



Opleiding FullStack - CURSUS CSS



Colofon

Titel: CSS

Auteurs: De Nittis Massimo

Uitgever: SyntraPXL vzw

Editor: De Nittis Massimo

© SyntraPXL 2023

Niets uit deze uitgave mag vermenigvuldigd worden en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze dan ook zonder voorafgaandelijke toestemming van de uitgever.

Inleiding

CSS staat voor Cascading Style Sheets. De 3 mogelijkheden:

- Inline Style (binnen (X)HTML element)
- Internal Style Sheet (binnen de <head> tag)
- External Style Sheet (apart document)

De Inline style wordt nauwelijks toegepast en is ook niet overzichtelijk, dus beperk het gebruik hiervan.

Diverse voordelen

Je kan de opmaak en vormgeving vanuit één bestand regelen voor de hele website. Je hoeft alleen maar veranderingen aan te brengen in dat ene bestand om een wijziging voor de hele website door te voeren.

Intern Stylesheet (Stijlblad)

Bij de interne methode plaatsen we de CSS code binnen de <head></head> tags van elk (X)HTML bestand. Deze methode wordt ook wel embedded stylesheet genoemd (ingesloten stijlblad). Een voorbeeld..

```
<head>
<title><title>
<style type="text/css">| ← begin regel intern stijlblad
h1 { color: red; font-size:24px;} ← CSS code
</style> ← sluitregel intern stijlblad
</head>
<body>
```

Extern Stylesheet (stijlblad)

Indien de CSS code kan gelden voor meerdere bestanden dan is een externstijlblad aan te bevelen. De onderstaande regel plaatsen we tussen de <head> tags en via het link element wordt er een koppeling gelegd naar het externe stijlblad

EXTERN STIJLBLAD

```
<link rel="stylesheet" type="text/css" href="css/stijl.css" />
```

← de map en naam van het css bestand

Welke manier aan te bevelen is, hangt af van wat je wilt doen. Als je slechts 1 bestand hebt waar de opmaak voor nodig is dan maken we gebruik van plaatsing ervan binnen de <head></head> tags (interne). Als je van plan bent om de css voor meerdere bestanden te gebruiken, dan is het aan te bevelen en ook meer overzichtelijk om voor de externe bestand methode te kiezen.

Inline style

Deze stijl alleen gebruiken voor een incidentele en individuele afwijking van een element. In HTML bestaat een attribuut style, welke we op een html-pagina kunnen toepassen. Maar gelden niet meer de regels van css. Dus we verliezen het doel uit het oog om een "centrale opmaak" CSS, te hebben. Ik geef een voorbeeld, maar raad toch sterk af hiervan veel gebruik te maken.

Voorbeeld in de html-pagina:

```
<p style="color:green;">
```

Hier geen selector, geen accolades. Wel zie je tussen de " " een css regel.

Voorbeeld:

```
<head>
<style type="text/css">
p{background:#e4e4e4; color:green;}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Naamloos document</title>
</head>
<body>
<p style="background: blue; color: white;">Een achtergrond en tekstkleur via inline css
<p style="color:green;">deze kleur is groen</p> zonder p is de tekst zwart
</body>
</html>
```

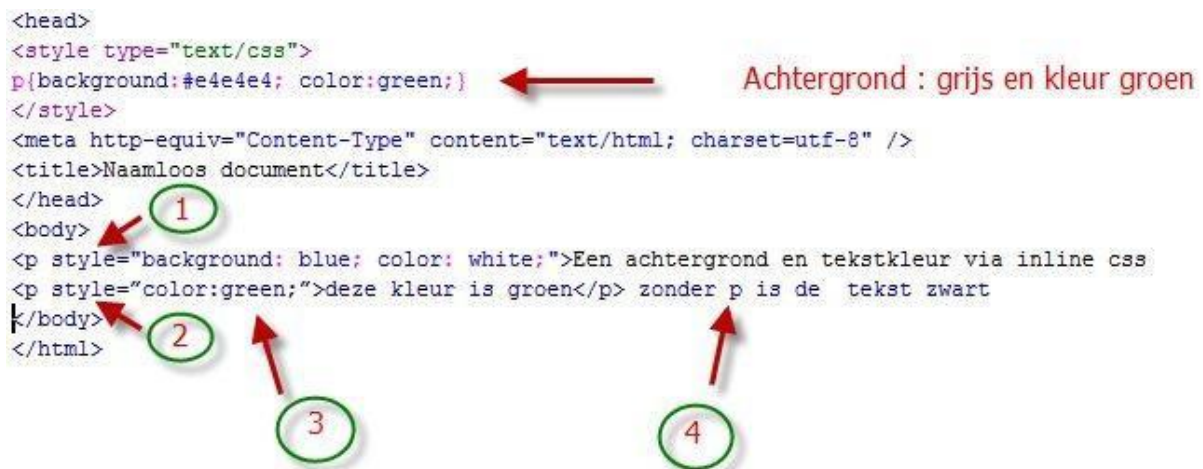
Achtergrond : grijs en kleur groen

1

2

3

4



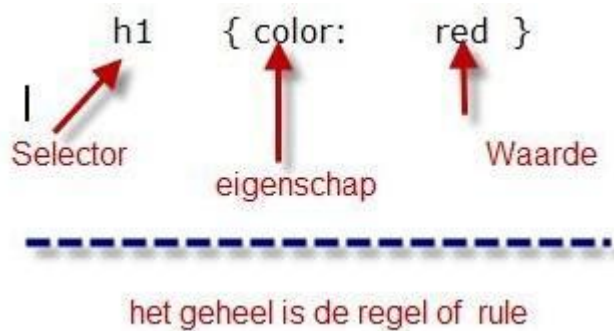
Achtergrond e4e4e4 (=grijs) en tekstkleur groen

Via <p style> wordt de achtergrond blauw en de tekstkleur wit.
Zonder <p> wordt de default kleur van html (zwart) ingeschakeld.

Selectors

3 soorten Selectors:

- ★ tag selector
- ★ class selector
- ★ id selector (pseudo class)



De eigenschap is het attribuut dat steeds veranderd kan worden en de waarde, is de waarde van de eigenschap. De combinatie van eigenschap en waarde wordt een declaratie genoemd.

```
h1 {  
  color: red; /* kleur is rood */  
  text-align: center;  
}
```

h1, in rood en gecentreerd

Toelichting:

h1, is de selector, die zegt over welk element we het hebben; color, is een eigenschap, die iets over het element zegt; red, de waarde van color; /* Rood */, commentaar.

In een stijlblad mag men naar eigen goeddunken spaties en lege regels toevoegen. Dus het is niet belangrijk waar het staat, maar wel hoe het er staat. Dus daarom zijn de te gebruiken leestekens van groot belang.

Zoals:

```
{ accolade openen  
-      het streepje tussen text-align  
: de dubbele punt tussen de eigenschap en de waarde  
; de puntkomma aan het einde van de waarde  
} en de sluit accolade
```

Let op:

Elke selector kan meerdere eigenschappen hebben en elke eigenschap binnen die selector kan allerlei waardes bevatten.

Voorbeeld met meerdere stijl definities:

```
body {  
color: #e4e4e4; /* kleur grijs */  
background-color: blue;  
font-family: "Trebuchet MS", Verdana, Arial, serif;  
}
```

Waardes die uit meerdere woorden bestaan, worden tussen aanhalingstekens gezet. (zie bij trebuchet MS)

De zelfde stijlkenmerken toe kennen aan verschillende elementen:

```
h1,h2,p {  
font-weight: bold; color:#008080;  
}
```

Dus hierboven krijgen de selectoren h1,h2 en p alle drie een identieke kleur en de tekst worden vet weergegeven.

Inheritance (overerving)

Wanneer men een element binnen een ander plaatst (nesten heet dat), zal het geneste element de eigenschappen erven, toegekend aan het element. Tenzij u een andere waarde toewijst aan een eigenschap zie hieronder:

```
body { font-family: Verdana;}
```

Nu krijgt alle tekst binnen de body het font verdana. Als men bepaalde tekst een ander font wil geven kan dat als volg:

```
h1 {font-family: Arial, sans-serif;} p {font-family: Tahoma, serif;}
```

Nu krijgen alle <h1> tags binnen het bestand het font Arial en de <p> tags krijgen Tahoma, terwijl de font in andere elementen het eerdere font houden, die bij body is opgegeven { font:verdana.

Class

Veel html-elementen kunnen een Class krijgen. Het element krijgt dan de naam van een bepaalde klasse. De naam mag slechts een woord zijn zonder spatie. Met een class selector maak je een stijl die toegepast kan worden op elke HTML tag. De selector wordt dan als volgt geschreven: .naam van de class .(dus punt en naam van de klasse) Het class attribuut van de HTML tag wordt gebruikt om de stijl toe te kennen aan de tag: class="naam_class".

Uitgewerkt voorbeeld:

Bij 1 maak we gebruik van 2 classes n.l. eerste en tweede met resp. de kleuren blauw en rood.

Bij 2 zie je de html code. Beginnend met een normale tekst zonder class, dan is de kleur altijd zwart en daarna resp. verwijzend naar de classes eerste en tweede.

```
<style type="text/css">
p.eerste{ color: blue; }
p.tweede{ color: red; }
</style>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Naamloos document</title>
</head>
<body>
<p>Dit is de normale tekstkleur.</p>
<p class="eerste">Hier gebruikt gemaakt van de class: Eerste! met kleur blauw</p>
<p class="tweede">En nu van class: tweede met kleur rood!</p>
</body>
```





Dit is de normale tekstkleur.

Hier gebruikt gemaakt van de class: Eerste! met kleur blauw

En nu van class: tweede met kleur rood!

Als we nu al een kleur voor <p> gedefinieerd hebben in css. Bijv met

```
p {color:green;font-size:12px;}  
p.eerste{color:blue;}  
p.tweede{color:red;font-size:10px;}
```

Dan wordt <p> overal in de website standaard groen met een lettergrootte van 12px en niet meer in de zwarte kleur. De twee classes "eerste en tweede" geven uiteraard dezelfde kleuren als bij het vorige voorbeeld.

In bovenstaand voorbeeld hebben we 1 class gebruikt in de html , maar hieronder zie je dat je de classes ook kunt combineren tot 1 regel.

CSS definitie:

```
.class1{ color: black; }  
.class2{ color: red;}
```

Deze kan men als volgt combineren:

```
<p class="class1 class2">dit is een voorbeeld van classes met een zelfde belang.</p>
```

In het bovenstaande voorbeeld zal de link kleur rood zijn. Omdat de specificiteit van beide selectors dezelfde is. Met de volgende CSS classes zal de link kleur zwart zijn.

Css:

```
a.class1{ color: black; }  
.class2{ color: red; }
```

Html:

```
<a class='class2 class1' href='/'>link</a>
```


id selectors

Een ID selector en een Class geven beide de opmaak van een bepaald element in de CSS weer. Bij ID gebruiken we een **hekje #** voor de naam, terwijl dit bij class een punt is. Het grote verschil is, dat je een Class meerdere keren kunt gebruiken in de pagina en een ID uniek is en dus maar 1 keer gebruikt mag worden in een pagina.

Kenmerken

- Zeer gelijkend op class selectors.
- (Bijna) alle HTML tags hebben een id attribuut.
- De waarde van een id attribuut moet uniek zijn binnen een pagina. Eenzelfde id mag slechts 1 keer in een pagina voorkomen.
- Het verschil tussen class en id is dus dat class waarden onbeperkt mogen voorkomen in een pagina, en id waarden slechts 1 keer.
- id selectors worden gebruikt om speciale stukken in een document aan te duiden, bijvoorbeeld de navigatie.
- De selector wordt dan als volgt geschreven: `#waarde_id {css_eigenschap: waarde}`.

Wanneer kun je het beste ID gebruiken

Een id gebruik je als het gaat om een unieke naam die gekoppeld is aan hoofdelementen van een website. bijvoorbeeld `<div id="header">`, en zo ook voor: de navigatie, de inhoud, de zijkant, de footer en de zoekfunctie. Deze onderdelen komen maar een keer op een webpagina voor en krijgen daarom een unieke naam.

Div (afkorting voor division) verdeelt de inhoud in afzonderlijke secties. Elke sectie kan dan zijn eigen opmaak hebben, zoals gespecificeerd door de CSS. Div is een block-level container, wat betekent dat er een nieuwe regel komt na de `</div>`-tag .

Diverse soorten selectors

De Universal selector `*` geldt voor elk element. Als deze de enige selector is kan hij ook worden weggelaten.

```
*.html {margin:0;padding:0;}
```

Descendant selectors zijn een elegante manier om stijlen toe te passen op specifieke gebieden van uw pagina, terwijl het ook de noodzaak vermindert voor embedded classes binnen elementen.

```
div.nav ul {font-size:1.1em;}
```

Voor alle `ul` elementen binnen de `div` met een Class `.Nav`. krijgen de `font-size:1.1em`

Selectoren combineren

Verschillende selectors mogen gecombineerd worden.
Zo maak je regels die gelden in een specifieke context.

Voorbeeld combinatie id en class selector.

In CSS:

```
#navigatie .rand {  
    border: 10px solid #ff0;  
}
```

In HTML:

```
<div id="navigatie">  
<p class="rand">Hello</p>  
</div>
```



Enkel toegepast op elementen die opgeroepen worden via id navigatie en class rand!

Pseudo-classes

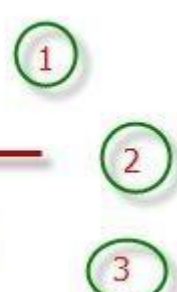
Pseudo classes worden gebruikt om een bepaald effect aan een selector mee te geven.

De bekendste pseudo-class is wel de `:hover` die zorgt ervoor dat de weergave van een element wijzigt, wanneer je er met de muisaanwijzer boven komt. Een Rollover effect.

Een pseudo-class selector is vergelijkbaar met een klasse in HTML, maar heeft als basis een dubbele punt gevolgd door de naam van de pseudo-class.

De Syntax: `selector:pseudo-class {eigenschap: waarde}`

```
div#menu a {  
    font-size: 12pt;  
    color: green;  
}  
  
div#menu a:hover {  
    font-size: 12pt;  
    color: yellow;  
    background-color: red;  
}
```



Meest voorkomende pseudo-selectors :

Onderstaande `a:links` (ankers) met hun resp. waardes

`a:link {color:#FF0000;}` /* niet bezochte link */ `a:visited`

`{color:#00FF00;}` /* Reeds bezochte link */ `a:hover`

`{color:#FF00FF;}` /* mouse over link */ `a:active`

`{color:#0000FF;}` /* geselecteerde link */

Minder gebruikte Pseudo classes zijn nog `:focus` , Geef de kleur van een input veld dat de focus krijgt,

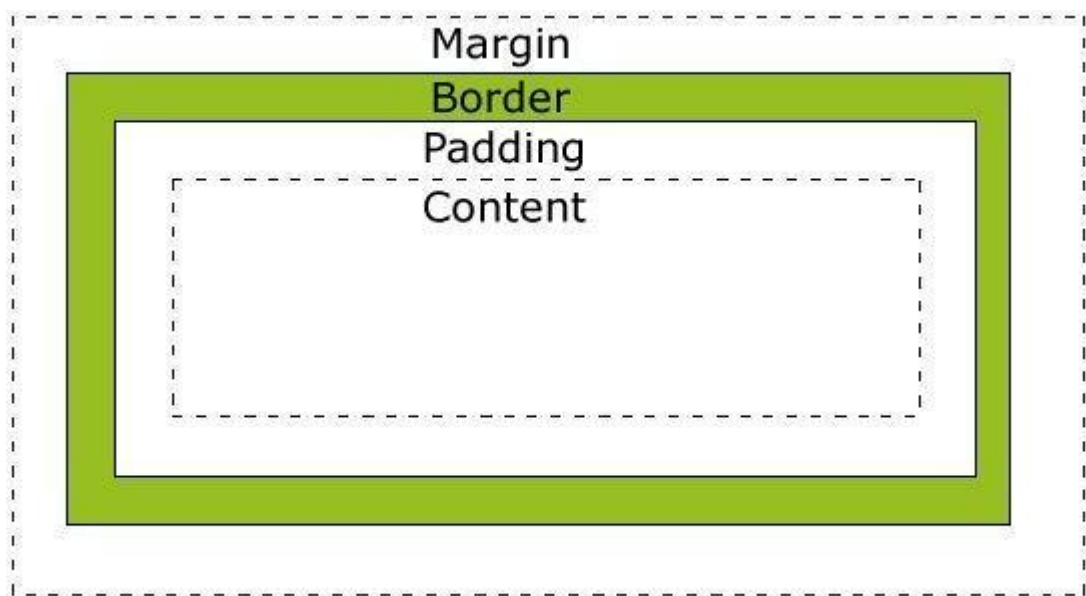
```
input:focus  
{  
    color:yellow;  
}
```

Span tags

Span's zijn vergelijkbaar met divisions (DIV), maar dan een inline element t.o.v van een block level element. Tekst met een `ROOD` woord.

Margin, Border, Padding

Aan de hand van een schematisch voorbeeld krijgen we een goed inzicht.



top right bottom left (clockwise)

De margin (Marge) is de afstand van de rand van een element tot de rand van een aangrenzend element. Margin gebruiken we om ruimte te creëren tussen elementen. We kunnen dezelfde waarden aangeven voor alle 4 zijden van een element, maar dit kunnen uiteraard ook 4 aparte waarden zijn. **Voldoende om slechts 1 waarde op te geven, indien deze voor elke zijde gelijk zijn. Deze regel is ook van toepassing op padding en border .**

`P { margin:0;}` (elk element p heeft geen waarde, dus nul px) M.a.w. we schakelen de marge uit. `P { margin:3px 5px 8px 10px;}` De marge vanaf de top is 3px, rechts 5px, van onder 8px en van de linkerkant 10px)

De border is de rand van een element.. De randen kunnen voorzien worden met kleuren en met diktes en soorten randen, waarbij de volgorde al beschreven is bij Margin.

De randstijlen: dotted, dashed, groove, solid, inset, double, ridge en outset. De

diktes van de rand: medium, thin, thick en lengte zoals px, %, em

Een voorbeeld:

```
#header{  
  height: 100px; width:50%;  
  border: 5px #339933 solid;  
  background-color: #990000;  
  color: #fff;  
}
```



De padding geeft aan de hoeveel ruimte die er moet komen tussen de inhoud van het element en de rand. Dat betekent natuurlijk wel dat er minder ruimte voor de tekst overblijft. De padding neemt de achtergrond over van het element. De waarde wordt aangegeven in lengte en percentage. Zoals 2px, 1em, 10%.

(De grootte van de box wordt bepaald door de som van de afmetingen van het element en de breedte van de margin, de border en de padding.)

Kleur

Kleuren maken het leven een stuk vrolijker en kunnen een website verrijken. CSS biedt zeer veel mogelijkheden om tekst of achtergronden een kleur te geven.

De Css-eigenschap color bepaalt de <voorgrondkleur> van de tekst. En <background> de kleur van de achtergrond.

```
Body{ color:#008000; background:#FFc;
```

Er zijn verschillende manieren om kleuren te schrijven:

Als RGB(0,23,103)

Hexadecimaal #003399

Maar ook benoemen bij de engelse naam zoals blue, red, etc.

Voor de eenheid en duidelijkheid stel ik voor om de hexadecimale kleuren te gebruiken. Een hexadecimale kleurcode bestaat uit 3 groepen van ieder 2 tekens.

Deze 3 groepen vertegenwoordigen de RGB-waarden, dus de 3 kleuren Rood, Groen en Blauw. Als de 2 tekens van een kleur dezelfde zijn, kun je deze inkorten door maar 1 teken te schrijven.

Enkele voorbeelden:

#000000 wordt #000

#778888 wordt #788

```
body {color:maroon; background:#FCF;}  
h2 {color:#FFC;background:maroon;}
```

Donkere tekst op licht achtergrond

lichte tekst op donkere achtergrond

Donkere tekst op licht achtergrond

lichte tekst van h2 op donkere achtergrond

Achtergrond

Background, omvat alle mogelijkheden om de achtergronden van een pagina in te stellen. Zowel met- als zonder afbeelding etc. Ook kan de achtergrondkleur van een element transparant zijn.

P {background:transparent;} < let op: transparent is met e !

We kunnen ook de onderstaande vormen gebruiken. Er kunnen dan specifieke waarden worden meegegeven.

Deze zijn:

background-color
background-image
background-repeat
background-attachment
background-position
Inherit

Background-attachment

Als je een afbeelding als achtergrond gebruikt op je pagina, zie je bij het scrollen de afbeelding mee beweegt naar beneden of naar boven. Door gebruik te maken van background-attachment kun je de afbeelding fixeren, zodat deze blijft staan en dat de pagina onafhankelijk van de afbeelding beweegt.

scroll
fixed
inherit

Scroll is de default. Wil je dat de afbeelding niet beweegt als je naar beneden scrollt dan gebruik je:

Body {background-image: url("afbeelding.gif"); background-attachment:fixed;}

Nu zal de afbeelding blijven staan en scrollt de pagina onafhankelijk van de afbeelding

Background-position

Als je een afbeelding op een bepaalde plaats als achtergrond wilt hebben dan kun je background-position gebruiken. Background-repeat moet dan wel op NO gezet worden. De browser plaatst de afbeelding standaard links boven in de hoek. Hierbij horen de waarden: 0%, 0%. De waarden:

percentage
lengte in pixels
top
left
left top
left center
left bottom
right
right top
right center
right bottom
center center
top
center center
center bottom
x% y%
xpos ypos
Inherit

Een voorbeeld om een achtergrondaafbeelding op de pagina te plaatsen:

```
Body {background-image:url("afbeelding.gif"); background-repeat: no-repeat;  
background-position: 50% 50%;}
```

De afbeelding zal nu in het midden van de pagina komen.

Wil je de afbeelding links onder in de hoek krijgen, dan zet je 0%, 100% bij background-position. Wil je echter de afbeelding rechts onder in de hoek dan wordt het 100%, 100% als waarde

Fonts

Eigenschappen van Fonts

font-family – Hierbij kun je een voorkeurslijst opgeven welke lettertypen je wilt gebruiken p

{font-family: verdana, sans-serif, "Times New Roman";}

font-size – Je kunt de grote opgeven in absolute afmetingen, zoals px of pt, maar ook in relatieve, zoals procenten of em.

font-weight – Keuze: normal, bold, bolder en lighter enz. *font-style*

– Keuze: normal, italic en oblique.

font-variant – Keuze: normal of smallcaps.

Definieer je standaardhoogte in de bodydefinitie: `font-size: 80%;` of `font-size: xsmall;`

Dit is een relatieve grootte, ten opzichte van het gekozen lettergrootte in de browser . Door de rest van de tekstgrootte relatieve afmetingen te geven, bijvoorbeeld percentages of em, kun je snel de lettergrootte van je teksten veranderen door alleen die in de body definitie te wijzigen. De verhouding tussen de teksten en de koppen blijven dan behouden.

Font-style

De font-style eigenschap bepaalt of de tekst normaal, cursief of schuin naar rechts hellend moet worden weergegeven. Er zijn 3 waarden voor font-style mogelijk

Normaal (is ook standaard)

Italic

Oblique

Font-weight

De font-weight eigenschap bepaalt het gewicht van de letters. De meest voorkomende waarden zijn:

normal

bold

waarde tussen 100 - 1000

Font-size

De font-size eigenschap bepaalt de grootte van de letters.

px, pt, em of %

Maten in CSS

CSS ondersteunt talrijke maten zoals pica (pc), inches (in), centimeters (cm), millimeters (mm) enzovoort. Maten kunnen gebruikt worden om lettergroottes te definiëren, diktes van randen te bepalen enzovoort. We kunnen al deze maten onderverdelen in 2 soorten...

De absolute maten definiëren een exacte grootte, onafhankelijk van andere elementen op de webpagina. De relatieve maten geven geen exacte grootte weer, maar hun weergave is afhankelijk van andere elementen op de webpagina.

Absolute maten

De "veiligste" maten in deze categorie zijn px (pixels) en pt (punten). Een fontgrootte van 13 pixels komt overeen met 10 punten.

Relatieve maten

Het procent en em zijn 2 maten in deze categorie. 200% is hierbij twee maal de standaard lettergrootte. 2em staat ook gelijk aan het dubbel van de standaard lettergrootte.

Teksteigenschappen

Aansluitend aan font eigenschappen kijken we nu naar een aantal teksteigenschappen die de vormgeving van de tekst eveneens beïnvloeden.

text-align - Uitlijning

Text-align bepaalt hoe tekst en afbeeldingen binnen een Block-element zijn uitgelijnd .

Waarden : left | right | center | justify | inherit

```
h4 { text-align: center; }
```

text-decoration - Tekstdecoratie

Decoratie toegevoegd aan de tekst, zoals een lijntje.

Waarde: none | underline | overline | line-through | blink | inherit

```
h1 { text-decoration: underline; }
```

line-height

Line-height bepaalt de regelaafstand tussen 2 opeenvolgende basislijnen van een tekst.

```
P {line-height: 2; }  
P {line-height: 1.1em; }  
P {line-height: 130%; }
```

letter-spacing - Letterspatiëring

Hiermee kun je de Ruimte die de browser tussen de lettertekens laat, vergroten of verkleinen

```
p { letter-spacing: 6px; }
```

Lijsten

Html kent drie soorten lijsten:

*de genummerde lijst (ol), onderverdeeld in list items (li);
de niet-genummerde lijst (ul), onderverdeeld in list items (li);
de definitielijst (dl), onderverdeeld in paren definition terms (dt) en definition descriptions (dd).*

Door een lijst op te nemen in een li of dd van een andere lijst ontstaat een geneste lijst. In een stijlblad kunt u de uiterlijke verschijningsvorm van een lijst bepalen. De eigenschap list-style beïnvloedt het opsommingsteken

(UL) unordered list / (OL) ordered list

De begintag of . li geeft elke keer een optie aan.

```
<ul>  
<li>Item 1</li>  
<li>Item 2</li>  
<li>Item 3</li>  
<li>Item 4</li>  
<li>Item 5</li>  
</ul>
```

Li { List-style-type:square;}

Hieronder zijn de waarden die men kan gebruiken voor een unordered list:

```
disc  
square  
circle  
None
```

We kunnen ook kiezen om geen cirkeltje of vierkantje voor het Item te krijgen. We gebruiken dan eenvoudig Li { List-style-type:None }.

Float en clear

Float

Door deze eigenschap in te stellen kan een element een andere positie krijgen dan normaal.

Standaard: none waarde: none | left | right

Manier om een element te kunnen uitlijnen ten opzichte van een ander element. Zo kan elk dom element weergegeven worden ten opzichte van andere elementen.

Clear

Aangeven of er links en/of rechts van het element zwevende elementen mogen voorkomen. Standaard: none. Aan de opgegeven kant mogen geen zwevende elementen meer voorkomen.

Waarde: none | left | right | both

"Niet geclearde elementen die volgen op een float (of: verschillende floats), kruipen in het hoogst mogelijke gaatje naast de float(s)."

clear gebruik je om te bepalen aan welke kant een niet-floating element mag verschijnen. wanneer je both gebruikt mag er aan beide zijden niets verschijnen, met left en right bepaal je dit voor links en rechts. Vaak gebruikt om bijvoorbeeld een footer onderaan een pagina te definiëren.