HTML


CSS

Result

EDIT ON  
CODEPEN

```
1▼ :root {font-size: 62.5%;}  
2  
3▼ body {  
4     background-color: firebrick;  
5 }  
6  
7▼ main {  
8     background-color: white;  
9     box-sizing: border-box;  
10    width: 80vw;  
11    height: 80vh;  
12    margin-left: auto;  
13    margin-right: auto;  
14    margin-top: 10vh;  
15    max-width: 80rem;  
16    min-width: 40rem;  
17 }
```

LIVE



Resources 1x 0.5x 0.25x Rerun

<https://codepen.io/admkrm/pen/aPbqax>

# Overflow

- Wanneer de afmetingen van een element te klein zijn voor de content dat het bevat, dan kan je remediëren met de overflow eigenschap
- waarden: visible (default) | hidden | scroll | auto

```
main {  
  width: 60rem;  
  height: 20rem;  
  overflow: hidden;  
}
```

# Padding

- Padding toevoegen aan een element doe je wanneer je een border of een achtergrondkleur gebruikt
- Geeft content ademruimte
- Kan je toevoegen aan zowel block als inline elementen
- Eigenschappen: padding-top, padding-right, padding-bottom, padding-left
- Waarden: lengte eenheid | percentage

```
div.mijnklasse {  
  padding-top: 3rem;  
  padding-right: 4rem;  
  padding-bottom: 3rem;  
  padding-left: 4rem;  
  background-color: #eee;  
}
```



# Padding: de verkorte notatie

- Je kan ook de verkorte notatie gebruiken voor padding
- Eigenschap: padding
- Als waarden kan je 4, 3, 2 of 1 waarden meegeven
- Bij 4 waarden is de volgorde: bovenaan rechts onderaan links
- Het voorbeeld van daarnet wordt dan:

```
div.mijnklasse {  
  padding: 3rem 4rem 3rem 4rem;  
  background-color: #eee;  
}
```

# Padding: de verkorte notatie

- Bij 3 waarden: bovenaan rechts/links onderaan
- Bij 2 waarden: bovenaan/onderaan links/rechts
- Bij 1 waarde: bovenaan/rechts/onderaan/links

```
div.mijnklasse {  
  padding: 3rem 4rem 3rem;}
```

```
div.andereklasse {  
  padding: 3rem 4rem}
```

```
div.nogandereklasse {  
  padding: 3rem;}
```

# Borders

- Een border is een lijn rond een content element + zijn eventuele padding
- Borders kunnen rond het gehele element worden aangebracht, of aan slechts een enkele zijde
- Je kan ze uitgebreid of verkort weergeven
- Bestaat uit 3 componenten: border-width border-style border-color

```
footer {  
  border-top: 0.1rem solid black;  
  border-right: 0.1rem solid black;  
  border-bottom: 0.1rem solid black;  
  border-left: 0.1rem solid black;}
```

```
footer {  
  border: 0.1rem solid black;}
```

# Border-style / border-width / border-color

- Mogelijke waarden border-style: none (standaard) | solid | hidden | dotted | dashed | double | groove | ridge | inset | outset
- Mogelijke waarden voor border-width: lengte eenheden
- Mogelijke waarden voor border-color: alle kleuren en hun notaties (inclusief transparent)
- Je kan ook verwijzen naar border-top-style, border-bottom-width, border-left-color, ...

# Afgeronde randen met border-radius

- Je kan de randen van de border afronden met border-radius
- Waarden: lengte eenheden | percentages

```
footer {  
  border: 0.1rem solid black;  
  border-radius: 1rem;}
```

```
footer {  
  border-top-left-radius: 10%;  
  border-top-right-radius: 10%;  
  border-bottom-right-radius: 10%;  
  border-bottom-left-radius: 10%;}
```

# Margin

- Een margin is een optionele hoeveelheid ruimte die je kan toevoegen langs de buitenzijde van de border
- Margins zorgen voor afstand tussen elementen onderling of met de buitenrand van de browser
- De eigenschap margin en verkorte versie werken op dezelfde manier als bij padding (met 4,3,2,1 waarden)
- Waarden: lengte eenheden | percentages | auto

```
div.mijnklasse {  
  margin-top: 3rem;  
  margin-right: 4rem;  
  margin-bottom: 3rem;  
  margin-left: 4rem;}  
  
main {  
  margin: 3rem 4rem 3rem 4rem;}
```

# Margin: %, auto en op body

- Werk je met een percentage dan wordt dit percentage berekend op de beschikbare breedte van het moederelement
- Gebruik je auto voor margin-left en margin-right dan gaat de browser de beschikbare ruimte zelf verdelen over beide eigenschappen
- Margin toevoegen aan het body element zorgt voor ruimte tussen de browserrand en de pagina-content

```
div.mijnklasse {  
  margin-top: 3rem;  
  margin-right: auto;  
  margin-bottom: 3rem;  
  margin-left: auto;}  
  
main {  
  margin: 3rem auto 3rem auto;}
```

# Ineenvallen van top en bottom margins

- De top en bottom margins van elementen die onder of boven elkaar staan heffen elkaar gedeeltelijk op
- De grootste waarde blijft over. In onderstaand voorbeeld zal margin 3rem zijn
- Tip: bepaal afstand tussen elementen consequent met margin-bottom op bovenste element

```
<h1>Dit is een titel</h1>  
<h2>Dit is nog een titel</h2>
```

```
h1 {  
  margin-bottom: 3rem;}  
  
h2 {  
  margin-top: 2rem;}
```



# Margins op inline elementen

- Je kan top en bottom margins toevoegen aan inline tekstelementen, maar het zal geen ruimte creëren
- Margin left en right werken hier wel
- Bij niet-tekstuele inline elementen zoals afbeeldingen kan je WEL langs alle kanten margin toevoegen

```
em {  
  margin: 3rem;} /* zal boven en onder geen 3rem toevoegen, enkel links+rechts */  
  
img {  
  margin: 3rem;} /* zal ruimte langs alle zijden toevoegen */
```

# Negatieve margins

- Je kan voor margins ook negatieve waarden gebruiken
- Hiermee beweeg je het element in de omgekeerde richting dan zou gebeurd zijn met een positieve waarde
- Je zal zien dat dit in bepaalde situaties handig is om een element ergens een duwtje te geven zodat het perfect komt te staan waar je het wil

```
p {  
  margin-left: -3rem;}
```

# Display types

- We hebben tot nu toe block en inline elementen gezien. We observeerden ook hun kenmerken/verschillen
- De display eigenschap bepaalt welk soort element box een element in de layout krijgt toegewezen
- Je kan hiermee het element van type doen veranderen
- Er zijn veel display types en ik zal ze je besparen. Ik lijst hier enkel deze op die we zullen gebruiken
- Mogelijke (gebruikte) waarden: inline | block | flex | grid | inline-block | none

```
img, a {display: block;}  
  
p {display: none;}  
  
li {display: inline;}
```

# Display: none

- Met display: none laat je een element helemaal verdwijnen uit de DOM
- Het neemt dan ook geen plaats meer in. De ruimte wordt ingenomen door de elementen die volgen
- Er is een alternatief dat een element onzichtbaar maakt, maar het wel de plaats blijft innemen: visibility

```
p {display: none;}  
  
h2 {visibility: hidden;}  
  
h2:hover {visibility: visible;}
```

# Box drop shadows

- We zagen eerder text-shadow. Onze boxes kunnen ook schaduw krijgen: met box-shadow
- Waarden: 'horizontal offset' 'vertical offset' 'blur distance' 'spread distance' color inset | none (standaard)
- Je kan meerdere schaduwen toevoegen aan hetzelfde element. Je scheidt dan elke schaduwnotatie met een komma
- Je kan ook negatieve waardes kiezen: experimenteer!

```
p {box-shadow: .6rem .6rem black;}  
p {box-shadow: .6rem .6rem .5rem grey;}  
p {box-shadow: .6rem .6rem .5rem 1rem black;} /* met een spread */
```

# Box drop shadows

HTML

CSS

Result

EDIT ON CODEPEN

1▼<div>

2▼<h1>The picture of Dorian Gray</h1>

3▼<p>Dorian Gray threw his hat and coat upon the table and passed into the library. For a quarter of an hour he walked up and down the room, biting his lip and thinking. Then he took down the Blue Book from one of the shelves and began to turn over the leaves. "Alan Campbell, 152, Hertford Street, Mayfair." Yes; that was the man he wanted. At nine o'clock the next morning his servant came in with a cup of chocolate on a tray and opened the shutters. Dorian was sleeping quite peacefully, lying on his right side, with one hand underneath his cheek. He looked like a boy who had been tired out with play, or study.</p>

4

5▼<p>The man had to touch him twice on the shoulder before he woke, and as he opened his eyes a faint smile passed across his lips, as though he had been lost in some delightful dream. Yet he had not dreamed at all. His night had been untroubled by any images of pleasure or of pain. But youth smiles without any reason. It is one of its chiefest charms.</p>

6

7▼<p>He turned round, and leaning upon his elbow, began to

LIVE

1▼:root {font-size: 62.5%;}

2

3▼body {

4margin: 5rem;

5font-family: helvetica;

6font-size: 1.8rem;

7}

8

9▼div {

10border: 0.1rem solid black;

11padding: 5rem;

12box-shadow: .5rem .5rem 0rem black;

13}

14

15▼p {

16margin-bottom: 2rem;

17line-height: 2.2rem;

18}

19

20▼h1 {

21font-weight: 900;

22margin-bottom: 2.5rem;

LIVE

Resources

<https://codepen.io/admkrm/pen/madoRw>

# IT1 - Cascading Style Sheets - Het box model

Vragen?

[kristof.michiels01@ap.be](mailto:kristof.michiels01@ap.be)