

design your future

## Web Development 1 – Labo CSS 2

Auteurs: Yanic Inghelbrecht, Christophe De Waele

handelswetenschappen en bedrijfskunde

bachelor in de toegepaste informatica

campus Kortrijk

academiejaar 2021-2022



katholieke hogeschool  
associatie KU Leuven

# Inhoudsopgave

<b>Inhoudsopgave.....</b>	<b>2</b>
<b>1 Inleiding.....</b>	<b>3</b>
1.1 Verslag .....	3
<b>2 Labo.....</b>	<b>4</b>
2.1 Structuur-gebaseerde selectoren .....	4
2.2 Opdracht 1 .....	4
2.3 De box van een element .....	5
2.4 Voorbeeld background border buiten border .....	6
2.5 Opdracht 2 .....	6
2.6 Opdracht 3 .....	7
2.7 Borders.....	7
2.8 Pseudo (class) selectoren.....	7
2.9 Opdracht 4 .....	7
2.10 Opdracht 5 .....	8
2.11 Hyperlinks en lijsten vormgeven .....	8
2.12 Opdracht 6 .....	8
2.13 Extra .....	9
2.13.1 Meer over achtergrond afbeeldingen .....	9
2.13.2 Browser prefixes .....	9
2.13.3 Box shadow.....	9
2.13.4 Inheritance .....	9

# 1 Inleiding

Deze les behandelt verdere mogelijkheden voor CSS-selectoren en style properties, ook komt inheritance (overerving) aan bod.

## 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 2**" 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 2**". Per opdracht maak je in dit project een aparte folder waarin je de bestanden (en subfolders) plaatst.

## 2 Labo

### 2.1 Structuur-gebaseerde selectoren

Lees Secties 2.2.3 t.e.m. 2.2.5 uit de CSS3 cursus. Hierin wordt uitgelegd hoe je meerdere selectoren kunt combineren om bv. elementen te selecteren op basis van hun relatieve positie in de DOM-tree structuur. Let hier echter mee op want het maakt je CSS-code afhankelijk van de structuur in je HTML-document. Als die structuur wijzigt zul je dus ook je CSS-regels moeten aanpassen.

Bouw dus liefst geen al te lange kettingen die de structuur van je HTML reflecteren, zoals bv.

```
section > section > ul > li > a > img { ... }
```

Eerder zagen we dat er ook hiërarchische selectoren bestaan waarmee we elementen te pakken kunnen krijgen. Om bijvoorbeeld alle hyperlinks in een lijst op onze pagina selecteren:

```
body > article > section > ul > li > a
```

Dit soort selector is echter te zeer gekoppeld aan de huidige structuur van de DOM-tree in die pagina. Zodra de pagina structureel wijzigt moeten we de selector herzien. Je schrijft dan beter een algemenere selector als:

```
ul a2
```

Bovendien zul je zelden alle `<ul>`-elementen op een pagina op dezelfde manier willen behandelen. De menu balk bovenaan de pagina is typisch gedefinieerd met een `<ul>` element maar die zal totaal anders gestyled moeten worden dan een opsommingslijst in de hoofdtekst van de pagina. In zo'n geval kun je beter aan dergelijke `<ul>` elementen aparte classes toekennen en de selector daarop baseren:

- `.menubalk a`
  - Dit selecteert alle hyperlinks in een element met class 'menubalk'.
- `.opsomming a`
  - Dit selecteert alle hyperlink in een element met class 'opsomming'.

Let op, een spatie in een CSS selector betekent echt wel iets: het is de descendant selector!

Descendant selectoren zijn bv. handig om herhaling te vermijden. In plaats van elk individueel `<li>` element in een lijst van een class te voorzien, kun je beter een class toekennen op het bovenliggende `<ul>` element.

### 2.2 Opdracht 1

Onderstaand HTML fragment maakt deel uit van een menubalk op een bedrijfswebsite

```
<ul>
<li class="menuitem"><a href="...">products</a></li>
<li class="menuitem"><a href="...">distributors</a></li>
```

```
<li class="menuitem"><a href="...">contact</a></li>
</ul>
```

De items in de menu balk worden gestyled met:

```
.menuitem {
  display:inline-block;
  width:150px;
}

.menuitem > a {
  color:red;
}
```

Herschrijf dit fragment en de CSS-regels zonder classes op elk `<li>` element te moeten gebruiken.

Merk op dat een HTML-element meerdere classes kan hebben, bv.:

```
class='voorbeeld belangrijk codefragment'
```

In het class attribuut scheid je ze door spaties, maar in een selector schrijf je ze aaneen! Dit kan dus makkelijk tot fouten leiden, bv.:

- `.voorbeeld.belangrijk`
  - Selecteert alle elementen die van zowel class 'voorbeeld' als 'belangrijk' zijn.
- `.voorbeeld .belangrijk`
  - Dus met spatie ertussen, betekent iets heel anders!
  - Selecteert alle elementen van class 'belangrijk' die een descendant zijn van een element met class 'voorbeeld'

Sommige elementen krijgen enkel een class of id toegekend om ze vanuit Javascript te benaderen, deze class of id zul je dan niet in de CSS definities terugvinden. Als je bv. code wil laten uitvoeren als er op een element geklikt wordt, zul je op een of andere manier in je Javascript code precies dat element te pakken moeten krijgen om het klikgedrag eraan te koppelen.

Bijvoorbeeld, deze Javascript opdracht:

```
document.querySelector(".voorbeeld").addEventListener("click", ...);
```

zorgt ervoor dat de `'...'` code uitgevoerd wordt telkens de gebruiker op een element klikt dat de class 'voorbeeld' draagt.

## 2.3 De box van een element

Lees Sectie 7.1 waarin uitgelegd wordt dat in CSS elk element uit de DOM-tree in een box wordt geplaatst.

Als je geen border nodig hebt zou je misschien veronderstellen dat margin en padding op hetzelfde neerkomen, maar bedenk dat de background van het element de padding area overlapt maar niet de margin area. Een margin kan trouwens ook negatief zijn, een padding niet.

In de tekst staat een belangrijk stuk over hoe verticale margins van aangrenzende elementen in elkaar schuiven, dit heet 'collapsing margins'. In het document 'extra uitleg collapsing margins.pdf' kun je nog meer uitleg vinden.

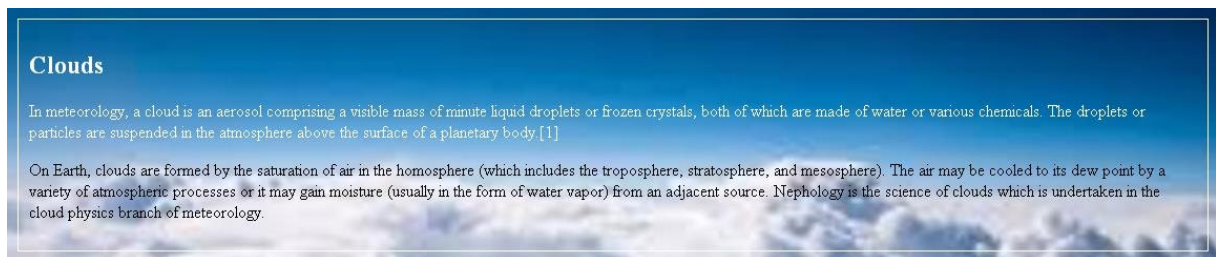
Je kan ook een goed overzicht met verantwoording vinden in onderstaand antwoord op deze stackoverflow vraag: <https://stackoverflow.com/questions/4903549/why-do-the-CSS-width-and-height-properties-not-adjust-for-padding/4903697#49036974>

## 2.4 Voorbeeld background border buiten border

Soms zul je in je HTML extra elementen moeten toevoegen om een bepaald effect te verkrijgen.

Zo'n extra element heeft geen inhoudelijke betekenis maar dient enkel om de visualisatie goed te krijgen. Men gebruikt hier doorgaans een <div>-element voor.

Als je de HTML pagina in 'voorbeeld background buiten border.zip' opent krijg je onderstaande te zien:



We zien dat de titel en twee paragrafen vervat zitten in een rechthoek met een achtergrond en een kader. Merk op dat de background buiten de border doorloopt.

In de bespreking van het box model hebben we echter gezien dat de background van een element **niet** buiten de border kan komen, we kunnen dit effect dus niet realiseren met een background en border op een enkel element.

Bekijk de HTML en CSS code in dit voorbeeld en ga na hoe een extra <div>-element ons probleem oplost.

## 2.5 Opdracht 2

Open de pagina <http://www.CSSzengarden.com/163/> en selecteer in de DOM-tree het element met id="zen-preamble".

Als je op het Styles tabblad naar onder scrollt kun je netjes zien wat de afmetingen zijn van de margin (oranje), border (lichtoranje), padding (groen) en content (blauw) van dit element.

Deze kleuren komen trouwens ook terug in de highlight op de eigenlijke pagina. Merk op hoe vreemd de oranje margin highlight eruit ziet (deze start pas een stukje onder de bovenkant van het element) door de negatieve margin. Probeer negatieve margins te vermijden als je hetzelfde effect op een andere manier kunt bereiken, het maakt het moeilijker om je layout te begrijpen.

## 2.6 Opdracht 3

Download 'opdracht\_resume.zip' en maak de Opdracht.

## 2.7 Borders

Lees Sectie 4.3 over borders.

## 2.8 Pseudo (class) selectoren

Lees Secties 2.4 en 2.5 over pseudo selectoren en pseudo class selectoren.

Bedenk bij de pseudo-selectoren die over de parent-child relatie gaan (bv. nth-child), dat deze niet zo vaak nodig zijn omdat we gewoon een nieuwe class kunnen toekennen. Immers,

- Waarom wil je een element een andere visualisatie geven?
  - Omdat het iets anders is dan de andere elementen!
- Als het iets anders is dan andere elementen?
  - Dan is het op semantisch vlak van een andere categorie.
  - Dus ken gewoon een nieuwe class toe aan dit element!

Als we de HTML broncode kunnen bewerken, kunnen we beter CSS-classes toekennen aan het element en daardoor een eenvoudigere CSS-selector schrijven op basis van die class.

In een dynamische pagina, waarbij HTML elementen aan de DOM-tree worden toegevoegd via Javascript code, kun je dit enkel doen als je de Javascript source code kunt aanpassen. Als je dit niet kunt, zul je je moeten behelpen met ingewikkeldere selectoren.

Merk op dat content toevoegen met bv. `::before` maar één keer zal plaatsvinden per element, dus als je je meerdere CSS-regels hebt die allen content 'before' willen inlassen dan zal er slechts eentje daadwerkelijk uitgevoerd worden. Welke wint, heeft te maken met de voorrangsregels voor CSS-regels, nml. de specificiteit (zie volgend labo) en hun onderlinge volgorde in het geval van eenzelfde specificiteit.

## 2.9 Opdracht 4

Maak zelf een HTML-pagina met minstens een h1 hoofding, een ongenummerde lijst een hyperlink en een paar paragrafen. Koppel daaraan een CSS-bestand dat de volgende vormgeving vastlegt:

- Zet voor het body-element het font op Verdana 1.1 em, de achtergrondkleur op zilverachtig en de letterkleur op donkerblauw.
- Definieer h1 als volgt: het font op kleine hoofdlettertjes (font-variant) en de achtergrondkleur op knalrood.

- Maak het zo dat als de muis de link raakt (`a:hover`), er een kadertje om de link heen komt. Bepaal zelf de styling van het kadertje.
- Maak een klasse met de naam 'stippeltjesrand'. Deze omgeeft een HTML element met een border die gestippeld is (1 px breed). Ken deze stijl toe aan een paar willekeurige elementen in je pagina.

## 2.10 Opdracht 5

Op onze website zouden we graag hyperlinks beter doen opvallen door er een icoontje voor te plaatsen, aan de linkerzijde dus. Bijvoorbeeld:

*In deze regel staat een  [hyperlink](#) met een icoontje ervoor*

Het icoontje mag je zelf kiezen. Om zoveel mogelijk vrijheid te hebben voor de visualisatie geven we voor deze opdracht de voorkeur aan een oplossing gebaseerd op een afbeelding.

Schrijf een versie op basis van een *background image*.

## 2.11 Hyperlinks en lijsten vormgeven

Lees Hoofdstuk 6 over hoe je hyperlinks en links in navigaties menu's kunt stylen.

Lees volgend artikel over hoe je lijsten kunt vormgeven: <https://CSS-tricks.com/almanac/properties/l/list-style/>.

## 2.12 Opdracht 6

We zouden graag een checklist in onze pagina's verwerken. Aangevinkte items in zo'n lijst moeten voorafgegaan worden door een vinkje ☒, andere niet ☐. Merk op dat de gebruiker deze niet zelf kan aanvinken. Bijvoorbeeld:

- ☒ boter
- ☐ kaas
- ☒ eieren

Een checklist is een `<ul>` element met de class "checklist". Aangevinkte items krijgen een class "checked". Niet-aangevinkte items hebben deze class niet.

Bovenstaande voorbeeld ziet er dus in HTML als volgt uit:

```
<ul class="checklist">
  <li class="checked">boter</li>
  <li>kaas</li>
```



```
<li class="checked">eieren</li>
</ul>
```

Schrijf een HTML pagina met bijbehorend CSS bestand om dit te realiseren d.m.v. een icoontje (een afbeelding dus).

## 2.13 Extra

### 2.13.1 Meer over achtergrond afbeeldingen

Lees Sectie 5.1.4 over de *cover* waarde voor achtergrond afbeeldingen en hoe je een achtergrond voor de volledige pagina kunt instellen.

Merk op dat de achtergrond van een element steeds beperkt blijft tot aan de border box, m.a.w. de padding en content area. Als je een achtergrond wil die buiten de border komt, zul je een extra parent element moeten toevoegen. In 'voorbeeld background buiten border.zip' wordt getoond hoe dit kan.

### 2.13.2 Browser prefixes

Lees het begin van Hoofdstuk 4 en Sectie 4.1 over browser prefixes.

### 2.13.3 Box shadow

Lees Sectie 4.3.3 over *box-shadow*, dit effect kan bv. mooi gebruikt wordt bij een afbeelding.

### 2.13.4 Inheritance

Lees Hoofdstuk 3 vanaf het begin t.e.m. Sectie 3.4.4 over inheritance en de volgorde waarin de regels van een of meerdere stylesheets worden toegepast. Qua inheritance is het belangrijk is dat je min of meer weet welke vaakgebruikte properties vanzelf overgeërfd worden, en dat je het bij de andere kunt forceren met de speciale 'inherited' waarde.