

# **Development 1 : Labo 5 CSS – 1**

studiegebied HWB

bachelor in het **toegepaste Informatica**

campus **Kortrijk**

academiejaar **2022–2023**



**Inhoudsopgave**..... Fout! Bladwijzer niet gedefinieerd.

<b>1</b>	<b>Inleiding</b> .....	<b>3</b>
1.1	Verslag .....	3
<b>2</b>	<b>Labo</b> .....	<b>4</b>
2.1	CSS inleiding .....	4
2.2	Opdracht 1 .....	4
2.3	Demonstratie DOM-tree .....	4
2.4	Opdracht 2 .....	5
2.5	Opdracht 3 .....	5
2.6	Opdracht 4 .....	5
2.7	Eenvoudige CSS-Selectoren .....	5
2.8	Opdracht 5 .....	7
2.9	Class en id selectoren .....	8
2.9.1	<i>Voorbeeld</i> .....	8
2.10	Tekst en achtergrond kenmerken.....	10
2.11	Opdracht 6 .....	10
2.12	Opdracht 7 .....	10

# 1 Inleiding

Deze les introduceert de basiswerking van cascading stylesheets aan de hand van enkele eenvoudige selectoren en style properties.

Dit document weidt verder uit over enkele topics uit de CSS3 cursus.

## 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 1**" 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 1**". Per opdracht maak je in dit project een aparte folder waarin je de bestanden (en subfolders) plaatst.

## 2 Labo

### 2.1 CSS inleiding

Lees Hoofdstuk 1 'Inleiding' uit de CSS3 cursus.

Sectie 1.4 over de DOM-tree is ZEER belangrijk voor het vervolg, zorg ervoor dat dat je dit goed begrijpt!

De getoonde boomstructuur in die sectie is gebaseerd op de volgende code:

```
<HTML>
<head>
<title>...</title>
</head>
<body>
<h1>...</h1>
<p>...</p>
<ul>
<li>...</li>
<li>...</li>
<li>...</li>
</ul>
</body>
</HTML>
```

### 2.2 Opdracht 1

Wat zijn de siblings van het `<ul>` element?

Wat zijn de ancestors van het tweede `<li>` element?

Wat zijn de descendants van het `<body>` element?

Hoeveel elementen in een geldig HTML document hebben geen parent?

Heeft elk element één of meerdere child elementen?

### 2.3 Demonstratie DOM-tree

Download en unzip demonstratie dom tree.zip. Open index.HTML in je browser.

Op deze pagina kun je aan de linkerkant een element selecteren en vervolgens op de knoppen klikken om bv. diens ancestors, siblings, descendants e.d. te highlighten in het voorbeeld aan de rechterkant.

## 2.4 Opdracht 2

Open je persoonlijke pagina in Chrome, bekijk de visualisatie van de DOM-tree op het Elements tabblad en vergelijk dit met de HTML code van je pagina.

## 2.5 Opdracht 3

Bekijk de onderstaande pagina's

- <https://www.vives.be>
- <https://www.miras.be>
- <https://www.vrt.be/vrtnws/nl/>

En noteer telkens op welke manier(en) ze de bijbehorende CSS-regels introduceren

- inline style attributen?
- <style> blokken?
- <link> verwijzingen naar aparte CSS-bestanden?
  - Noteer in dit geval ook naar hoeveel CSS bestanden verwezen wordt.



In deze cursus stoppen we alle CSS definities in **aparte bestanden**. We gebruiken geen inline styles en geen <style> blokken!

## 2.6 Opdracht 4

Open twee verschillende designs, rechtsklik en kies bij elk voor 'View page source'. Zet de twee vensters met HTML code naast elkaar op je scherm en vergewis je ervan dat de inhoud bij beiden (bijna) exact dezelfde is.

Welke verschillen zie je?

## 2.7 Eenvoudige CSS-Selectoren

Lees het begin van Hoofdstuk 2 t.e.m. Sectie 2.2.2.

Laat je niet misleiden door de 'element-selector' naamgeving: alle CSS-selectoren selecteren sowieso elementen in de DOM-tree. Daarvoor dienen ze immers: om elementen in de DOM-tree aan te duiden waarop we een bepaalde stijl willen toepassen.

In Secties 2.1 en 2.2.1 leer je een eerste soort CSS-selector kennen: het selecteren op basis van het soort element (i.e. het type). Vandaar dat men dit *element-selectoren* noemt.

In Sectie 2.2.1 worden enkelvoudige selectoren uitgelegd, dit zijn stijlregels die 1 selector hebben, bijvoorbeeld:

```
h1 {  
    ...  
}
```

In Sectie 2.2.2 komen meervoudige selectoren aan bod, deze combineren meerdere enkelvoudige selectoren (gescheiden door komma's) en passen de properties toe op alle elementen die door min. 1 van de enkelvoudige selectoren geselecteerd worden. Bijvoorbeeld:

```
h1, h2 {  
    ...  
}
```

Een meervoudige selector kan gebruikt worden om duplicatie te vermijden door gemeenschappelijke stukken af te splitsen. Bijvoorbeeld, veronderstel dat X en Y hieronder enkelvoudige selectoren zijn:

<pre>X {     property1 = value1;     property2 = value2;     property3 = value3;     property4 = value4; }</pre>	<pre>Y {     property1 = value1;     property2 = value2;     property3 = value3;     property5 = value5; }</pre>
--	--

Dit kun je met een meervoudige selector herleiden tot

<pre>X,Y {     property1 = value1;     property2 = value2;     property3 = value3; }</pre>	
<pre>X {     property4 = value4; }</pre>	<pre>Y {     property5 = value5; }</pre>

Maar doe dit enkel indien de selectoren X en Y betekenis ook iets met elkaar te maken hebben, m.a.w., het is niet omdat twee CSS-regels toevallig eenzelfde 'margin-left:10px' property bevatten dat je dit daarom moet afsplitsen.



Een dienstmededeling die zo belangrijk is dat ze herhaling verdient:

**In deze cursus stoppen we alle CSS definities in aparte bestanden, we gebruiken geen inline styles en geen <style> blokken!<sup>1</sup>**

---

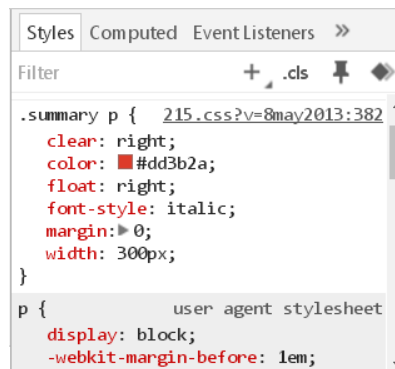
<sup>1</sup> Merk op dat de voorbeelden in de CSS-cursus soms wel een <style> blok gebruiken om de voorbeelden kort te houden.

## 2.8 Opdracht 5

Open de pagina <http://www.CSSzengarden.com/215/>. Rechtsklik op de rode tekst naast de robot en selecteer 'Inspect element'. Het element waarop je klikte zal nu geselecteerd zijn in de DOM-tree weergave van de Chrome Developer tools.

Als je met de muis over de elementen in de DOM-tree weergave hoovert, zullen deze in de eigenlijke pagina aangeduid worden met een highlight.

Rechts op het Elements tabblad zie je informatie over het geselecteerde element, waarvan twee kleinere tabbladen met CSS te maken hebben nml. 'Styles' en 'Computed'.



Figuur 1 Element CSS Styles in Dev Tools

Vink op het Styles tabblad enkele regels af en kijk welke effect dit heeft op de pagina. Probeer ook eens een aantal waarden aan te passen, verander bv. de tekstkleur van rood naar groen.

Merk ook op dat er meerdere CSS-regels op eenzelfde element van toepassing kunnen zijn!

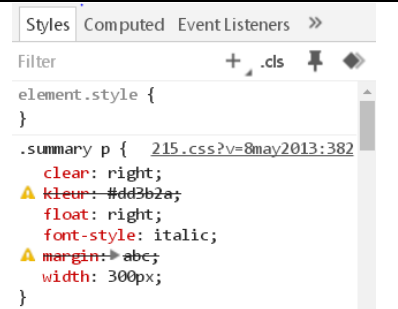
De wat vreemde tekst '215.CSS?v=8may2013:382' naast de eerste stijlregel toont dat deze op regel 382 in het bestand '215.CSS?v=8may2013' staat. Op het Sources tabblad kun je de volledige source code van dit bestand bekijken, alsook de inhoud van alle andere HTML/CSS/JS bronbestanden die werden ingeladen.



**TIP:** als je in je project een CSS bestand hebt toegevoegd dat geen effect lijkt te hebben, kijk dan eerst eens of het in de lijst op het Sources tabblad voorkomt.

Indien **niet**, dan heeft de browser het niet ingeladen. Bijvoorbeeld omdat je een typfout hebt gemaakt in de href waarde van het <link> element. Dit kun je checken op het Network tabblad, waar wellicht een 404 statuscode staat bij de bijbehorende request. Indien er helemaal geen request voor je CSS bestand is verstuurd, heb je wellicht het <link> element vergeten toe te voegen aan je HTML code.

Indien **wel**, dan heb je wellicht een fout gemaakt in je CSS bestand, bv. een verkeerde selector of een typfout in een property naam of value. Je kunt dit zien op het Elements tabblad, bij Styles. Het gele icoontje met het uitroepteken duidt op een fout in de gegeven regel.



## 2.9 Class en id selectoren

Lees Sectie 2.3 over class en id selectoren.

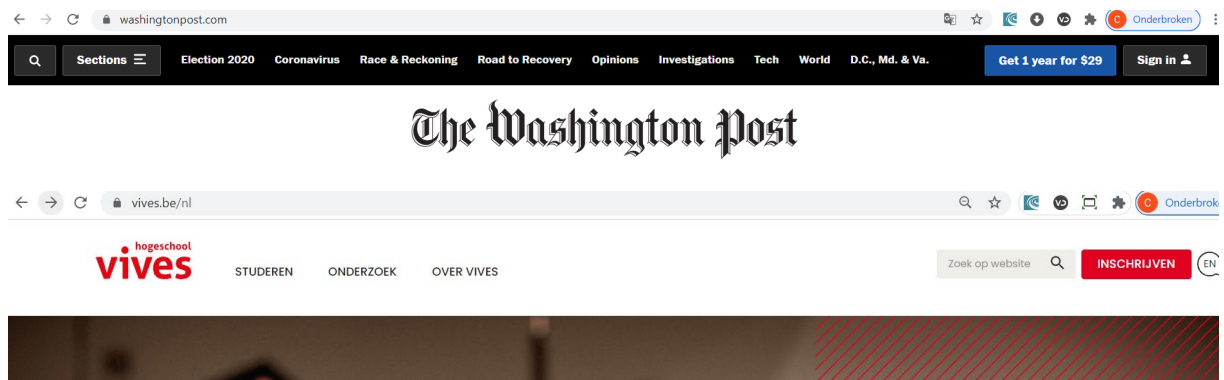
Merk op dat 'class' in CSS weinig te maken heeft met 'class' uit een object-georiënteerde programmeertaal zoals bv. Java. Het enige verband is de achterliggende betekenis van het woord zelf:

***class** - A set or category of things having some property or attribute in common and differentiated from others by kind, type, or quality.*

HTML elementen kunnen via het class attribute in een of meerdere categorieën worden ondergebracht. We geven een element een class attribuut als we het moeten kunnen selecteren om er CSS regels op toe te passen of het via javascript te manipuleren.

### 2.9.1 Voorbeeld

Veel websites hebben bovenaan een menu met links naar specifieke delen van de website. Dit wordt geïllustreerd in de volgende twee figuren:



Indien we een menu bovenaan een pagina willen, ligt het gebruik van een `<ul>` voor de hand want een menu is immers een soort oplijsting. Standaard worden de `<li>` elementen echter met bolletjes getoond, wat er niet bepaald als een menu uit ziet.

We kunnen de visualisatie echter via CSS aanpassen om de bolletjes weg te krijgen en de items naast elkaar te plaatsen. We willen echter niet dat alle `<ul>` elementen op die manier worden getoond, want dan zien eventuele oplijstingen in onze hoofdtekst er niet meer goed uit.



We kunnen onze "menu <ul>" onderscheiden van de gewone <ul> elementen door er een *class* aan toe te kennen:

```
<ul class='menu'>
  <li>...</li>
  <li>...</li>
  <li>...</li>
</ul>
```

In onze CSS file zullen we dan regels opnemen om elementen met de 'menu' class visueel naar onze hand te zetten.

Classes dienen dus om in onze HTML code elementen aan te duiden waarmee we iets willen doen via CSS (of javascript). Als een element noch voor je CSS, noch voor je javascript belangrijk is, moet je er geen class aan toekennen.

Let erop dat je een classname kiest die uitlegt wat het element betekent/voorstelt (semantiek dus) en niet hoe de style er moet gaan uitzien, e.g.:

`class='belangrijk'` is beter dan `class='groteRodeLetters'`

Een goeie uitleg met veel voorbeelden vind je op <https://CSS-tricks.com/semantic-class-names/>.

Je kan aan een element ook meerdere classes toewijzen. Je doet dit door ze op te sommen in de waarde van het class attribuut, gescheiden door spaties, e.g.:

```
<li class="belangrijk menuItem">...</li>
```

(dit <li> element heeft twee classes: 'belangrijk' en 'menuItem')

Je kan gerust eenzelfde class aan meerdere elementen toewijzen! Een pagina kan bijvoorbeeld meerdere <section> elementen bevatten die elk de class 'gallery' toebedeeld kregen.

Het **id attribuut** dient om één enkel element in de DOM-tree op unieke wijze te identificeren. Er mag dus slechts 1 element met een bepaalde id voorkomen in een HTML document.

Stel je hebt in je HTML document een class die slechts aan 1 element wordt toegekend. Moet je dan overschakelen op een id ipv een class? Niet noodzakelijk, het hangt ervan af of er in de toekomst ooit een ander element zou kunnen bijkomen met diezelfde class. En zelfs als dat niet het geval is hoeft je nog niet over te schakelen naar een id: een id dient om een element op unieke wijze te identificeren en komt doorgaans overeen met een externe identificatie wijze. Bv. bij een element dat een specifiek boek beschrijft, zou je het unieke ISBN code als id waarde kunnen gebruiken. Dit is wellicht relevanter voor je javascript code dan voor je CSS definities.

In CSS code wordt doorgaans meer met selectoren gewerkt op basis van classes (of id) dan op basis van het element type:

- Het gekozen element type voor een stukje content kan immers variëren.
  - Initieel zit een titel bijvoorbeeld nog in een <h2> element vervat maar door een herschikking van de pagina veranderen we dit later naar <h3>. Dan willen we natuurlijk liever niet onze CSS regels moeten aanpassen.

- Classes zijn dan als het ware de kapstokken waaraan je de CSS regels ophangt

Voor Javascript code geldt een gelijkaardige redenering.

## 2.10 Tekst en achtergrond kenmerken

Lees Hoofdstuk 5 (behalve 5.1.4) waarin wordt uitgelegd hoe je de tekst en achtergrond van een pagina onderdeel kunt vormgeven.

Zoek elke aangehaalde property op deze site:

<https://www.handleidinghtml.nl/css/eigenschappen/eigenschappen.html>

In Sectie 5.1.2 komt de 'em' afmetingseenheid aan bod. Deze eenheid is vooral handig om de afmetingen van andere elementen af te stemmen op de gebruikte tekstgrootte. Indien de gebruiker de tekstgrootte van zijn browser vergroot (bv. slechtienden) dan groeien deze elementen mee.

In Sectie 5.1.3 zijn vooral de background, -image, -position, -repeat properties van belang, die worden vaak gebruikt. Bekijk zeker ook de 'gewone' color property, om de tekstkleur in te stellen.

## 2.11 Opdracht 6

Maak een webpagina met hoofdingen van drie niveaus: <h1>, <h2> en <h3>. Plaats onder elke hoofding een paragraaf met wat inhoud (de precieze tekst is niet belangrijk).

Koppel aan deze pagina een CSS bestand dat voor de volgende visualisatie zorgt:

- Alle hoofdingen gebruiken het 'Arial' font. De tekstgrootte is resp. 1.8em, 1.5em en 1.4em.
- Alle paragrafen gebruiken het 'Times New Roman' font en hebben een tekstgrootte van 1.2em.
- Zorg ervoor dat de drie paragrafen elk een andere tekstkleur krijgen: rood, groen en blauw. Je zult hiervoor classes moeten gebruiken.

## 2.12 Opdracht 7

In een stylesheet staat het volgende beschreven:

- De fontgrootte van hoofding h1 staat op 2ex
- De fontgrootte van hoofding h2 staat op 2em.

Welk van beide hoofdingen wordt volgens de CSS standaard het grootst weergegeven en waarom?