

Interface Design Lab 1 - Web

Inleiding tot CSS

Richtlijnen voor opmaak

Kristof Michiels

In deze presentatie

- Voordelen en kracht van CSS
- Hoe schrijf je CSS stijlregels?
- Toevoegen aan HTML documenten
- Basisconcepten: overerfbaarheid, specificiteit, de cascade, volgorde van regels, het box model
- Meeteenheden
- Demo's

De voordelen en kracht van CSS

CSS in een notendop

- CSS staat voor cascading style sheets
- Wij gebruiken versie 3
- CSS is een afzonderlijke taal, met een geheel eigen syntax
- Voorziet precisiecontrole over tekstzetting en layout
- Eén enkele stylesheet kan de vormgeving van een gehele site bepalen: eenduidig en efficiënt!
- Alle vormgeving bij CSS laat focussen op structuur en betekenis toe in HTML: toegankelijkheid!

Geschiedenis van CSS

- HTML is een stuk ouder dan CSS
- Eerste versie van CSS in 1996
- Eigenschappen voor kleur, lettertypes, ruimte tussen pagina elementen
- Gebrek aan voldoende ondersteuning in de browsers: heel moeilijk bruikbaar
- Periode van "elke browser een eigen versie HTML/CSS"

Geschiedenis van CSS

- CSS2 kwam er in 1998
- Ondersteuning voor pagina layouts
- Ondersteuning voor andere pagina-types: print, handhelds,...
- CSS 2.1 werd een officiële aanbeveling in 2011

Geschiedenis van CSS

- CSS3 is de versie die we vandaag gebruiken
- Eerste documenten stammen uit 1999 / nu samen met HTML5 iets minder dan 10 jaar echt in gebruik
- Opgedeeld in verschillende modules: animatie, multiple column layouts, borders,...
- Bepaalde modules gestandaardiseerd, anderen nog steeds experimenteel en in ontwikkeling
- Modules hebben een level-nummer: hoe lager hoe experimenteler
- Grid Layout module: starten op level 1
- Enorm veel modules nog in ontwikkeling. Je kan dit zelf volgen (zie 3 volgende slides)

W3C Consortium

Views: desktop mobile print

W3C By Region

STANDARDS PARTICIPATE MEMBERSHIP ABOUT W3C

Skip

WEB AND INDUSTRY

- Automotive
- Entertainment (TV and Broadcasting)
- Publishing
- Web Commerce
- Web Payments
- Web of Data
- Web and Telecommunications
- Web of Things
- Web Security

WEB FOR ALL

- Accessibility
- Internationalization
- W3C A to Z (site map)

W3C WORK GROUPS

- W3C Groups
- Participant guidebook

New Resource: Developing an Accessibility Statement
30 November 2018 | [Archive](#)



The WAI Education and Outreach Working Group (EOWG) has published [Developing an Accessibility Statement](#). The resource helps you create an accessibility statement for your website, mobile app, or other digital content. It includes guidance, examples, and a free generator tool. The tool helps you collect and enter relevant information to create an accessibility statement for your particular content and situation. See more information in [How to Create Accessibility Statements](#) blog post.

Learn more about the [Web Accessibility Initiative \(WAI\)](#).

First Public Working Draft: Extensions to the Semantic Sensor Network Ontology
22 November 2018 | [Archive](#)

First Public Working Draft: CSS Shadow Parts
15 November 2018 | [Archive](#)

New Resource: The Business Case for Digital Accessibility
9 November 2018 | [Archive](#)

TTML1 3rd, TTML2, TTML-IMSC1 are W3C Recommendations
8 November 2018 | [Archive](#)

Selectors Level 3 is a W3C Recommendation
6 November 2018 | [Archive](#)

ABOUT W3C

The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web. W3C operates under a [Code of Ethics and Professional Conduct](#). Become a [Friend of W3C](#): support the [W3C mission](#) and free developer tools.

W3C BLOG

How to Create Accessibility Statements
29 November 2018 by Shadi Abou-Zahra

JSON-LD WG at TPAC 2018
27 November 2018 by Benjamin Young

EPUBCheck intermediate 4.1 release
26 November 2018 by Luc Audrain

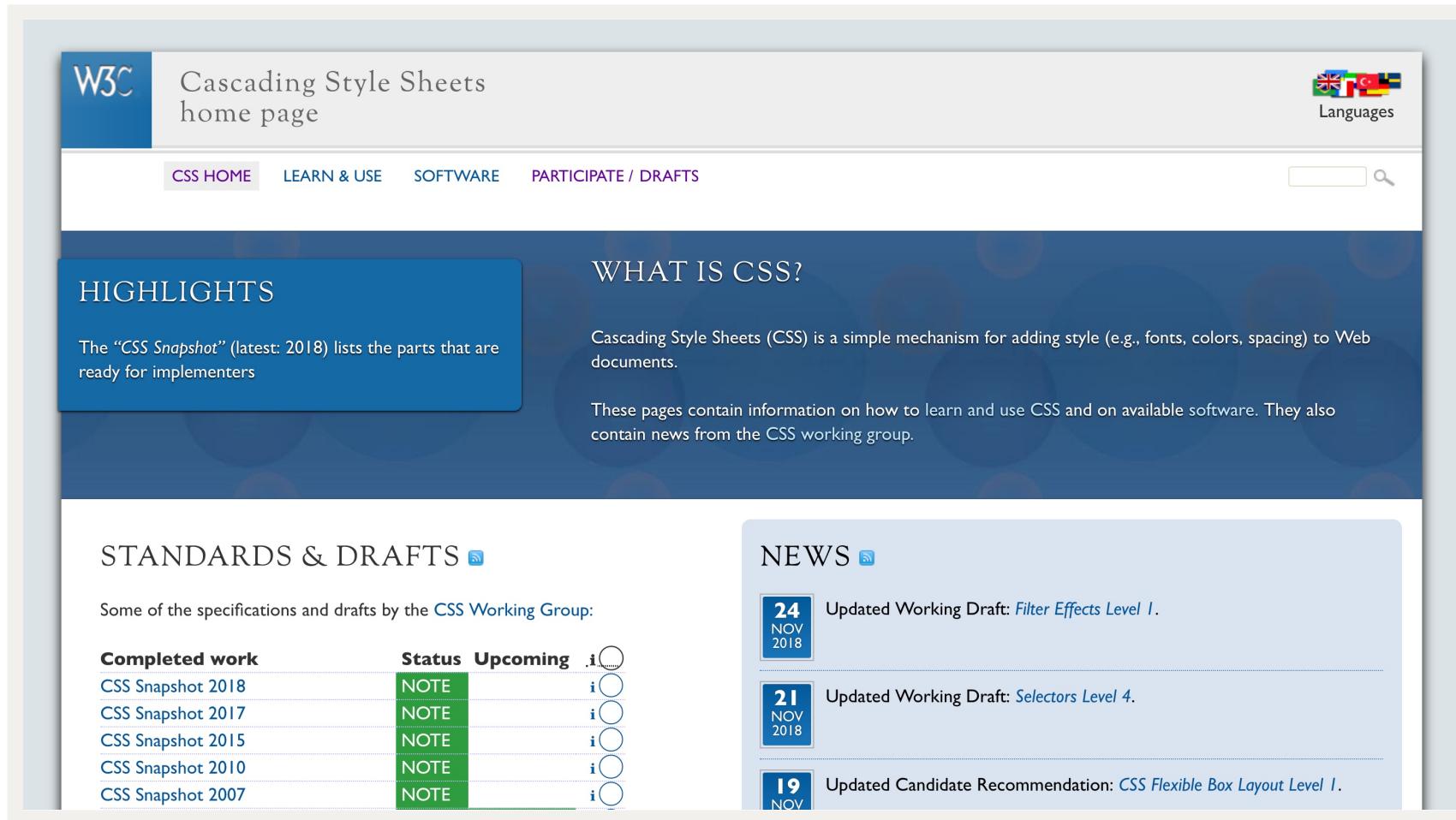
JOBS

We are hiring:

An [Assistant Director of Finance](#)
A [Web Authoring Tools Accessibility Specialist](#)
An [Emerging Web Technologies Accessibility](#)

<https://www.w3.org/>

W3C: CSS home page



The screenshot shows the W3C CSS home page. At the top, there is a navigation bar with the W3C logo, the text "Cascading Style Sheets home page", and language selection options (English, French, German, Spanish, Italian, Portuguese, Chinese, Japanese, Korean, Russian, and Simplified Chinese). Below the navigation bar are links for "CSS HOME", "LEARN & USE", "SOFTWARE", and "PARTICIPATE / DRAFTS". A search bar is also present. The main content area has a dark blue background with white text. On the left, a "HIGHLIGHTS" box contains the text: "The ‘CSS Snapshot’ (latest: 2018) lists the parts that are ready for implementers". To the right, a "WHAT IS CSS?" section defines CSS as a simple mechanism for adding style to Web documents. Below this, a "STANDARDS & DRAFTS" section lists "Completed work" (CSS Snapshot 2018, 2017, 2015, 2010, 2007) and "Upcoming" (all marked as "NOTE"). On the right, a "NEWS" section lists three items: "Updated Working Draft: Filter Effects Level 1." (24 Nov 2018), "Updated Working Draft: Selectors Level 4." (21 Nov 2018), and "Updated Candidate Recommendation: CSS Flexible Box Layout Level 1." (19 Nov).

<https://www.w3.org/Style/CSS/>

W3C: CSS current work and how to participate

W3C CSS current work & how to participate  Languages

CSS HOME LEARN & USE SOFTWARE PARTICIPATE / DRAFTS 

CSS SPECIFICATIONS

This page contains a list of all completed specifications and drafts by the [CSS WG](#) (formerly 'CSS & FP WG'). If you want to follow the development of CSS3, this is the place to start. You have ideas? Contributions? See 'If you want to help' on this page.

A specification is not a manual. There is no excuse for badly written drafts and please complain if you find one. But specs do target a specific audience. See [fantasai's Understanding the CSS Specifications..](#) [J. David Eisenberg](#) has written another useful [How to read W3C specs](#). Or you can read about 'modules', 'levels', 'snapshots' and the CSS process.

WHAT'S NEW?

- 2018-11-24 Updated Working Draft: [Filter Effects Level 1](#).
- 2018-11-21 Updated Working Draft: [Selectors Level 4](#).
- 2018-11-19 Updated Candidate Recommendation: [CSS Flexible Box Layout Level 1](#).
- 2018-11-15 New Working Draft: [CSS Shadow Parts](#). New Note: [CSS Snapshot](#)

	Cur- rent	Up- coming	Notes
Rewriting			
CSS Generated Content Level 3	WD	WD	
Abandoned			
CSS Level 1	SPSD		
CSS Print Profile	NOTE		
CSS Mobile Profile 2.0	NOTE		
The CSS 'Reader' Media Type	NOTE		
CSS Presentation Levels	NOTE		
CSS TV Profile 1.0	NOTE		

<https://www.w3.org/Style/CSS/current-work>

W3C: Grid Layout module

W3C Candidate Recommendation

TABLE OF CONTENTS

- 1 Introduction**
 - 1.1 Background and Motivation
 - 1.1.1 Adapting Layouts to Available Space
 - 1.1.2 Source-Order Independence
- 2 Overview**
 - 2.1 Declaring the Grid
 - 2.2 Placing Items
 - 2.3 Sizing the Grid
- 3 Grid Layout Concepts and Terminology**
 - 3.1 Grid Lines
 - 3.2 Grid Tracks and Cells
 - 3.3 Grid Areas
- 4 Reordering and Accessibility**
- 5 Grid Containers**
 - 5.1 Establishing Grid Containers: the 'grid' and 'inline-grid' 'display' values
 - 5.2 Sizing Grid Containers
 - 5.3 Clamping Overly Large Grids
- 6 Grid Items**
 - 6.1 Grid Item Display
 - 6.2 Grid Item Sizing
 - 6.3 Reordered Grid Items: the 'order' property
 - 6.4 Grid Item Margins and Paddings
 - 6.5 Z-axis Ordering: the 'z-index' property
 - 6.6 Automatic Minimum Size of Grid Items
- 7 Defining the Grid**

CSS Grid Layout Module Level 1
W3C Candidate Recommendation, 14 December 2017

This version: <https://www.w3.org/TR/2017/CR-css-grid-1-20171214/>

Latest published version: <https://www.w3.org/TR/css-grid-1/>

Editor's Draft: <https://drafts.csswg.org/css-grid/>

Previous Versions:

<https://www.w3.org/TR/2017/CR-css-grid-1-20170509/>
<https://www.w3.org/TR/2016/WD-css-grid-1-20160519/>
<https://www.w3.org/TR/2015/WD-css-grid-1-20150917/>
<https://www.w3.org/TR/2015/WD-css-grid-1-20150806/>
<https://www.w3.org/TR/2015/WD-css-grid-1-20150317/>
<https://www.w3.org/TR/2014/WD-css-grid-1-20140513/>
<https://www.w3.org/TR/2014/WD-css-grid-1-20140123/>
<https://www.w3.org/TR/2013/WD-css3-grid-layout-20130402/>
<https://www.w3.org/TR/2012/WD-css3-grid-layout-20121106/>

Test Suite: http://test.csswg.org/suites/css-grid-1_dev/nightly-unstable/

Issue Tracking:

[Disposition of Comments](#)
[Inline In Spec](#)
[GitHub Issues](#)

Editors:

[Tab Atkins Jr. \(Google\)](#)
[Elika J. Etemad / fantasai \(Invited Expert\)](#)
[Rossen Atanassov \(Microsoft\)](#)

Former Editors:

<https://www.w3.org/TR/css-grid-1/>

De kracht van CSS

- CSS is een uitermate krachtig designsysteem
- Veel meer dan wat lettertypen en kleuren bepalen zoals je in het begin gaat doen
- Bewijs? CSS Zen Garden (2003-2013, David Shea)
- Alle designs gebruiken exact dezelfde HTML
- Geen enkel img-element in de html

CSS Zen Garden



A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [HTML FILE](#) and [CSS FILE](#)

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

MID CENTURY MODERN

by Andrew Lohman

GARMENTS

by Dan Mall

STEEL

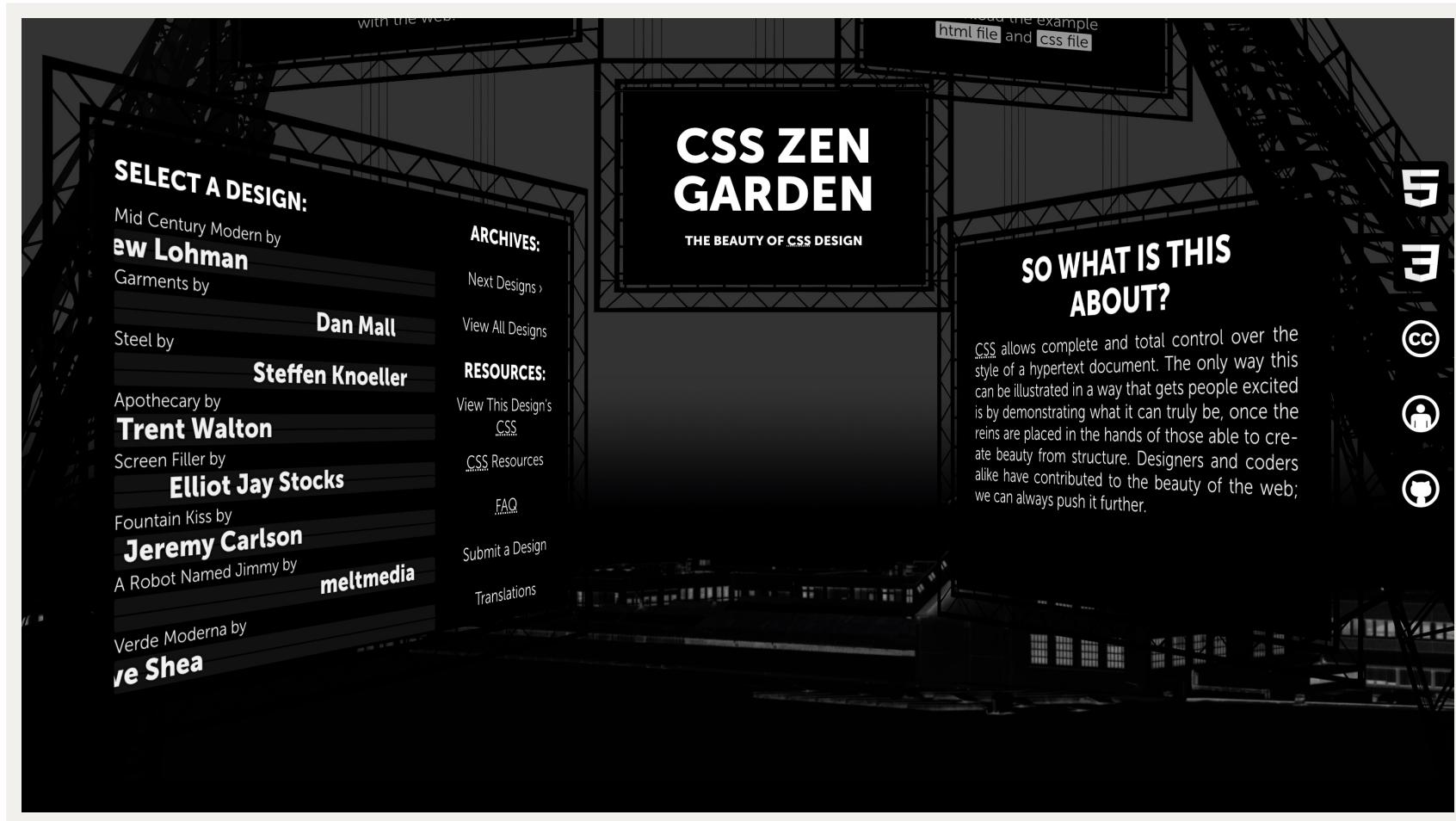
by Steffen Knoeller

APOTHECARY

by Trent Walton

SCREEN FILLER

CSS Zen Garden



<http://www.csszengarden.com>

Hoe stylesheets werken?

- Je maakt het document op met HTML
 - Focus op betekenis en structuur van het document = de structuurlaag
 - Meesterschap hier is cruciaal voor je werk als schrijver van CSS
- Je schrijft een CSS-stylesheet
 - Verzameling stijlregels die beschrijven hoe een (groep van) element(en) getoond moeten worden
- Je voegt de stijlregels toe aan het document

Hoe schrijf je CSS stijlregels?

CSS stijlregels

```
selector {  
    eigenschap: waarde;}  
  
andere_selector {  
    andere_eigenschap: waarde;  
    nog_andere_eigenschap: waarde;}
```

- Een selector identificeert het element of de elementen die je wenst aan te passen
- Daarna volgen tussen accolades {} één of meerdere declaraties
- Een declaratie bestaat uit eigenschappen/waarde-koppels
- De eigenschappen en de waardes worden gescheiden door een dubbelpunt en eindigen op een puntkomma

Selectors

```
h1 {  
    color: green;}  
  
p {  
    font-size: 35px;  
    font-family: sans-serif;}
```

- h1 en p noemen we *element type selectors*
- Dit is het meest eenvoudige type selector
- De declaraties zullen toegepast worden op elke h1 en p in het document

Selectors

```
<h2 id="recept">Titel van mijn recept</h2>
```

```
#recept {  
    color: blue;}
```

- Een ander type selector is de ID selector
- Op basis van het id attribuut van een bepaald element
- Binnen CSS verwijst je ernaar met een hashtag (#)
- Er bestaan naast element type selectors en ID selectors nog tal van andere: zij komen nog aan bod
- Het goed selecteren en gebruiken van selectors is een belangrijke stap in het beheersen van CSS

Declaraties

- Declaraties bestaan uit een eigenschap/waarde paar
- Alle declaraties en de accolades worden samen de declaratieblok genoemd
- Elke declaratieblok kan bestaan uit meerdere declaraties
- Elke declaratie eindigt op een puntkomma om ze te scheiden van de volgende
- Om de leesbaarheid te vergroten geven we alle declaraties hun eigen regel en tab-insprong

```
p {  
  font-size: 35px;  
  font-family: sans-serif;  
  font-weight: 900;  
}
```

Declaraties

- Eigenschappen
 - Centraal bij CSS staan de eigenschappen die aan elementen kunnen worden toegekend
 - Het zijn er momenteel een kleine 360 :-) Wij focussen ons op de belangrijkste
- Waarden
 - Mogelijke waarden variëren van eigenschap tot eigenschap
 - Wanneer je een eigenschap gebruikt is het belangrijk te weten welke waarden mogelijk zijn

CSS toevoegen aan HTML documenten

Er zijn verschillende manieren om CSS toe te voegen aan een HTML document:

- 1. Direct in de HTML pagina

- 2. In een externe CSS bestand

- 3. In een externe CSS bestand en dan in de HTML pagina

- 4. In een externe CSS bestand en dan in de CSS bestand

- 5. In een externe CSS bestand en dan in de CSS bestand

- 6. In een externe CSS bestand en dan in de CSS bestand

- 7. In een externe CSS bestand en dan in de CSS bestand

- 8. In een externe CSS bestand en dan in de CSS bestand

- 9. In een externe CSS bestand en dan in de CSS bestand

- 10. In een externe CSS bestand en dan in de CSS bestand

- 11. In een externe CSS bestand en dan in de CSS bestand

- 12. In een externe CSS bestand en dan in de CSS bestand

- 13. In een externe CSS bestand en dan in de CSS bestand

- 14. In een externe CSS bestand en dan in de CSS bestand

Stijlregels toevoegen aan het HTML document

- Dit kan gebeuren op drie manieren:
 - Een externe stylesheet
 - Een ingesloten stylesheet
 - Inline stijlen toevoegen
- Hint: wij gaan het met een externe stylesheet doen

Een ingesloten stylesheet

- De CSS komt hier binnen een style-element in het head-gedeelte van de pagina
- Draagt niet de voorkeur weg
- Wordt soms gebruikt in combinatie met externe stylesheets voor kleinere aanpassingen

```
<head>
  <title>Titel van het document</title>
  <style>
    /* hier komt jouw CSS */
  </style>
</head>
```

Commentaar in CSS

```
/* Commentaar komt hier terecht */
```

```
/* HEADER STIJLEN */
```

```
body {  
    font-size: 2rem;  
    /* bovenstaande nog aan te passen */  
}
```

```
/* body {  
    font-size: 2rem;  
} */
```

Inline stijlen toevoegen

- Je kan in de HTML stijlen toevoegen met het style attribuut
- Om meerdere stijlen toe te voegen scheidt je ze door een puntkomma
- Absoluut door ons te vermijden

```
<h1 style="color: red">Een rode hoofding</h1>
<h2 style="color: red; margin-top: 2em">Een rode sub-hoofding</h2>
```

Een externe stylesheet

- Dit is een afzonderlijk tekstbestand dat een aantal stijlregels bevat
- De naam van het bestand moet eindigen op .css
- Het .css bestand wordt vervolgens gelinkt met één of meerdere html-documenten
- Op deze manier kunnen alle html-documenten van een website gelinkt worden aan één stylesheet
- Wij gaan deze methode - de meest krachtige! - gebruiken

```
<head>
  <link rel="stylesheet" href="/css/master.css">
</head>
```

CSS Basisconcepten

Overerving

- Net zoals ouders sommige kenmerken doorgeven aan hun kinderen...
- HTML geven sommige stijlen door aan de elementen die ze bevatten
- Overerfbaarheid geeft je een mechanisme om elementen vorm te geven zonder deze zelf expliciet van stijlregels te moeten voorzien
- We vermeldden eerder: HTML documenten volgen een hiërarchie, met ouders en kinderen als een soort familiestamboom

Overerving

- Bepaalde elementen worden doorgegeven, andere niet
- Eigenschappen rond tekst worden over het algemeen doorgegeven: lettertype, kleur, stijl, ...
- Randen, marges, achtergronden die de ruimte rond elementen beschrijft dan weer niet
- Daarom is goede kennis van de HTML documentstructuur van groot belang voor de vormgeving
- Je kan dan overerving voor jou laten werken

```
body {  
    color: blue;  
    font-family: helvetica, sans-serif;  
    margin: 25px;  
}
```

Cascade

- CSS laat toe verschillende stylesheets/stijlregels te koppelen aan zelfde document: mogelijke conflicten!
- Er is een hiërarchisch systeem (cascade = waterval) dat gewichten toekent aan de verschillende stijlregels die gebruikt werden
- Een CSS stijlregel blijft gelden tot ze mogelijk overruled wordt door een stijlregel met meer gewicht
- Dat gewicht is gebaseerd op prioriteit, specificiteit van de selector en de volgorde van de stijlregel

Cascade: prioriteit (van hoog naar laag)

- Elke stijlregel die door de gebruiker gemarkerd is als !important
- Elke stijlregel die door de auteur van de pagina gemarkerd is als !important
- Style sheets geschreven door de auteur van de webpagina (*author style sheet*)
- Style sheets gecreëerd door de bezoeker van de webpagina (*reader style sheet*)
- Interne stylesheet van de browser (*user agent style sheet*)

```
p {  
  color: blue !important;}
```

Cascade: specificiteit

- Conflicten kunnen ook volgen uit meervoudige stijlinstructies
- De aard van de selector zal dan uitmaken wie wint
- Onderstaand voorbeeld: ID-naam krijgt meer gewicht dan de element selector
- Algemene regel: hoe specieker de selector, hoe meer gewicht

```
p {  
    color: orange;}  
  
#mijnparagraaf {  
    color: green;  
}
```

Cascade: volgorde

- Bij conflicten met zelfde prioriteit en specificiteit speelt de volgorde
- De cascade volgt hier het "de laatste wint" principe
- We gaan dit principe in ons voordeel gaan gebruiken, bvb bij ondersteuning voor meerdere schermbreedtes

```
p {  
  color: blue;}  
  
p {  
  color: red;}
```

Het box model

- De hoeksteen van het visueel formatteren van HTML elementen
- De browser ziet elk element (block en inline) als binnen een kleine rechthoekige box
- Je kan aan elk element eigenschappen aan deze box toekennen: borders, margins, padding en backgrounds
- Block elementen kan je positioneren t.o.v. andere elementen, inline element boxes bevatten enkel de content en hier speelt het box model beperkter

```
p {  
  border: 5px solid red;  
  margin: 25px;  
  padding: 15px;  
  background-color: yellow;}
```

Gegroepeerde selectors

- Je kan stijlregels toepassen op meerdere selectoren tegelijk in één enkele stijlregel
- De selectoren dienen gescheiden te worden door een komma
- Belangrijk om je style sheet beknopter en leesbaar te houden

```
h1, h2, p, em, img {  
    border: 1px solid black;}
```

CSS Meeteenheden

CSS meeteenheden

- CSS biedt tal van mogelijkheden om groottes en afstanden uit te drukken
- Gebruik: lettergrootte, breedte en hoogte van elementen, afstanden tussen elementen, ...
- Sommigen ken je en spreken voor zich, andere zijn heel specifiek voor CSS
- Goed en efficiënt gebruik: belangrijk voor soepel werken met CSS
- Twee soorten: absolute en relatieve meeteenheden

```
p {  
    font-size: 14px;  
    margin-bottom: 25px;  
    line-height: 22px;  
}
```

Absolute eenheden

- Hebben een vastgelegde waarde en geven dus een precieze controle
- CSS biedt er een aantal: pt, cm, mm, px
- Meest populaire eenheid is de px of pixel. In CSS3 gedefinieerd als 1/96 inch
- Maar vandaag kennen we een veelheid aan schermen met uiteenlopende pixeldensiteit. Van 68.84 ppi tot wel 445ppi...
- Daarom minder en minder geschikt om vorm te geven voor het web
- Pixels kunnen wel gebruikt worden voor zaken als borders en voor afbeeldingen

Relatieve eenheden

- Relatieve eenheden zijn gebaseerd op iets anders zoals de standaard tekstgrootte of de grootte van het moederelement
- CSS biedt er verschillende: rem, em, vw, vh en %
- Ze zijn door hun aard vandaag beter geschikt om gebruikt te worden binnen de context van de veelheid aan schermen waarvoor we vormgeven

rem

- Nieuw bij CSS3. Staat voor root em en gebaseerd op de lettertype grootte van het root (html) element
- In moderne browsers is deze standaard 16 pixels
- Een rem staat dus gelijk aan 16 pixels (tenzij je de grootte van het root element gaat aanpassen)
- Je kan rem eenheden dus gaan gebruiken als een absolute meeteenheid
- De flexibiliteit zit hem in het veranderen van de grootte van het root element. Alles schaalt dan netjes mee en blijft proportioneel

```
div {  
    height: 10rem; /* zal 160px zijn */  
}
```

rem

Deze methode gaan wij gebruiken:

```
:root {  
    font-size: 62.5%;}  
  
div {  
    height: 10rem; /* zal 10x10px dus 100pixels zijn */  
}  
  
h1 {  
    font-size: 1.6rem; /* zal 1,6x10px dus 16pixels zijn */  
}
```

em

- Een em is een relatieve meeteenheid die uit de traditionele typografie komt
- Gebaseerd op de hoofdletter M (vandaar de naam "em")
- CSS berekent em als de afstand tussen de lijnen waarin de font is gezet
- Je kan em dan ook gebruiken voor insprongen, marges, breedtes en hoogtes
- Belangrijk: em verwijst naar de fontgrootte van het element waarop je de em gaat toepassen
- rem verwijst steeds naar de grootte van het root element en is in die zin eenduidiger

vw en vh

- Viewport width (vw) en viewport height (vh): relatief t.o.v. de viewport
- 1 vw is gelijk aan 1/ 100 van de breedte van de viewport
- 1 vh is gelijk aan 1/ 100 van de hoogte van de viewport
- Geschikt om tekst en afbeeldingen dynamisch af te stemmen op de beschikbare breedte
- Hieraan gerelateerd: vmin (= waarde van vw of vh, kleinste van de twee)
- Hieraan gerelateerd: vmax (= waarde van vw of vh, grootste van de twee)

```
header {  
  width: 90vw;  
  height: 90vh; }
```

Percentages (%)

- Ook een vaak gebruikte eenheid voor webpagina elementen
- Zijn relatief t.o.v. het ouder element
- Zorgt ervoor dat het element proportioneel blijft en vloeiend

```
img {  
  width: 50%;}
```

Interface Design Lab 1 - Web - Inleiding tot CSS

Vragen?
kristof.michiels01@ap.be