

# Cascading Style Sheets (CSS)

Copyright © 2001 - 2005 [Edwin Martin](mailto:edwin@bitstorm.nl) <[edwin@bitstorm.nl](mailto:edwin@bitstorm.nl)>

Laatst aangepast: 4 juni 2005.

Bij veel voorbeelden in de cursus wordt gebruik gemaakt van style sheets. Het spreekt vanzelf dat deze cursus bekeken dient te worden met een recente browser. Browsers met een goede style sheet-ondersteuning zijn Internet Explorer 6, Netscape 6, Mozilla en Firefox.

Na deze cursus en een cursus JavaScript kan de cursus DHTML worden gevolgd:

- [Cursus Dynamic HTML](#)

Onbekende woorden en kleuren kunnen worden opgezocht in deze lijsten:

- [Terminologie](#)
- [Kleuren](#)

Indien gewenst, kan deze cursus ook als PDF-bestand worden gedownload:

- [Cursus Cascading Style Sheets.pdf](#) (152 kB)

## Een beetje geschiedenis

Wie wel eens de specificaties van [HTML 2.0](#) heeft bekeken, zal het opgevallen zijn dat er haast geen mogelijkheden zijn om een pagina vorm te geven.

Gek genoeg was dat precies de bedoeling. We konden aangeven wat de titel was, wat de kopjes in een tekst waren en welke stukken tekst benadrukt moesten worden. Hoe de tekst uiteindelijk werd vormgegeven, viel buiten het bestek van HTML.

Maar met het populairder worden van het Internet heeft toenmalig marktleider Netscape op een gegeven moment besloten om tags te verzinnen waarmee we onder andere teksten verschillende kleuren konden geven en een plaatje als achtergrond van een pagina konden gebruiken. Webmasters over de hele wereld waren hier zo van verrukt, dan in een mum van tijd het halve Internet was voorzien van achtergrondplaatjes en gekleurde teksten.

Duidelijk werd dat men webpagina's mooi wil vormgeven. Duidelijk werd echter ook dat HTML daar helemaal niet geschikt voor is. Wie wel eens de frustraties heeft aanheard van een DTP-er die voor het eerst een webpagina poogt vorm te geven, weet wat bedoeld wordt.

Al snel werd duidelijk dat het slimmer is om de vormgeving in een style sheet (stijlblad) onder te brengen. In een style sheet kan onder andere worden vastgelegd dat een kopje tussen <H2> en </H2> een 24-punts vette Helvetica moest zijn.

Ook werd duidelijk dat de vormgever aan een webpagina een style sheet wil koppelen, maar dat de browser ook een ingebakken style sheet heeft en dat de bezoeker ook zo zijn voorkeuren heeft en een eigen style sheet aan de browser kan koppelen.

Om al deze style sheets bij elkaar te kunnen voegen werd een taal bedacht die style sheets beschreef en dit mogelijk maakte: Cascading Style Sheets (Overlappende stijlbladen). Met CSS is het mogelijk om allerlei HTML-elementen van vorm-kenmerken te voorzien.

In december 1996 werd [Cascading Style Sheets, level 1 \(CSS1\)](#) een 'W3C Recommendation' (aanbeveling).

Rond deze tijd komen Netscape Navigator 4 en Internet Explorer 3 uit met gedeeltelijke style sheets ondersteuning. Latere versies van deze browsers ondersteunen style sheets steeds beter. De versies 6 van beide browsers ondersteunen CSS1 volledig.

Een van de belangrijkste voordelen van style sheets is dat het backwards compatible is. Oudere browsers en eenvoudige browsers op bijvoorbeeld handcomputers, die geen CSS ondersteunen, laten alleen de eenvoudige HTML zien en nieuwere browsers laten mooiere en betere pagina's zien.

## HTML terminologie: element en attribuut

Voordat we echt van start gaan, eerst nog wat terminologie die in deze tekst gebruikt wordt, namelijk *element* en *attribuut*.

Als voorbeeld nemen we onderstaande HTML-code:

```
<h2>Vakantie in Carvoeiro</h2>

<p>Onze vakantie in <a href="http://www.carvoeiro.com/">Carvoeiro</a></p>
```

Een **element** is de HTML-open tag, de eventueel aanwezige sluit tag en de eventueel aanwezige inhoud. Zo is `<h2>Vakantie in Carvoeiro</h2>` een H2-element en `` een img-element.

Een **attribuut** is een parameter van een element (HTML-tag), in dit voorbeeld `src`, `width`, `height` of `href`.

## Een eerste style sheet

Stel, we hebben voor Katja de volgende webpagina gemaakt:

```
<html>
<head>
<title>Mijn homepage</title>
```

```
</head>
<body bgcolor="white">

<h1>Katja's homepage</h1>

Deze bands vind ik ongeloofelijk <i>cool</i>:

<ul>
<li>Destiny's Child
<li>Anouk
<li>Krezip
</ul>

</body>
</html>
```

Deze pagina zal er zo uitzien:

# Katja's homepage

Deze bands vind ik ongeloofelijk *cool*:

- Destiny's Child
- Anouk
- Krezip

Stel, we willen de kop met de tekst 'Katja's homepage' oranje hebben, want dat is Katja's lievelingskleur. De tekst tussen `<h1>` en `</h1>` moet dus oranje (orange) worden. Met Cascading style sheets wordt dit als volgt weergegeven:

```
h1 {color: orange}
```

Eenvoudig, niet? Eerst de naam van de HTML-tag en daarachter, tussen accolades, de stijkenmerken.

Hoe nemen we deze stijl nu in een HTML pagina op?

We gebruiken daarvoor de `<style>`-tag. Om aan te geven dat het om CSS gaat, geven we als attribuut `type="text/css"` mee.

```
<style type="text/css">
h1 {color: orange}
</style>
```

Het is gebruikelijk om styles in de head van het document te plaatsen. Katja's homepage komt er dan als volgt uit te zien:

```
<html>
<head>
<title>Mijn homepage</title>
<style type="text/css">
h1 {color: orange}
</style>
</head>
<body bgcolor="white">

<h1>Katja's homepage</h1>

Deze bands vind ik ongeloofelijk <i>cool</i>:
<ul>
<li>Destiny's Child
<li>Anouk
<li>Krezip
</ul>

</body>
</html>
```

In een browser wordt dit dan:

# Katja's homepage

Deze bands vind ik ongeloofelijk *cool*:

- Destiny's Child
- Anouk
- Krezip

Zoals zichtbaar is, neemt het de kopkenmerken van de browser over (vet en groot) en voegt het daar onze kenmerken aan toe (oranje). Dit wordt met cascading (overlappend) bedoeld.

## Een iets uitgebreidere style sheet

Natuurlijk kunnen we meerdere stijkenmerken geven. We willen bijvoorbeeld de kop helemaal niet vet, maar normaal (`font-weight: normal`) en we willen de kop 20 pixels laten inspringen (`margin-left: 20px`). Als we ook nog schuine tekst willen onderstrepen (`text-decoration: underline`), dan wordt de uiteindelijke pagina als volgt:

```
<html>
<head>
<title>Mijn homepage</title>
<style type="text/css">
h1 {color: orange; font-weight: normal; margin-left: 20px}
i {text-decoration: underline}
</style>
</head>
<body bgcolor="white">
```

```

<h1>Katja's homepage</h1>

Deze bands vind ik ongeloofelijk <i>cool</i>:
<ul>
<li>Destiny's Child
<li>Anouk
<li>Krezip
</ul>

</body>
</html>

```

Het resultaat in een browser:

## Katja's homepage

Deze bands vind ik ongeloofelijk *cool*:

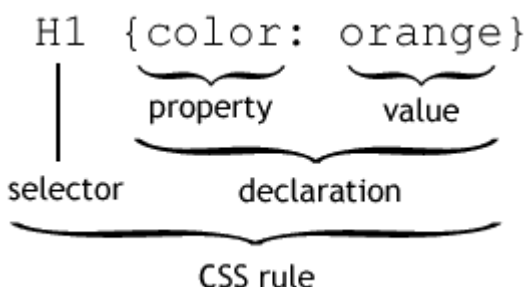
- Destiny's Child
- Anouk
- Krezip

Het resultaat is duidelijk: de kop is ingesprongen en niet meer vet en cursieve tekst is nu ook onderstreept.

Tussen de accolades kunnen dus meerdere stijkenmerken geplaatst worden. Ze worden gescheiden door de punt-komma (;).

## CSS terminologie

CSS kent zijn eigen terminologie, die aan de hand van een voorbeeld wordt uitgelegd:



Bovenstaand voorbeeld "H1 {color: orange}" is een eenvoudige **CSS rule** (regel). Een CSS rule bestaat uit een **selector** 'H1' en een **declaration** (declaratie) 'color: orange'. De declaration bestaat weer uit een **property** (eigenschap) 'color' en **value** (waarde) 'orange'.

## De paragraaf-tag <p>

Als we niet alleen het kopje willen laten inspringen, maar ook de normale paragraaftekst, dan kan dat vanzelfsprekend ook. We zetten daarvoor elke paragraaf tussen `<p>` en `</p>`.

Een voorbeeld van de paragraaf 20 pixels laten inspringen:

```
<html>
<head>
<title>Mijn homepage</title>
<style type="text/css">
H1 {color: orange; font-weight: normal; margin-left: 20px}
I {text-decoration: underline}
P {margin-left: 20px}
</style>
</head>
<body bgcolor="white">

<h1>Katja's homepage</h1>

<p>Deze bands vind ik ongeloofelijk <i>cool</i>:</p>
<ul>
<li>Destiny's Child
<li>Anouk
<li>Krezip
</ul>

</body>
</html>
```

Het resultaat in een browser:

## Katja's homepage

Deze bands vind ik ongeloofelijk *cool*:

- Destiny's Child
- Anouk
- Krezip

Om HTML-documenten zo style sheet-vriendelijk te maken, is het slim om voortaan alle paragrafen tussen `<p>` en `</p>`-tags te zetten.

### De body-tag `<body>`

Als we nu in het hele document één bepaald lettertype willen gebruiken, dan kunnen we dat in de style sheet aangeven voor de paragraaftekst, de kopjes en alle andere gebruikte elementen. Het is veel handiger om dit voor het hele document tegelijk te doen. We gebruiken hiervoor het body-element. Met het body-element veranderen we eigenschappen voor het hele document. Bijvoorbeeld ook de achtergrondkleur. We halen

dan natuurlijk de 'bgcolor="white"' uit de <body>-tag.

```
<html>
<head>
<title>Mijn homepage</title>
<style type="text/css">
H1 {color: orange; font-weight: normal; margin-left: 20px}
I {text-decoration: underline}
P {margin-left: 20px}
BODY {font-family: Helvetica; background-color: mistyrose}
</style>
</head>
<body>

<h1>Katja's homepage</h1>

<p>Deze bands vind ik ongeloofelijk <i>cool</i>:</p>
<ul>
<li>Destiny's Child
<li>Anouk
<li>Krezip
</ul>

</body>
</html>
```

Het resultaat in een browser:

## Katja's homepage

Deze bands vind ik ongeloofelijk *cool*:

- Destiny's Child
- Anouk
- Krezip

Het lettertype is nu Helvetica en de achtergrondkleur zacht roze.

## CSS1 Stijlkenmerken

Er zijn erg veel kenmerken die we kunnen veranderen. Hieronder staat een lijst met de CSS1-kenmerken. Ze zullen allemaal worden besproken.

De laatste browsers ondersteunen ook een aantal kenmerken van het nieuwere CSS2. Aan het einde worden de belangrijkste vernieuwingen besproken.

### Font properties

font-family  
font-style  
font-variant

	font-weight font-size font
<b>Color and background properties</b>	color background-color background-image background-repeat background-attachment background-position background
<b>Text properties</b>	word-spacing letter-spacing text-decoration vertical-align text-transform text-align text-indent line-height
<b>Box properties</b>	margin-top margin-right margin-bottom margin-left margin padding-top padding-right padding-bottom padding-left padding border-top-width border-right-width border-bottom-width border-left-width border-width border-color border-style border-top border-right border-bottom border-left border width height float clear
<b>Classification properties</b>	display white-space list-style-type list-style-image list-style-position



list-style

## Lettertypes (fonts)

Wat kan met de <font>-tag, kan natuurlijk ook met style sheets. De verschillende font-eigenschappen zijn:

Eigenschap	Waarde	Betekenis
<b>font-family</b>	Lijst van lettertypes	Zie hieronder.
<b>font-style</b>	<b>normal</b> <b>italic</b> <b>oblique</b>	Cursief of schuin schrift.
<b>font-variant</b>	<b>normal</b> <b>small-caps</b>	Bij small-caps worden onderkast (kleine) letters weergegeven als kapitalen (hoofdletters), maar iets kleiner. Voorbeeld: SOPHIA. (Internet Explorer ondersteunt dit correct vanaf versie 6).
<b>font-weight</b>	<b>normal</b> <b>bold</b> <b>bolder</b> <b>lighter</b> <b>100 - 1000</b>	Geeft mate van vetheid aan. De gangbare browsers ondersteunen enkel normal en bold.
<b>font-size</b>	<b>xx-small</b> <b>x-small</b> <b>small</b> <b>medium</b> <b>large</b> <b>x-large</b> <b>xx-large</b> <b>larger</b> <b>smaller</b> <i>grootte</i> percentage	Geeft de grootte van een font aan.
<b>font</b>	Meerdere eigenschappen tegelijk: font-style font-variant font-weight font-size font-family	

Bij `font-family` kunnen we, net als met de <font>-tag, een lijstje met lettertypes opgeven, gescheiden door komma's. De eerste font in het lijstje dat beschikbaar is, wordt gebruikt. Als de naam van een font een spatie bevat, dient de naam tussen aanhalingstekens te staan.

Aan het einde van de lijst kan een generiek font toegevoegd worden, die zal worden gebruikt als geen van de andere fonts beschikbaar is. De generieke fonts zijn:

- serif

- sans-serif
- cursive
- fantasy
- monospace

Voorbeeld met meerdere lettertypen:

```
H1 {font-family: "Trebuchet MS", Arial, Helvetica, sans-serif}
```

Voorbeeld met meerdere eigenschappen:

```
H1 {font: italic normal normal x-large "Trebuchet MS", Arial, Helvetica, sans-se
```

## Kleur en achtergronden

De bekende 'background' en 'bgcolor'-attributen hebben natuurlijk ook een CSS-evenknie. Hieronder staat een lijstje met de verschillende mogelijkheden.

Eigenschap	Waarde	Betekenis
<b>color</b>	<a href="#">kleur</a>	Tekstkleur.
<b>background-color</b>	<a href="#">kleur</a>	Achtergrondkleur.
<b>background-image</b>	<b>none</b> <b>url(path)</b>	Achtergrondplaatje.
<b>background-repeat</b>	<b>repeat</b> <b>repeat-x</b> <b>repeat-y</b> <b>no-repeat</b>	Achtergrond herhalen, alleen horizontaal herhalen, alleen vertikaal herhalen en niet herhalen.
<b>background-attachment</b>	<b>fixed</b> <b>scroll</b>	Vaste of meescrollende achtergrond.
<b>background-position</b>	y x y: <b>top</b> <b>center</b> <b>bottom</b> <a href="#">grootte</a> percentage x: <b>left</b> <b>center</b> <b>right</b> <a href="#">grootte</a>	Geeft de positie van het achtergrondplaatje weer met twee waarden y x, waarbij y de waarden top, center of bottom en x de waarden left, center of right kan hebben. Ook kan voor beiden een percentage of lengte worden opgegeven.

	percentage
<b>background</b>	Alle achtergrondeigenschappen kunnen ook in één keer worden opgegeven: background-attachment background-color background-image background-position background-repeat

Voorbeeld:

```
body {
  background-color: #006600;
  background-image: url(circle_bg.gif);
  background-position: bottom center;
}
```

In tegenstelling tot onze eerdere notatie, staan hier de kenmerken onder elkaar. Dit is veel overzichtelijker dan als de kenmerken achter elkaar staan.



Met bovenstaande afbeelding en code wordt het resultaat als volgt:



Dit voorbeeld laat zien dat een achtergrondafbeelding niet altijd linksboven hoeft te beginnen, maar bijvoorbeeld ook midden-onder kan beginnen.

## Tekstkenmerken

Met CSS kan, net als in HTML, tekst vet, cursief en onderstreept worden weergegeven. Met CSS kan ook de ruimte tussen regels, woorden en zelfs letters worden bepaald.

Eigenschap	Waarde	Betekenis
<b>word-spacing</b>	<b>normal</b> grootte	De ruimte tussen woorden. Dit is een voorbeeld. (Werkt in Internet Explorer vanaf versie 6).
<b>letter-spacing</b>	<b>normal</b> grootte	De ruimte tussen letters. Dit is een voorbeeld.

<b>text-decoration</b>	<b>none</b> <b>underline</b> <b>overline</b> <b>line-through</b> <b>blink</b>	Voorbeelden: _____ <u>underline</u> . <u>overline</u> , <del>line-through</del> en blink. Een browser kan <b>blink</b> (knipperen) negeren.
<b>vertical-align</b>	<b>baseline</b> <b>sub</b> <b>super</b> <b>top</b> <b>text-top</b> <b>middle</b> <b>bottom</b> <b>text-bottom</b> <b>percentage</b>	Er zijn drie soorten verticale uitlijning: <ul style="list-style-type: none"> <li>• <b>baseline</b> ■, <b>text-top</b> ■ en <b>text-bottom</b> ■ zijn afhankelijk van de huidige tekstregel.</li> <li>• <b>top</b>, <b>middle</b>, <b>bottom</b> en <b>percentage</b> zijn afhankelijk van de box waarin het element zich bevindt.</li> <li>• Een voorbeeld van <b>sub</b>: H<sub>2</sub>O en van <b>super</b>: 25 m<sup>2</sup>.</li> </ul>
<b>text-transform</b>	<b>none</b> <b>capitalize</b> <b>lowercase</b> <b>uppercase</b>	Voorbeelden: Capitalize, lowercase en UPPERCASE.
<b>text-align</b>	<b>center</b> <b>left</b> <b>right</b> <b>justify</b>	Uitlijning. Justify is het zowel links als rechts uitlijnen.
<b>text-indent</b>	<b>grootte</b> <b>percentage</b>	Laat de eerste regel van een tekst inspringen. Met een negatieve waarde zal de eerste regel uitspringen.
<b>line-height</b>	<b>normal</b> <b>getal</b> <b>grootte</b> <b>percentage</b>	Regelhoogte. Omdat er witruimte tussen regels zit, is <b>normal</b> gewoonlijk niet gelijk aan 100%, maar aan bijvoorbeeld 120%.

Hieronder staat een voorbeeld style sheet en een voorbeeld van de toepassing van de style sheet.

```

p {
    text-align: justify;
    text-indent: 4em;
    line-height: 170%;
}

```

Dit is een voorbeeldtekstje dat laat zien dat u met cascading style sheets typografische mogelijkheden aan HTML-pagina's kunt toevoegen die eerder niet mogelijk waren. De opmaakmogelijkheden

van een HTML-pagina komen zo steeds dichterbij in de buurt van de opmaakmogelijkheden van een DTP-programma. Misschien komt het nog zo ver dat documentformaten als PDF door CSS overbodig worden.

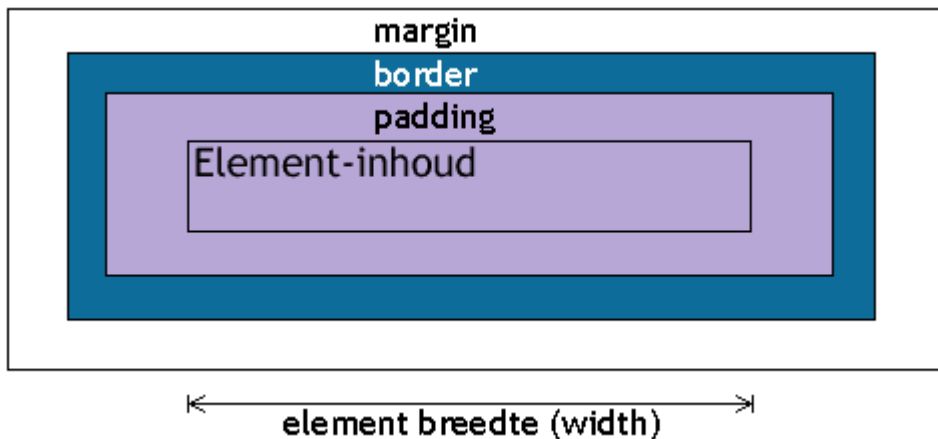
## Box kenmerken

De verschillende box-kenmerken op een rijtje:

Eigenschap	Waarde	Betekenis
<b>margin (-top, -right, -bottom, -left)</b>	<b>auto</b> <i>grootte</i> percentage	De marge. Zie hieronder.
<b>padding (-top, -right, -bottom, -left)</b>	<i>grootte</i> percentage	De opvulling. Zie hieronder.
<b>border (-width, -top-width, -right-width, -bottom-width, -left-width)</b>	<b>medium</b> <b>thin</b> <b>thick</b> <i>grootte</i>	De randdikte, zie de uitleg hieronder.
<b>border-color</b>	<i>kleur</i>	De randkleur.
<b>border-style</b>	<b>none</b> <b>dotted</b> <b>dashed</b> <b>solid</b> <b>double</b> <b>groove</b> <b>ridge</b> <b>inset</b> <b>outset</b>	De randstijl, zie de uitleg hieronder.
<b>border (-top, -right, -bottom, -left)</b>	Meerdere eigenschappen achter elkaar: width style color	
<b>width</b>	<i>grootte</i>	Breedte van de box
<b>height</b>	<i>grootte</i>	Hoogte van de box
<b>float</b>	<b>none</b> <b>left</b> <b>right</b>	Zie de tekst onder het kopje <i>Float en clear</i> .
<b>clear</b>	<b>none</b> <b>left</b> <b>right</b> <b>both</b>	

Elk HTML-element bevindt zich in een denkbeeldige box. Deze box heeft een marge

(margin), een rand (border) en een opvulling (padding). Van deze drie kenmerken kan de 'dikte' voor alle vier de zijden worden bepaald.



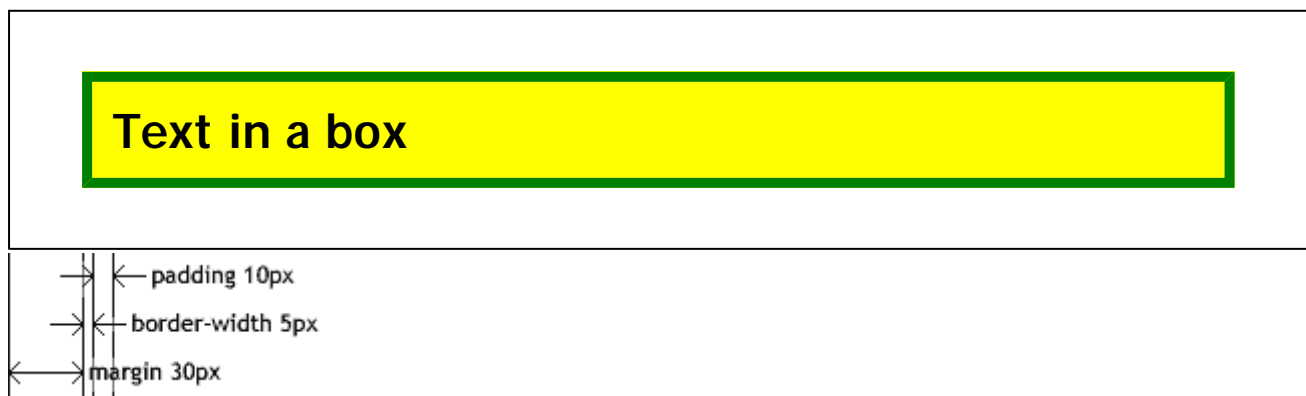
Als voorbeeld nemen we een h2-kop, waarvan we de opvulling 10 pixels maken, de rand 5 pixels dik maken en een marge aan alle kanten nemen van 30 pixels. Voor de duidelijkheid maken we ook nog de achtergrond geel en de rand groen.

```
h2 {
  background-color: yellow;
  padding: 10px;
  border-width: 5px;
  border-style: solid;
  border-color: green;
  margin: 30px;
}
```

Wat opvalt is dat een border-style moet worden opgegeven om een rand zichtbaar te maken.

```
<h2>Text in a box</h2>
```

wordt dan:



Duidelijk is dat de rand 5 pixels dik is, de afstand van de tekst tot de rand 10 pixels is en de ruimte om de box 30 pixels groot is. Het dunne zwarte lijntje is overigens geen onderdeel van deze style sheet, maar is toegevoegd om de marge te illustreren.

Stel nu dat we niet boven, onder, links en rechts dezelfde marges willen. We willen bijvoorbeeld boven 10 pixels, links 20 pixels, rechts 30 pixels en onder 50 pixels. In CSS wordt dit dan:

```
h2 {
  background-color: yellow;
  padding: 10px;
  border-width: 5px;
  border-style: solid;
  border-color: green;
  margin-top: 10px;
  margin-left: 20px;
  margin-right: 30px;
  margin-bottom: 50px;
}
```

Dit kan echter veel korter geschreven worden:

```
h2 {
  background-color: yellow;
  padding: 10px;
  border-width: 5px;
  border-style: solid;
  border-color: green;
  margin: 10px 30px 50px 20px;
}
```

Achter margin kunnen we gelijk de vier gewenste waarden zetten. De volgorde is natuurlijk wel belangrijk: top, right, bottom, left (boven, rechts, onder, links). Ook is het belangrijk dat tussen de waarden geen komma's geplaatst mogen worden.

Hoe dan ook, het resultaat is:



**Text in a box**

Vaak is de marge links en rechts gelijk en ook onder en boven. We kunnen dan eerst de verticale en dan de horizontale marge opgeven, bijvoorbeeld:

```
margin: 10px 20px;
```

Zijn alle marges gelijk, dan volstaat een enkele waarde. Zie het eerste voorbeeld van box-kenmerken.

Al deze verschillende notaties gelden natuurlijk ook voor padding en border.

Omdat border meerdere kenmerken heeft (width, style en color), werkt dit bij border iets anders. Achter border kunnen we gelijk de drie kenmerken opgeven. Het bovenstaande voorbeeld wordt dan:

```
h2 {
  background-color: yellow;
  padding: 10px;
  border: 5px solid green;
  margin: 10px 30px 50px 20px;
}
```

We kunnen ook eigenschappen voor alleen de bovenrand van de box opgeven:

```
border-top: 5px solid green;
```

Willen we, net als bij margin en padding mogelijk is, vier diktes in één keer opgeven, dan gebruiken we border-width:

```
border-width: 10px 10px 20px 30px;
```

In het voorbeeld hierboven gebruikten we `border-style: solid`. Er zijn echter nog veel meer verschillende randstijlen:



Met de kenmerken width en height kunnen we de breedte en hoogte van een box bepalen.

Voorbeeld:

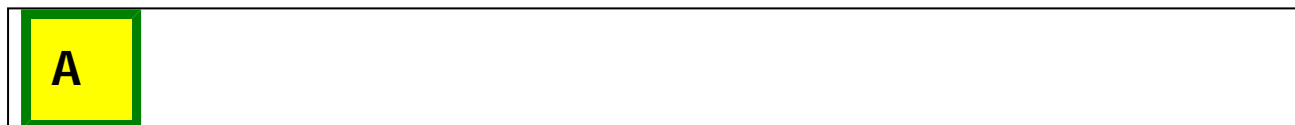
```
h2 {
  background-color: yellow;
  padding: 10px;
  border: 5px solid green;
  margin: 10px;
  width: 30px;
  height: 30px;
}
```

De HTML-code

```
<h2>A</h2>
```



wordt dan:



Het h2-kopje heeft nu niet alleen een achtergrondkleur en een rand, maar de inhoud heeft nu ook een grootte van 30 bij 30 pixels.

De width en height bepalen de grootte van de inhoud van de box, niet van de box zelf. **De diktes van de padding, border en margin komen dus bovenop de width en height!** Anders gezegd: de width en height is **exclusief** de padding, border en margin.

Er is een probleem: Internet Explorer 5.5 en eerder houdt zich niet aan deze specificatie en probeert de complete box in de width en height te stoppen. Explorer 6, Netscape 6 en Mozilla houden zich wel aan de specificatie.

Hieronder een voorbeeld van het probleem. Internet Explorer 5.5 trekt de border van het achtergrondplaatje af, zoals het niet hoort, waardoor de meeuw zijn kop en poten verliest. Internet Explorer 6, Netscape 6 en Mozilla laten wel het hele achtergrondplaatje zien.



## Float en clear

Block-elementen zoals `p`, `table` en `h2` worden gewoonlijk onder elkaar geplaatst. Met float (Engels voor drijven) kunnen we block-elementen naast elkaar plaatsen en zelfs tekst om een element laten heenlopen. Met clear kunnen we dat weer teniet doen.

Float is vergelijkbaar met het `align`-attribuut van bijvoorbeeld de `<img>`-tag. Clear is vergelijkbaar met `<br clear="...">`.

Voorbeeld:

```
<style type="text/css">
h1 {
    width: 5em;
    float: left;
    margin: 0;
}
img {
    float: right;
```

```


        padding-left: 3px;
    }
    p {
        margin-top: 0;
    }
</style>

<h1>De maan</h1>

<p>Elke 29,5 dagen draait de maan een rondje om de aarde.
We kunnen de maand zien omdat deze door de zon wordt verlicht.
Maar heel soms valt de maan in de schaduw van de aarde.
De maan krijgt dan geen zonlicht meer en wordt verduisterd.
We spreken dan van een maansverduistering.
Als we tijdens een maansverduistering op de maan zouden staan,
zouden we getuige zijn van een zonsverduistering!</p>

```

Het resultaat is een h1-kop met een vaste breedte, drijvend aan de linkerkant, een foto drijvend aan de rechterkant en de tekst die daartussenin loopt:

<h2 style="margin: 0;">De maan</h2> <p style="margin: 0;">Elke 29,5 dagen draait de maan een rondje om de aarde. We kunnen de maand zien omdat deze door de zon wordt verlicht. Maar heel soms valt de maan in de schaduw van de aarde. De maan krijgt dan geen zonlicht meer en wordt verduisterd. We spreken dan van een maansverduistering. Als we tijdens een maansverduistering op de maan zouden staan, zouden we getuige zijn van een zonsverduistering!</p>	
---	---

Het volgende voorbeeld toont het gebruik van clear. De bedoeling is dat de list niet in de tekstomloop (naast de foto) wordt opgenomen, maar onder de voorgaande code.

```


<style type="text/css">
img {
    float: left;
    padding-right: 5px;
}
p {
    margin-top: 0;
}
ul {
    clear: both;
}
</style>


<p>De maan draait rond de aarde en de aarde draait weer rond de zon.
Een maansverduistering treed op als deze drie hemellichamen precies in een
lijn staan met de aarde in het midden.</p>

<ul>
<li><a href="http://www.mira.be/articles/sterrenkunde/maansverduistering/">Maansv
<li><a href="http://www.knmi.nl/voorl/nader/maanenweer.htm">Maansverduisteringen
</ul>

```

Het resultaat, met de list *onder* de foto. De float van het img-element wordt hier opgeheven:

	<p>De maan draait rond de aarde en de aarde draait weer rond de zon. Een maansverduistering treed op als deze drie hemellichamen precies in een lijn staan met de aarde in het midden.</p>
<ul style="list-style-type: none"> <li>• <a href="#">Maansverduisteringen</a></li> <li>• <a href="#">Maansverduisteringen en het weer</a></li> </ul>	

## Classificatie kenmerken

Dit is een categorie van niet echt verwante kenmerken: `display`, `white-space` en `list-kenmerken`.

### display

Display kent vier verschillende waarden:

Eigenschap	Waarde	Betekenis
<code>display</code>	<code>none</code> <code>inline</code> <code>block</code> <code>list-item</code>	Zie hieronder.

De verschillende waarden hebben de volgende betekenissen:

<code>none</code>	Het element wordt niet weergegeven
<code>inline</code>	Het element wordt in de 'tekststroom' opgenomen, bijvoorbeeld vette tekst.
<code>block</code>	Het element komt los van de omliggende tekst te staan, bijvoorbeeld kopjes en tabellen
<code>list-item</code>	Het element is een list-item, gewoonlijk <code>&lt;ul&gt;</code> , <code>&lt;ol&gt;</code> en <code>&lt;dl&gt;</code>

Voorbeeld van block-elementen zijn de `h1`, `p` en `hr`-tags. Alle tekst binnen bijvoorbeeld `<h1>` en `</h1>` komt los van de omliggende tekst te staan. Voorbeeld van inline elementen zijn de `b`, `a` en `tt`-tags. Alles tussen bijvoorbeeld `<b>` en `</b>` wordt in de omliggende tekst opgenomen.

Ter illustratie gebruiken we een tekst met een vet gedeelte.

Vette tekst heeft als `display`-eigenschap `inline`. Voor de duidelijkheid wordt dit toch in de volgende style sheet opgenomen, samen met een blauwe kleur:

```
b {
    display: inline;
    background-color: turquoise;
}
```

De voorbeeldtekst ziet er als volgt uit. Niets bijzonders eigenlijk.

Seneca Lucilio suo salutem. **Epistulas ad me perferendas tradidisti, ut scribes, amico tuo; deinde admones me, ne omnia cum eo ad te pertinentia communicem, quia non soleas ne ipse quidem id facere.** Ita eadem epistula illum et dixisti amicum et negasti. Sed si aliquem amicus existimas, cui non tandundem credis quantum tibi, vehementer erras et non satis nosti vim verae amicitiae.

Door vette tekst nu als `display`-eigenschap `block` te geven, wordt het verschil met `inline` duidelijk.

```
b {
    display: block;
    background-color: turquoise;
}
```

De voorbeeldtekst ziet er nu zo uit:

Seneca Lucilio suo salutem.  
**Epistulas ad me perferendas tradidisti, ut scribes, amico tuo; deinde admones me, ne omnia cum eo ad te pertinentia communicem, quia non soleas ne ipse quidem id facere.**  
 Ita eadem epistula illum et dixisti amicum et negasti. Sed si aliquem amicus existimas, cui non tandundem credis quantum tibi, vehementer erras et non satis nosti vim verae amicitiae.

Een element met als `display inline` wordt dus in de omliggende tekst opgenomen en een element met `block` komt los van de omliggende tekst te staan, in de vorm van een rechthoek.

## white-space

Eigenschap	Waarde	Betekenis
<code>white-space</code>	<code>normal</code> <code>nowrap</code> <code>pre</code>	Zie hieronder.

De verschillende waarden hebben de volgende betekenissen:

<b>normal</b>	Normal is normale HTML-formattering.
<b>nowrap</b>	Nowrap zorgt dat tekst niet wordt afgebroken, net als bijvoorbeeld de <code>nowrap=yes</code> in een <code>&lt;td&gt;</code> -tag.
<b>pre</b>	Pre geeft hetzelfde resultaat als het plaatsen tussen <code>&lt;pre&gt;</code> en <code>&lt;/pre&gt;</code> -tags: spaties en geregeleides worden zonder wijziging getoond.

Voorbeeld van pre:

```
<style type="text/css">
h4 {white-space: pre}
</style>

<h4>      Deze zomer
wordt heel   erg   mooi</h4>
```

Wordt:

```
      Deze zomer
erg
rg  mooi
```

ndersteund.

## list-style

Eigenschap	Waarde	Betekenis
<b>list-style-type</b>	circle disc square decimal lower-alpha lower-roman upper-alpha upper-roman none	Geeft aan wat voor elk list-item wordt getoond.
<b>list-style-image</b>	none url(path)	Geef de lokatie van een plaatje op dat voor elk list-item wordt getoond.

<b>list-style-position</b>	<b>inside</b> <b>outside</b>	Outside is de gebruikelijke, ingesprongen weergave, bij inside wordt niet ingesprongen.
<b>list-style</b>	Meerdere eigenschappen achter elkaar: list-style-type list-style-position list-style-image	

Voorbeeld:

```
ol { list-style: lower-alpha inside }
```

Geeft (met de betekenis van *hamster* uit de [Jargon File](#)):

- a. [Fairchild] A particularly slick little piece of code that does one thing well; a small, self-contained hack. The image is of a hamster {happily} spinning its exercise wheel.
- b. A tailless mouse; that is, one with an infrared link to a receiver on the machine, as opposed to the conventional cable.
- c. [UK] Any item of hardware made by Amstrad, a company famous for its cheap plastic PC-almost-compatibles.

Een voorbeeld van het gebruik van list-style-image:

```
ol {list-style-image: url(pijltje.gif)}
```

Geeft:

- ➔Gebruik geen hoofdletters in de subjectregel van een e-mail
- ➔Houd de subjectregel zo kort mogelijk

Let op: de URL in `list-style-image` staat **niet** tussen aanhalingstekens, maar **wel** tussen `url( en )`.

## Kleuren

Tot nu toe hebben we als kleur steeds de naam van een kleur gebruikt, zoals orange en green. Natuurlijk kunnen we, net als in HTML, de kleur ook als RGB-waarde opgeven. Daar zijn drie verschillende notaties voor:

```
color: #00FF00
color: rgb(0, 255, 0)
color: rgb(0%, 100%, 0%)
```

Deze drie voorbeelden hebben dezelfde groene kleur als resultaat. Het eerste voorbeeld is de hexadecimale kleurnotatie, dezelfde als in HTML, het tweede en derde voorbeeld geven de rood/groen/blauw-waarden apert op: eerst steeds als een waarde tussen 0 en 255, daarna als een percentage.

## Grootte eenheden

Tot nu tot werd als grootte-eenheid steeds de pixel gebruikt, als in `padding: 10px`. Naast de pixel zijn nog een aantal manieren om de grootte op te geven.

<b>px</b>	Aantal pixels Voorbeeld <code>height: 20px</code> : 
<b>mm, cm</b>	Aantal millimeters of centimeters Voorbeeld <code>height: 1cm</code> : 
<b>em, ex</b>	Aantal ems of exes. De em is de hoogte van het huidige lettertype en ex is de hoogte van de huidige onderkast 'x' Voorbeeld <code>height: 1ex</code> : 
<b>pt, pi</b>	Aantal points of picas. Een point is 1/72 van een inch en een pica is 12 points. Deze termen uit de drukkerswereld kunnen we ook in webpagina's gebruiken Voorbeeld <code>height: 12pt</code> : 

Hoewel geen grootte, kunnen we in plaats van een grootte vaak ook een percentage opgeven.

Voorbeeld:

```
font-size: 120%
```

Geeft:

Voorbeeldtekst

Tegenwoordig hebben veel browsers de mogelijkheid om de tekst te vergroten of te verkleinen. Dit is zeer handig voor slechtzienden. Dit heeft echter geen invloed op tekst waarvan de grootte in pixels of millimeters is opgegeven. Om tekst te laten meegroeien

of -krimpen, dient de grootte in punten of een percentage opgegeven te worden. Bijvoorbeeld 150% voor grote tekst of 80% voor kleine tekst.

## Formulieren

Met style sheets is het niet alleen mogelijk om het uiterlijk van tekst te veranderen, maar ook van formulieren.

In dit voorbeeld gaan we uit van het volgende formulier:

```
<form action="http://www.google.com/search">
<input name="q" type="text" value="dhtml" size="40">
<input type="submit" value="zoek">
</form>
```

Dit geeft gewoonlijk het volgende 'formulier':



dhtml      zoek

Nu zullen we aan het element `<input>` de volgende style sheet koppelen:

```
input {
  border: 2px solid navy;
  font-family: "Comic Sans MS", Arial, Helvetica, sans-serif;
  font-size: 12pt;
  color: navy;
  background-color: cornsilk;
}
```

Het resulterende formulier is nu zachtgeel met blauwe tekst:



dhtml      zoek

Het gebruik van style sheets met formulieren werkt niet in elke browser even goed. Netscape 4 verandert alleen het lettertype en Netscape 6 verandert de voor- en achtergrondkleur. Internet Explorer 5 past de hele style sheet toe.

## Groeperen

Het komt vaak voor dat we meerdere elementen van dezelfde stijl willen voorzien. Bijvoorbeeld alle kopjes willen we blauw hebben. We kunnen dan de koppen groeperen. De style sheet wordt dan als volgt:



```
h1, h2, h3, h4, h5, h6 {color: blue}
```

## Contextuele selectie

Een andere interessante mogelijkheid is om een stijl te bepalen voor elementen in een ander element. We kunnen bijvoorbeeld aangeven dat een cursieve tekst in een h2-kop niet cursief, maar rood moet worden. We schrijven dan de *i* achter de h2, zonder komma.

```
<style type="text/css">
h2 i {color: red; font-style: normal}
</style>

<h2>Dit is een <i>knappe</i> kop</h2>
<i>Met een cursieve tekst.</i>
```

Geeft dit als resultaat:

**Dit is een knappe kop**

*Met een cursieve tekst.*

Een dergelijke constructie is handig om bijvoorbeeld de stijl van een lijst binnen een lijst te bepalen (een ul in een ul, dus).

Let op: bij groeperen worden de elementen door komma's gescheiden, bij contextuele selectie niet.

## Classes

Tot nu toe hebben we steeds gezien hoe de stijl van een HTML-tag kan worden veranderd. Maar wat als we <p> in meerdere stijl-varianten willen hebben? Bijvoorbeeld een zwarte op een witte achtergrond, maar ook een witte voor op een blauwe achtergrond? Of een speciale, opvallende variant?

Dit is mogelijk door gebruik te maken van classes. Als voorbeeld voegen we aan <h2> en <p> een zelfbedachte class 'opvallend' toe, die de tekst rood en cursief maakt.

In de style sheet schrijven we dan `p.opvallend {...}` in plaats van `p {...}`. In de HTML schrijven we dan `<p class="opvallend">` in plaats van `<p>`.

```
<style type="text/css">
h2.opvallend {color: red; font-style: italic}
p.opvallend {color: red; font-style: italic}
</style>
```

```
<h2>Normaal</h2>
<p>Dit is normale tekst.</p>

<h2 class="opvallend">Opvallend</h2>
<p class="opvallend">Dit is opvallende tekst.</p>
```

Dit wordt in de browser:

## Normaal

Dit is normale tekst.

## *Opvallend*

*Dit is opvallende tekst.*

Een class hoeft niet per se aan een element gekoppeld te zijn. In de style sheet schrijft men dan gelijk de punt en class-naam. De twee regels in de style sheet hierboven hadden we dus ook kunnen schrijven als:

```
.opvallend {color: red; font-style: italic}
```

De style sheet geldt dan voor alle elementen met als attribuut `class="opvallend"`.

## Meerdere classes opgeven

Het is mogelijk om in het class-attribuut meerdere classnamen op te nemen. De classnamen dienen gescheiden te worden door een spatie.

```
<style type="text/css">
.opvallend {color: red}
.groot {font-size: large}
</style>

<p class="opvallend groot">Opvallend en groot</p>
```

Dit wordt in de browser:

*Opvallend en groot*

Het is duidelijk dat de tekst zowel de eigenschappen van de *opvallend*-class als van de

*groot*-class overneemt.

## Pseudo-classes

In HTML kunnen we met `<body link=orange vlink=red>` de kleuren vastleggen van de hyperlinks en bezochte hyperlinks. We weten inmiddels dat we met `a {color:orange}` de hyperlinks oranje kunnen maken. Maar hoe veranderen we de kleur van een bezochte hyperlink?

Met pseudo-classes is dat mogelijk. Achter de `a` schrijven we een dubbele punt en daarachter de naam van de pseudo-class. Naast een pseudo-class voor bezochte hyperlinks zijn er ook pseudo-classes voor het klikken op een link en voor een mouse-over.

<b>a:link</b>	Gewone (onbezochte) hyperlink
<b>a:visited</b>	Bezochte hyperlink
<b>a:active</b>	Actieve hyperlink (er wordt op geklikt)
<b>a:hover</b>	Hover hyperlink (de muis beweegt zich boven de link). <b>Hover</b> hoort eigenlijk bij CSS2.

Een voorbeeld:

```
<style type="text/css">
a {color: orange}
a:visited {color: red}
a:active {color: blue; font-weight: bold}
a:hover {background-color: lightgrey}
</style>

Ik zoek met <a href="http://www.google.com/linux">Google</a>.
```

Wordt:

Ik zoek met [Google](#).

Er zijn ook twee typografische pseudo-classes:

<b>P:first-line</b>	Eerste regel van een tekst.
<b>P:first-letter</b>	Eerste letter van een tekst.

Typografische pseudo-classes kunnen we ook op andere elementen dan `<p>` toepassen.

## Id selecteren

Soms willen we de stijl van één enkel element wijzigen. Dit is mogelijk door dat element een 'id' te geven. In de style sheet verwijzen we naar dit element met een hekje (#). Een voorbeeld:

```
<style type="text/css">
ul {border: 1px solid orange}
#nummer1 {color: purple}
</style>

<ul>
<li id="nummer1">Internet Explorer 5 (76%)
<li>Netscape 4 (8%)
<li>AOL (6%)
</ul>
```

Geeft dit resultaat:

- Internet Explorer 5 (76%)
- Netscape 4 (8%)
- AOL (6%)

Elke HTML-tag kunnen we op deze manier een id geven. Id's dienen wel altijd uniek te zijn, geen twee tags kunnen dezelfde id hebben. Het is daarom niet nodig om in de bovenstaande style sheet `li#nummer1` te schrijven.

## Het style-attribuut

Het mooie van style sheets is dat inhoud en vorm van elkaar worden gescheiden. Speciaal voor de mensen die dat onzin vinden, is het style-attribuut uitgevonden. Daarmee kan de stijl direct in de HTML-tag worden opgenomen. Het bovenstaande voorbeeld had volgens deze zondaars net zo goed geschreven kunnen worden als:

```
<ul style="border: 1px solid orange">
<li style="color: purple">Internet Explorer 5 (76%)
<li>Netscape 4 (8%)
<li>AOL (6%)
</ul>
```

## Span en div

Met span en div is het mogelijk om een heel stuk HTML van een stijl te voorzien. Zo kunnen we bijvoorbeeld een kopje en een paragraaf in een blauwe box zetten:

```
<style type="text/css">
div.blauw {border: 3px solid aqua; background-color: aquamarine}
</style>

<div class="blauw">
<h2>Yet another limeric</h2>
<p>
There once was a man from the stix,<br>
Who liked to write limerics.<br>
But he failed at the sport,<br>
Because he wrote them too short.
</p>
</div>
```

Wordt:

## Yet another limeric

There once was a man from the stix,  
Who liked to write limerics.  
But he failed at the sport,  
Because he wrote them too short.

Maar waarom kent HTML `div` én `span`? `Div` heeft als stijl `display: block` en `span` heeft als stijl `display: inline`. Gebruik `div` voor elementen die losstaan van de omliggende tekst, net als `<h2>` en `<table>`. Gebruik `span` voor elementen binnen een tekst, net als `<b>` en `<a...>`.

Zie [display](#) voor een uitgebreide uitleg.

## Commentaar

In HTML kunnen we commentaar tussen `<!--` en `-->` zetten, maar dat werkt niet in style sheets. In style sheets zetten we commentaar tussen `/*` en `*/`, net als in de programmeertaal C.

Een voorbeeld van commentaar:

```
h2 { /* definitie sub-kopje */
    background-color: yellow;
    padding: 10px; /* ruime padding is belangrijk */
    border: 5px solid green;
    margin: 10px 30px 50px 20px;
}
```

## Import

Eén van de handige eigenschappen van CSS is dat de style sheet ook in een apart bestand kan staan. Op deze manier kunnen alle pagina's van een website gebruik maken van dezelfde style sheet. Een kleurtje of maatje hoeft dan nog maar in één bestand te worden gewijzigd en de hele site verandert mee. Een voorbeeld:

Het bestand stijl.css:

```
body {background-color: yellow}
h2 {font-family: arial, helvetica, sans-serif}
```

Het HTML-document:

```
<html>
<head>
<title>Mijn document</title>
<link rel="stylesheet" type="text/css" href="/styles/stijl.css">
</head>
<body>

... de tekst van het document ...

</body>
</html>
```

Meerdere style sheets achter elkaar laden is ook mogelijk. Zo kunnen we bijvoorbeeld een style sheet laden die over de hele website wordt gebruikt en een style sheet voor een gedeelte van de website, bijvoorbeeld een discussie forum.

Een style sheet vanuit een andere style sheet laden is ook mogelijk. Om vanuit stijl.css het bestand inhoud.css te laden wordt `@import` gebruikt:

```
@import url(/styles/inhoud.css);
body {background-color: yellow}
h2 {font-family: arial, helvetica, sans-serif}
```

## !important

Soms komt het voor dat we een style sheet hebben, waarvan eigenschappen later worden 'overschreven', bijvoorbeeld bij het importeren van een andere style sheet.

Dit is te veranderen door de eerdere eigenschap belangrijker te maken. Dit doen we door achter de eigenschap `!important` te zetten.

Bijvoorbeeld:

```
p { font-size: 20pt !important } /* Paragraaftekst altijd in 20-punts letter */
```

Paragraaftekst is nu altijd 20 punts tekst, zelfs als later een andere grootte wordt gespecificeerd.

## DOCTYPE SGML-declaratie

Om style sheets goed weer te geven, moesten de browsermakers soms de manier veranderen waarop browsers pagina's weergegeven. Dit zou weer tot gevolg hebben dat de browsers al bestaande pagina's ineens anders (en soms verkeerd) weergegeven.

Om dit probleem op te lossen, is de 'standards mode' en 'compatibility mode' bedacht. In standards mode werkt de browser volgens de standaard en in compatibility mode werkt de browser op de oude, niet-standaard manier.

Standards mode is in te schakelen door bovenin de HTML-pagina een SGML-declaratie op te nemen. Is deze niet aanwezig, dan schakelt de browser over naar compatibility mode. Dit principe werkt in Internet Explorer 6, Netscape 6 en Mozilla.

De volgende SGML-declaratie op de eerste regel van een HTML-document schakelt standards mode in:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Er zijn nog meer doctypes die standards mode inschakelen. Microsoft heeft een lijst gemaakt bij welke doctypes Internet Explorer naar de standards mode overschakelt.

- [HTML and DHTML Reference: !doctype](#)

## Browsers en CSS-ondersteuning

Cascading Style Sheets bestaan nu al meer dan zes jaar. De meest recente browsers ondersteunen style sheets vrij goed. Voorbeelden daarvan zijn:

- [Microsoft Internet Explorer](#)
- [Firefox](#)
- [Mozilla](#)
- [Netscape 6](#)

Netscape 4 ondersteunt CSS maar voor een gedeelte. Dat was altijd erg jammer, want soms zagen pagina's met geldige style sheets er in Netscape 4 mismaakt uit. Hierdoor waren websitebouwers nooit gretig om style sheets te gebruiken.

Nu wordt Netscape 4 nog amper gebruikt en is het nauwelijks nog een argument om geen style sheets te gebruiken.

Het door een kleine groep geroemde [Opera](#) ondersteunt style sheets ook nog niet volledig, hoewel het met elke versie wel een stuk beter wordt.

Bij het maken van pagina's met style sheets zullen we voorlopig onze creaties in meerdere

browsers moeten testen. Om precies te zien wat wel en wat niet in welke browsers ondersteund wordt, heeft Eric Meyer een erg handig overzicht gemaakt:

- [Master Compatibility Chart](#)

## Cascading Style Sheets, level 2 (CSS2)

Sinds mei 1998 bestaat er een opvolger van CSS1, namelijk CSS2. De belangrijkste toevoegingen zijn:

- Style sheets voor verschillende media, zoals TV, monitor en printer.
- Style sheets voor gesproken documenten voor blinden, waarmee aangegeven kan worden hoe een tekst moet worden uitgesproken.
- Betere ondersteuning voor verschillende schriften, bijvoorbeeld Arabisch en Chinees.

Browsers beginnen nu ook CSS2 steeds beter te ondersteunen. Alleen Internet Explorer ondersteunt nog erg weinig van deze specificatie. Daardoor is het nog steeds weinig aantrekkelijk om CSS 2 te gebruiken. Dat is erg jammer. De specificatie van CSS2 is ook bij het W3C te vinden:

- [Cascading Style Sheets, level 2](#)