

Uitgebreide cursus CSS

http://www.websitemaken.be/index.php?page=show_item&id=107

CSS, de onmisbare taal voor iedere webmaster, leer je nu ook hier. Deze belangrijkste uitbreiding op HTML is noodzakelijk voor een echt goede site.

HTML is bedoelt om je site in te delen, CSS zou je moeten gebruiken voor het 'opmaken' van je site. Ook zul je zien dat je, door gebruik te maken van stylesheets, erg veel tijd kunt besparen als je het design van je site aan zou willen passen. De mogelijkheden van CSS zijn erg groot. Na deze cursus weet je er meer over.

Inleiding: wat is CSS?

Eerst even een inleiding, zodat je beter weet wat het nut van CSS precies is.

In het begin van de korte geschiedenis van het World Wide Web was er HTML, een opmaaktaal om hypertext documenten mee te beschrijven. Opmaak van die documenten was in het begin absoluut niet belangrijk, het ging om de inhoud en het klikken, dat was toen al heel wat.

Maar naarmate de techniek zich ontwikkelde kwam de nadruk steeds meer te liggen op de opmaak van een website, maar de grafische mogelijkheden van HTML schoten duidelijk tekort. Er ontstonden steeds meer tags die slecht op een browser werken(bijvoorbeeld de beruchte IE tag <marquee>). Om dit tegen te gaan begon het World Wide Web Consortium(W3C) met de ontwikkeling van zogenaamde stylesheets, waarin de opmaak van een html-document wordt beschreven. Zo wordt de site meer platformafhankelijk. Een andere bedoeling was het aantal tags(en dus de tijd die je in je site moet steken) te verminderen. Dat is allemaal aardig gelukt in Cascading Style Sheets(CSS), Engels voor Trapsgewijs Opmaakmodel. Een ander groot voordeel is dat het nu veel makkelijker is om een aantal documenten dezelfde stijl mee te geven en die stijl ook snel weer te veranderen. Hoe we dat gaan doen, leren we je nu.

Het eerste wat je moet weten is dat er drie manieren zijn om een stijl aan je html-bestand toe te kennen(CSS is een aanvulling en zeker geen vervanging van HTML!). Deze drie zijn:

- ♦ Linked Stylesheet: je linkt een html-bestand (of zoveel je wilt) aan een stijlbestand op je server.
- ♦ Embedded Stylesheet: Je definiëert de stijlen in het html-bestand zelf.
- ♦ Inline Stylesheets: Zo kun je een paar tags, of een stuk tekst, een bepaalde style toekennen. Hier heb je nog niets aan, dus we zullen de manieren wat verder toelichten, op de volgende pagina's.

Linked stylesheets

Als eerste dus de linked stylesheets. Zoals gezegd koppel je een stylesheet aan een of meer html-bestanden. Over hoe je zo'n stylesheet maakt hebben we het straks wel, eerst leggen we je even uit hoe je een stylesheet aan het html-bestand koppelt. We gaan er even vanuit dat je HTML al onder de knie hebt, zoniet, volg dan de HTML-cursus nog even.

Je weet dat een html-pagina er zo uit kan zien:

```
# Code
1 <html>
2 <head>
3 <title>Welkom op mijn website!</title>
4 </head>
5 <body>
6 <p>Dit is mijn website</p>
7 </body>
8 </html>
```

Dit stelt natuurlijk niet veel voor, maar daar gaat het nu ook niet om. Het gaat hier even om de manier waarop we de stylesheet koppelen. En dat doen we in de head-sectie, dus tussen <head> en </head>. Koppelen we aan het bovenstaande document bijvoorbeeld het stylesheet opmaak.css, dan ziet de head-sectie er zo uit:

```
# Code
1 <head>
2 <title>Welkom op mijn website!</title>
3 <link rel="stylesheet" type="text/css" href="opmaak.css">
4 </head>
```

Helemaal niet moeilijk dus, zoals je ziet. Nu hebben we dus aangegeven dat de website er volgens de door ons opgegeven opmaak uit moet zien. Volgende vraag is natuurlijk: hoe maken we dan zo'n stylesheet? Ook dit is weer behoorlijk eenvoudig, het lijkt een beetje op HTML, maar er zijn toch zeker verschillen. Ten eerste kun je het gewoon in bijvoorbeeld het Windows-programma Kladblok of Notepad maken. We zullen hieronder laten zien hoe je de stijl van een alinea (de tag <p>) aangeven:

```
# Code
1 p {
2 font-size: 11 pt;
3 font-family: Arial, Swiss, Helvetica;
4 color: black;}
```

Waarschijnlijk spreekt het allemaal voor zich, maar we zullen het allemaal toch even toelichten. Met p (dit noem je een *selector*) geef je natuurlijk aan dat het hier op de tag <p> gaat, tussen de { en } staat de stijl aangegeven. Verder heten bijvoorbeeld font-size en color *properties*, de waarde ervan, bijvoorbeeld 11pt en black heten *values*. Na iedere value volgt een punt-komma teken.

Met 11pt hebben we de hoogte opgegeven. Dat had ook gekund in pixels (px), inches, centimeters of wat dan ook. Met font-family geven we het lettertype aan. Door 3 lettertypes op te geven gaat de browser ze allemaal af. Wordt bijvoorbeeld Arial niet op de pc aangetroffen, dan zoekt hij evrder naar Swiss. Is ook die niet te vinden, dat gaat hij voor Helvetica. Je kunt er ook nog meer opgeven.

De kleur kun je zowel als hexadecimale code(#000000) of als woord(black) opgeven.

Koppel je dit bestand aan bijvoorbeeld 15 html-documenten, dan worden de voordelen al snel duidelijk. Kom je er bijvoorbeeld achter dat zwart toch wel erg slecht te lezen is op een donkerblauwe achtergrond en wil je de tekst wit maken, dan hoef je niet bij elke alinea de kleur op te geven. Het enige wat je doet is black veranderen in white, en alles is aangepast! Het zal duidelijk zijn dat dit enorme tijdwinst oplevert, en je een hoop vervelend werk afneemt.

Embedded stylesheets

Behalve in een extern stylesheet, kun je de style ook opgeven in het html-document zelf. Dit noemen we een Embedded Stylesheet. Ook dat doe je weer in de head-sectie. Hieronder zie je hoe:

```
# Code
1 <html>
2 <head>
3 <title>Welkom op mijn website!</title>
4 <style type="text/css">
5 <!--
6 p {
7 font-size: 11 pt;
```

```
8 font-family: Arial, Swiss, Helvetica;
9 color: black;}
10 -->
11 </style>
12 </head>
```

We geven de tag `<p>` dezelfde eigenschappen als we deden bij de linked stylesheets. Het meeste zal je wel duidelijk zijn, je vraagt je waarschijnlijk alleen af waarom er `<!--` en `-->` staat. Dit zijn zogenaamde commentaar-tags, waarin je commentaar kunt toevoegen aan je site, waar de browser verder niets mee doet, een soort aantekeningen zeg maar. Hier gebruiken we dat ook, om te zorgen dat browsers die geen CSS ondersteunen de tags negeren. Browsers die CSS ondersteunen zullen juist de commentaartags negeren en de styletags toch gebruiken.

Houdt er dus rekening mee dat de style die je hier opgeeft alleen geldt voor het webdocument waarin je de style opgeeft.

Naast algemene styles op te geven, kun je ook de opmaak voor slechts één element op je website bepalen. Daarvoor zijn de Inlinestyles. Dat leer je hierna.

Inline stylesheets

De laatste manier om een style in je html document op te nemen is de inline-style. Hiermee kun je een bepaald deel van je pagina een style toekennen, bijvoorbeeld alleen een alinea, een link of een tabel. We moeten hiervoor CSS en HTML gaan combineren, en dat gaat vrij simpel. Willen we bijvoorbeeld een alinea 15 pixels laten inspringen vanaf de rand, en moet de kleur rood zijn, dan doen we dat zo:

```
# Code
1 <p style="margin-left: 15px; color: red">Dit is de alinea waar het allemaal over gaat</p>
```

Je ziet het, het is helemaal niet moeilijk. tussen `style="` en `"` komt de style te staan. De verschillende dingen die je opgeeft scheidt je van elkaar door middel van de punt-komma.

Wil je nu bijvoorbeeld dat een bepaalde style op 4 alinea's van toepassing is, dan zou je dus bij alle vier de `<p>` tags de style moeten opgeven. Dat is behoorlijk veel werk, maar daar is gelukkig iets op bedacht: het element `<div>`. Hiermee kun je een bepaald gedeelte van je site markeren. De vier alinea's uit ons voorbeeld omringen we dan met de openingstag `<div>` en de sluitingstag `</div>`. Daartussen staan de vier alinea's.

Het enige wat je nu nog moet doen is in de openingstag de style opgeven, ook weer erg makkelijk:

```
# Code
1 <div style="margin-left: 15px; color: red">
2 <p>Alinea 1</p>
3 <p>Alinea 2</p>
4 <p>Alinea 3</p>
5 <p>Alinea 4</p>
6 </div>
```

Je ziet het, stylesheets kunnen je een hoop werk schelen. Maar er zijn nog meer voordelen, dat vertellen we je in deel 5.

Je eigen styles maken

Het kan allemaal nog beter: je eigen styles maken. Ook dat is met CSS mogelijk. Dus niet alleen de bestaande styles `p`, `table`, `td` enzovoorts, maar bijvoorbeeld mijnstijl. Wil je

bijvoorbeeld een stuk tekst in het rood, de linkermarge op 15 pixels en onderstreept, dan kan dat zonder telkens de style op te geven. Je maakt gewoon in je linked of je embedded stylesheet de style aan, en wel zo:

```
# Code
1 .mijnstijl{
2 color: red;
3 margin-left: 15px;
4 text-decoration: none
5 }
```

Zo, de style is nu bekend in de stylesheet. Nu moeten we hem alleen nog koppelen aan een bepaalde alinea, of een <div>, of wat jij wilt. Willen we de vier alinea's uit de vorige les in mijnstijl hebben, dan doe je dat zo:

```
# Code
1 <div class="mijnstijl">
2 <p>Alinea 1</p>
3 <p>Alinea 2</p>
4 <p>Alinea 3</p>
5 <p>Alinea 4</p>
6 </div>
```

We doen het dus met het attribuut class. Let erop dat we in de stylesheet een punt voor de style neerzetten, maar bij de class doe je dat niet!

Meer over styles

Links omschrijf je in CSS als volgt

```
# Code
1 a:link {
2 font-size: 11pt; color: blue; text-decoration: underline;}
3 a:visited {
4 font-size: 11pt; color: black; text-decoration: none;}
5 a:over {
6 font-size: 11pt; color: red; text-decoration: underline;}
7 a:active {
8 font-size: 11pt; color: black; text-decoration: underline;}
```

Dat is even heel wat informatie, we zullen het eerst bespreken. a.link geeft aan dat als we te maken hebben met een link, dus alles wat tussen en staat, 11 punten groot moet zijn, de kleur blauw moet zijn en dat er een streepje onder moet zitten. dat laatste geven we aan met text-decoration: underline;

Vervolgens zie je a.visited, a.over en a.active. De eerste geeft een link aan die door de bezoeker al eens bezocht is, de tweede geeft aan wat er moet gebeuren als je met je muis over de link zweeft (zweven=to hover) en de derde geeft de eigenschappen van een link aan die is aangeklikt. Let op: de vier linkproperties moeten altijd in deze volgorde worden gebruikt, omdat anders het effect van hover en active verdwijnen!

Verder kunnen we zo'n beetje iedere tag wel een een style meegeven. Bij de body kan dat ook, hier kunnen we wat meer mee dan met gewone tags. Ten eerste kun je door

```
# Code
1 body{
2 font-size: 11pt; color: black;}
```

op te geven alle tekst die niet binnen een <p> of een ander attribuut staat een kleur, grootte en dergelijke meegeven.

Ten tweede, en nu wordt het leuk, kunnen we de randen van de pagina opgeven. Zoals je ook op deze site ziet, is er links een witte rand. Wil je boven geen rand, dan kun je dat ook opgeven. Dit kan overigens ook in HTML, maar we leren het je nu in een stylesheet te doen.

```
# Code
1 body {
2 margin-left:0.5%;
3 margin-right:0.5%;
4 margin-bottom:0.5%;
5 margin-top:0%;}
```

Dit spreek allemaal voor zich, maar toch nog even: left is de linkerkant, right de rechter, bottom is de onderkant en top, je raadt het nooit, de bovenkant. Verder kun je de marges opgeven in pixels maar ook zoals hierboven in procenten. Pixels doe je door bijvoorbeeld margin-left: 2px;.

De margins kun je ook bij andere elementen gebruiken, zoals de <p>-tag. Dan krijg je een soort inspring-effect. Je kunt op de pixel nauwkeurig bepalen waar de tekst begint. Dit is met HTML onmogelijk.

Tekst en CSS

Hier zullen we bespreken wat je allemaal in CSS met tekst kunt doen. We hebben het hier over het element text. We kunnen hier het volgende mee doen:

Eigenschap	Werking	Mogelijkheden
text-decoration	geeft de decoratie aan	none, underline, overline, line-through, blink.
vertical-align	Verticale uitlijning bepalen t.o.v. ander object(parent)	baseline, sub, super, top, text-top, middle, bottom, text-bottom
text-transform	Zet tekst om in hoofdletters, kleine letters of beginhoofdletters	capitalize, uppercase, lowercase, none
text-align	Horizontale uitlijning van de tekst	left, right, center, justify(uitvullen)
text-indent	Inspringing van eerste tekstregel	alle waarden in px, pt, cm, em en %. Mag ook negatief zijn
line-height	Regelafstand	normal, en alle waarden in px, pt, cm, em en %.

Verder kunnen we ook nog veel met de lettertypen doen. dat zie je hieronder:

Eigenschap	Werking	Mogelijkheden
font-size	Puntgrootte van de letter	absolute waarden(xx-small tot xx-large), relatief(smaller of larger) en eenheden(pt, px, cm, em en %)
font-family	Het lettertype, meerdere opties mogelijk, scheiden door een komma, browser kiest de eerste die hij herkent.	Alle lettertypen, bijvoorbeeld Arial, Swiss, Helvetica(drie ongeveer gelijke lettertypen)
font-weight	Gewicht van de letter(hoe vet)	normal, bold, bolder, lighter of 100, 200, 800 etc.

font-style	opmaakkenmerken zoals <i>cursief</i>	normal, italic, oblique
font-variant	kleinkapitaal variatie op lettertype	normal of small-caps

Achtergronden met CSS

De achtergrond van je pagina instellen in HTML, zoals je kunt leren in onze afbeeldingencursus, is erg simpel, maar aan de andere kant zijn de mogelijkheden beperkt. de afbeelding begint altijd linksbovenin en gaat zich daarna herhalen. Met CSS zijn er veel meer opties. Bovendien kun je niet alleen de achtergrond van je pagina instellen, maar ook alleen die van een link, een alinea of tabel.

Een achtergrondkleur geef je op als *background-color*. Moet de achtergrondkleur rood worden, dan doen we dus zo:

```
body{
background-color: #ff0000}
```

of natuurlijk

```
body{
background-color: red}
```

Dat moet wel lukken. Een afbeelding is iets lastiger, dat werkt zo:

```
background-image:
url(afbeelding.gif)
```

Het mooie van CSS is dat je kunt opgeven of en zoja hoe de achtergrondafbeelding zich moet herhalen. Dat doen we met `body{background-repeat: optie}`. Dit zijn de opties:

Waarde	Gevolg
repeat	herhaal alles
repeat-x	herhaal horizontaal
repeat y	herhaal verticaal
no-repeat	geen herhaling

Ook mooi is dat je kunt zeggen of de afbeelding stil moet blijven staan of herhaald moet meebewegen als je gaat scrollen. Meebewegen geven we aan met

```
body{background-attachment: scroll}
```

Stil blijven staan krijg je met

```
body{background-attachment: fixed}
```

Als laatste is het nog mogelijk om precies op te geven waar t.o.v. de linkerbovenhoek moet beginnen. Dat doen we met `body{background-position: optie}`. Hieronder zie je de opties:

```
relatief(0%, 0%, dit is de linkerbovenhoek)
absoluut(cm, px of pt t.o.v. de linkerbovenhoek
benoemd(bottom, top, left, right.) Bottom, right wordt dus rechtsonderin.
```

We zeiden het al, het hoeft niet per se om een hele afbeelding te gaan. Ook een link is mogelijk. Zo kun je het dus ook voor elkaar krijgen om een rode link een gele achtergrond te geven als je er met de muis overheen beweegt. Dat doen we zo. Eerst geven we de style van een gewone link op:

```
a:link{  
font-size: 11pt;
```

```
color: red;Let op:
```

```
geen font-color gebruiken
```

```
background: none}
```

Dan de style van de link als de muis erover heen gaat

```
a:hover{  
font-size: 11pt;  
  
color: red;  
  
background: yellow}
```

Tot zover deze cursus CSS. Meer informatie over CSS vind je, naast onze artikelen, op [de website van W3C](#).