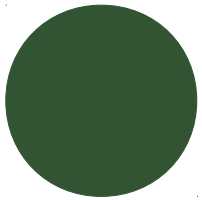
Mijn-
eigen-
website
.nl

Wat je moet weten over...

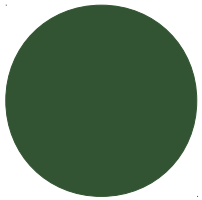
Teksten op je website

Inhoudsopgave



Voorwoord

Dit e-book gaat volledig over het vormgeven van de teksten op je website. Je leest bijvoorbeeld hoe je ervoor zorgt dat speciale karakters altijd goed worden weergegeven, hoe je een lijst met cijfers of bullets maakt, en hoe je stukjes tekst beter naar voren kunt laten komen. Alles om ervoor te zorgen dat je webpagina's prettig zijn om te lezen. Want van een beeldscherm lezen is nooit zo fijn als lezen van papier. Dus grijp alles aan om de aandacht van je lezer vast te houden. En prettig vormgegeven tekst is een van de manieren waarop je dat kunt doen.



1. De juiste karakterset kiezen

Wat je allereerst moet weten als je teksten op je website wilt zetten, is dat er verschillende karaktersets bestaan. En dan heb ik het niet over lettertypes, zoals *Comic Sans* of `Courier`. Ik doel op verzamelingen van letters, cijfers, punten, komma's, e.d. waarmee teksten uit een bepaald taalgebied worden weergegeven.

Het Spaans heeft bijvoorbeeld een vraagteken dat op zijn kop wordt neergezet vooraan een vragende zin (¿), een karakter dat we in het Nederlands niet kennen. Scandinavische talen hebben letters met schuine strepen erdoor (ø), rondjes erboven (å) of lettercombinaties die strak tegen elkaar aan staan (æ). En talen als Russisch, Chinees of Grieks hebben veel meer tekens die voor ons onbekend zijn.

Om teksten goed op een webpagina te laten weergeven, moet je daarom op twee dingen letten:

1. de karakterset waarin je je webpagina opslaat
2. de karakterset die de browser gebruikt om je webpagina weer te geven

Dit moeten dezelfde karaktersets zijn, anders kun je zoiets krijgen als dit:

Ã©Ã©n

Hier is niets anders aan de hand dan dat de webpagina waarop deze letters staan, is opgeslagen in de ene karakterset, terwijl de browser die de pagina vertoont van een andere karakterset gebruikmaakt.

De twee karaktersets die voor Nederlandse websites het belangrijkst zijn, zijn `iso-8859-1` en `utf-8`. De eerste set bevat wat minder karakters dan de tweede, maar voldoende voor het Nederlandse taalgebied. De tweede karakterset – `utf-8` – is een enorm grote karakterset, die tekens uit praktisch alle talen ter wereld kan weergeven. Dus daarom raad ik meestal aan om webpagina's op te slaan in `utf-8`.

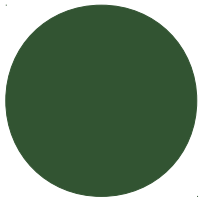
(Niet in alle html-editors kun je zelf kiezen in welke karakterset je je webpagina's wilt opslaan. Is er geen keuze, dan worden je pagina's meestal opgeslagen in `utf-8`.)

Heb je je pagina opgeslagen, dan moet je de browsers laten weten welke karakterset ze moeten gebruiken als ze die pagina weergeven. Dat moet dezelfde karakterset zijn als die waarin je je pagina hebt opgeslagen. Dat doe je door in de `head` van je pagina de volgende regel te zetten:

```
<meta charset="utf-8">
```

Ook in je stylesheet kun je een karakterset-code aangeven. Ik heb het zelf eigenlijk nooit gedaan, en er nooit problemen door gehad, maar voor de volledigheid noem ik het hier toch even. Helemaal bovenaan je stylesheet kun je de volgende regel neerzetten:

```
@charset "utf-8";
```



2. Structuur aanbrengen met HTML-tags

Álle teksten die je op een webpagina neerzet, moeten op de een of andere manier worden gelabeld of gemerkt: je moet in de eerste plaats per tekst(gedeelte) aangeven of het om het belangrijkste tekstgedeelte op die pagina gaat, of juist om kortere stukken tekst aan de boven-, onder- of zijkant van je pagina (de menu- of navigatiebalk sla ik in dit verhaal even over). Bínnen die verschillende gedeeltes moet je weer aangeven of een stukje tekst een kop of subkop is, een gewone alinea, of misschien een lijst, een citaat, een link, enzovoort. Uiteindelijk moet echt álle tekst gelabeld zijn.

Labelen (of merken) doe je in HTML met behulp van zogenaamde ***tags***. De meeste *tags* komen in setjes: een openings-*tag* en een bijbehorende sluit-*tag*. Met de eerste geef je aan waar een bepaald gedeelte begint, en met de tweede waar het eindigt. De sluit-*tag* ziet er bijna hetzelfde uit als de openings-*tag*, er staat alleen een extra *forward slash* (*/*) binnen de *tag*. Een alinea bijvoorbeeld, komt tussen een set p-*tags* te staan (de letter p staat voor *paragraph*):

```
<p>
   Dit is een alinea. Vooraan de alinea staat de openings-tag, achteraan staat
   de corresponderende sluit-tag.
</p>
```

Door deze alinea te labelen met een set *paragraph-tags* heb je er een zogenaamd **element** van gemaakt: het is een van de vele bouwstenen of onderdelen van je pagina geworden. Je kunt het nu duidelijk identificeren omdat je het begin en eind ervan hebt aangegeven. Ook heb je er - impliciet - een betekenis aan gegeven.

Door te kiezen voor *paragraph-tags* heb je gelijk duidelijk gemaakt dat dit gedeelte een alinea is (en niet een kop of subkop bijvoorbeeld).

Bínnen deze alinea kun je ook weer *tags* aanbrengen. Stel dat ik wil dat browsers de twee woorden *tag* cursief weergeven, dan doe ik dat zó:

```
<p>
   Dit is de inhoud van een alinea. Vooraan de alinea staat de openings-
   <i>tag</i>, achteraan staat de corresponderende sluit-<i>tag</i>.
</p>
```

Binnen het *paragraph*-element staan dus nu twee andere elementen: woorden die cursief moeten worden weergegeven. De `i`-*tags* zijn hier zgn. **geneste *tags***: ze staan binnen de `p`-*tag*.

Een overzicht van de belangrijkste *tags* die je voor teksten gebruikt, staan hieronder. De letters die voor de *tags* zijn gebruikt, zijn niet willekeurig gekozen, maar afgeleid van Engelse woorden. Als je weet welke woorden dat zijn, kun je gemakkelijker onthouden welke *tags* je moet gebruiken voor een bepaald tekstgedeelte.

article

Wat een `article` is, hoef ik denk ik niet uitgebreid uit te leggen: een verzameling bij elkaar horende alinea's met in ieder geval een kop erboven, en misschien een of meer subkoppen. Dus het merendeel van de tekst voor je webpagina komt allereerst binnen een set `article`-*tags* te staan.

aside

Binnen een `aside`-element zet je tekst (en eventueel foto's) neer die niet echt onder je `article` vallen, maar er wel zijdelings mee te maken hebben. Bij veel websites staat de inhoud van de *sidebar* tussen een set `aside`-*tags*.

header, footer

Binnen het `header`-element van een webpagina staan meestal een domeinnaam en/of logo en binnen het `footer`-element in veel gevallen een regel met een copyright-vermelding. Maar je kunt hier natuurlijk nog veel meer neerzetten.

Bínnen het `article`-, `aside`-, `header`- en `footer`-element gebruik je de volgende *tags* om de teksten die erin staan te structureren. Er bestaan nog meer *tags* dan degene die ik hieronder noem, maar dit zijn de belangrijkste, en degene die je het meest zult gebruiken.

h1, h2, h3, h4, h5, h6

De `h` staat voor *heading* ofwel 'kop'. Binnen je webpagina mag je zes kopniveaus gebruiken: van koppen tot subkoppen tot subsubkoppen, enzovoort.

```
<h1>De eerste kop boven de tekst op je webpagina</h1>
```

p, br

De *paragraph-tags* (`p`) gebruik je voor de verschillende alinea's waaruit je tekst bestaat. Wil je binnen een alinea de tekst op een nieuwe regel laten beginnen, dan gebruik je daar de *break-tag* voor (`br`). Deze heeft geen aparte openings- en sluit-*tag*. Dat hoeft ook niet, want het element dat je met deze *tag* maakt (het equivalent van een harde return op je toetsenbord) heeft verder geen inhoud. Je kunt het bijvoorbeeld neerzetten in een alinea waarin je naam en adres staan:

```
<p>
   Voor- en achternaam<br>
   Straatnaam<br>
   Plaatsnaam
</p>
```

ol, ul, li

Lijsten maak je door stukken tekst binnen `ol`-, `ul`- en `li`-*tags* te zetten. Voor een lijst met cijfers of letters ervoor gebruik je een set `ol`-*tags* (waar `ol` staat voor *ordered list*, ofwel: geordende lijst). Wil je een lijst met bullets ervoor, dan gebruik je een set `ul`-*tags* (`ul` staat voor *unordered list*: ongeordende lijst).

En alle lijstonderdelen — of ze nu binnen een geordende of ongeordende lijst staan — komen tussen `li`-*tags* te staan (waarbij `li` staat voor *list item*: lijstonderdeel).

```
<ul>
   <li>maandag</li>
   <li>dinsdag</li>
   <li>woensdag</li>
   <li>donderdag</li>
   <li>vrijdag</li>
</ul>
```

Bestaat je opsomming niet uit een lijstje woorden, maar uit een lijstje zinnen, dan kun je elke zin ook weer tussen zijn eigen *paragraph-tags* zetten. Je hebt dan dus een alinea bínnen elk lijstonderdeel. Omdat browsers standaard wat ruimte inbrengen boven en onder een alinea, zullen de verschillende lijstonderdelen zo wat verder uit elkaar komen te staan.

```
<ul>
   <li><p>zondag is de eerste dag van de week</p></li>
   <li><p>maandag is de tweede dag van de week</p></li>
   <li><p>dinsdag is de derde dag van de week</p></li>
   <li><p>woensdag is de vierde dag van de week</p></li>
   <li><p>donderdag is de vijfde dag van de week</p></li>
   <li><p>vrijdag is de zesde dag van de week</p></li>
   <li><p>zaterdag is de laatste dag van de week</p></li>
</ul>
```

Wil je een lijst bínnen een lijst maken, dan doe je dat zó (voor de overzichtelijkheid heb ik de code voor de tweede lijst een ander kleurtje gegeven):

```
<ol>
   <li>rood
      <ul>
         <li>donkerrood</li>
         <li>felrood</li>
         <li>lichtrood</li>
      </ul>
   </li>
   <li>oranje</li>
   <li>geel</li>
   <li>groen</li>
   <li>blauw</li>
</ol>
```

i, em, b, strong

Met de *tags* voor *italic* (`i`) of *emphasis* (`em`) laat je een tekstgedeelte cursief weergeven. Voor het eindresultaat maakt het niet uit welke van de twee je gebruikt; in beide gevallen zal het gedeelte wat tussen de openings- en de sluit-*tag* staat, cursief worden.

Toch is er een subtiel verschil: als je de `em`-*tags* gebruikt, dan geef je daarmee aan dat je het stukje tekst dat ertussen staat, ook inhoudelijk wilt benadrukken omdat het om de een of andere reden belangrijk is. Dus wil je een stukje tekst alleen cursief laten weergeven omdat je dat mooier vindt, gebruik dan de `i`-*tags*, wil je het ook inhoudelijk een bepaalde nadruk geven, kies dan voor de `em`-tags.

Hetzelfde geldt voor de *tags* waarmee je een stukje tekst vet kunt laten weergeven (`b` voor *bold* en `strong` spreekt voor zichzelf: sterk). Gaat het je alleen om het uiterlijk, kies dan de `b`-*tags*, wil je ook aangeven dat het stukje tekst inhoudelijk belangrijk is, kies dan voor de `strong`-*tags*.

Om tekst vet én cursief te laten weergeven, kun je de elementen die je met behulp van deze *tags* aanmaakt ook weer binnen elkaar neerzetten:

```
<p>Sommige woorden binnen deze alinea zullen <i>cursief</i> worden
weergegeven, sommige <b>vet</b>, en sommige <i><b>vet én cursief</b></i>.</p>
```

Zorg er bij deze geneste *tags* wel voor dat het element dat je het eerste aanmaakt, het laatste afsluit. Dus wél: `<i><b>vet én cursief</b></i>`, maar NIET: `<i><b>vet én cursief</i></b>`.

mark

Wil je een stukje tekst laten highlighten met een marker, dan zet je het tussen `mark`-*tags*:

```
<p>Het woord <mark>highlighten</mark> zal er in deze zin uitzien alsof het is gemarkeerd met een gele marker.</p>
```

sub, sup

Staan er in je tekst woorden als H_2O of km^2, dan moet je van de letters die een klein stukje naar beneden of juist naar boven moeten worden geplaatst een `sub`- *(subscript)* of `sup` *(superscript)*-element maken. Dat doe je door de betreffende gedeeltes tussen `sub`- of `sup`-*tags* te zetten:

```
H<sub>2</sub>O en km<sup>2</sup>
```

a

Wil je van een of enkele woorden een link maken, dan zet je ze tussen een set `a`-tags. De `a` staat voor *anchor* (anker). In de openings-*tag* van de link zet je neer wat de linkbestemming moet zijn, met andere woorden, waar de bezoeker heen geleid moet worden als iemand op deze link klikt. Deze linkbestemming staat achter de letters `href`, ofwel *hypertext reference*:

```
<a href="https://www.mijn-eigen-website.nl">Mijn-eigen-website.nl</a>
```

Als iemand op de woorden *Mijn-eigen-website.nl* klikt, dan wordt hij naar de homepage van deze site geleid.

Over links is nog veel meer te vertellen, zoals van welke html-onderdelen je links kunt maken, op welke manier je bezoeker moet worden doorgestuurd naar de linkbestemming en hoe je het uiterlijk van je links beïnvloedt. Maar dat valt buiten het bestek van dit e-book. Meer over dit onderwerp lees je in het e-book HTML en Hyperlinks.

blockquote, cite, q

Het kan zijn dat er in de tekst op je webpagina citaten staan. Dat kunnen korte citaten zijn, niet meer dan een gedeelte van een zin bijvoorbeeld, maar ook langere, van een of meer alinea's. In het eerste geval maak je van het citaat een `q`-element door het tussen een set `q`-tags te zetten; in het tweede geval maak je er een `blockquote`-element van. De `q` staat natuurlijk voor *quote* (en het woord `blockquote` spreekt voor zichzelf):

```
<p>Een bekend citaat van Einstein is: <q>Logica brengt je van A naar B. Verbeelding brengt je overal.</q></p>

<blockquote>Het is het verstandigste om zoveel mogelijk te luisteren, want dat is goed voor je algemene ontwikkeling, als je er tien procent van leert, dan ben je de anderen al een eind voor. (Johan Cruijff)</blockquote>
```

Browsers zetten automatisch aanhalingstekens om `q`- en `blockquote`-elementen heen, dus dat hoef je zelf niet meer te doen.

figcaption

Als foto's of andere afbeeldingen die je op je webpagina opneemt een bijschrift hebben, dan zet je die bijschriften tussen een set `figcaption`-*tags*. Over de plaatsing van foto's en de positie van eventuele bijschriften die erbij horen, lees je meer in het e-book Foto's op je website.

span, div

Soms wil je bepaalde stukken uit je tekst wel een afwijkend uiterlijk geven, ook al hebben die stukken verder geen speciale betekenis. Daar bedoel ik mee dat het geen opsomming is, of citaat, of bijschrift van een foto bijvoorbeeld. Er bestaat dus geen speciale *tag*-set voor je tekstgedeelte, maar je wilt het toch wel graag onderscheiden van de rest van de tekst. In dat geval gebruik je een set `span`-*tags* (als het om een gedeelte binnen een enkel woord of een enkele zin gaat), of een set `div`-*tags* (als het om een of meerdere alinea's gaat).

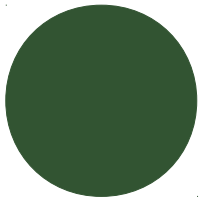
Stel dat ik bijvoorbeeld om de een of andere reden een paar woorden binnen een alinea rood en vet wil laten weergeven. Ik moet dan deze woorden tussen een set `span`-*tags* zetten:

```
<p>Binnen deze alinea moeten <span>deze woorden rood en vet</span> worden
weergegeven.</p>
```

Met alleen deze code zal er nog niet veel gebeuren, maar de paar woorden waar het om gaat, zijn in ieder geval apart gezet van de tekst eromheen. In hoofdstuk 3 leg ik uit hoe je de woorden die tussen deze set `span`-*tags* staan naar eigen inzicht kunt vormgeven.

Is het stuk tekst dat je wilt markeren groter dan enkele woorden of letters, op zijn minst een volledige alinea, dan gebruik je geen `span`-*tags*, maar `div`-*tags*:

```
<div><p>Deze alinea moet straks een heel ander uiterlijk krijgen dat de rest
van de alinea's die op de pagina staan.</p></div>
```



3. Vormgeven met CSS-eigenschappen

Als je structuur in je webteksten hebt aangebracht door ervoor te zorgen dat álles binnen HTML-*tags* staat, dan is de volgende stap dat je die verschillende elementen gaat vormgeven met behulp van CSS. De afkorting CSS staat voor *cascading stylesheets*. Een *stylesheet* is een stijlblad: een bestand waarin de layout (of stijl) van een website en alle onderdelen daarbinnen is vastgelegd.

Het woord *cascading* (letterlijk vertaald: als een waterval) betekent dat de stijlkenmerken van een bepaald html-element automatisch ook worden toegepast op alle elementen die daarbinnen staan. Dus als je bijvoorbeeld instelt dat alinea's moeten worden weergegeven in een bepaald lettertype, dan zullen ook de woorden die je binnen die alinea met `i`- of `b`-*tags* hebt gemarkeerd, datzelfde lettertype krijgen.

Drie manieren van vormgeving

Er zijn drie manieren waarop je met behulp van CSS de onderdelen van je website kunt vormgeven:

1. Je kunt in de openings-*tag* van een element de code `style=""` neerzetten. Binnen de set dubbele aanhalingstekens zet je dan de stijlkenmerken neer die dat element moet krijgen. Dit heet een **inline style**. Stel dat bijvoorbeeld een paar woorden uit een alinea rood en vetgedrukt moeten worden weergeven, dan doe je dat zó:

```
<p>Binnen deze alinea moeten <span style="color: red; font-weight:
bold">deze woorden rood en vet</span> worden weergegeven.</p>
```

2. Wat je ook kunt doen, is in de `head` van je pagina een stijlblok neerzetten. Dit heet een **internal stylesheet**. Deze methode is gemakkelijk als je op een bepaalde pagina álle elementen van een bepaald type hetzelfde uiterlijk wilt geven. Zo'n stijlblok begint met `<style>` en eindigt met `</style>`.

   Stel dat de de tekst van élke alinea op je pagina blauw moet worden, en dat de eerste regel van elke alinea 24 pixels moet inspringen. In de `head` van je pagina zet je dan het volgende stijlblok neer:

```
<style>
  p { color: blue; text-indent: 24px; }
</style>
```

   Zoals je in dit voorbeeld ziet, begin je met het noemen van het bewuste element. Dat heet de **selector** (hier is dat het element `p`). Daarachter zet je een set accolades neer en tussen die accolades plaats je de verschillende stijlkenmerken voor dit element. Eerst komt de **eigenschap** (in het voorbeeld: `color`), daarachter een dubbele punt, en daarachter de **waarde** (in het voorbeeld: `blue`). De *combinatie van eigenschap en waarde* heet **declaratie**. Elke declaratie sluit je af met een puntkomma. Ben je klaar met de vormgeving van je selector, dan kun je daaronder verdergaan met de kenmerken voor een ander element. In zo'n stijlblok kun je dus de stijlkenmerken van meerdere elementen kwijt.

   NB: binnen een stijlblok mag je zoveel witregels en spaties toevoegen als je wilt. Dus vind je het overzichtelijker om elke declaratie op een nieuwe regel te laten beginnen, ga gerust je gang! Dus het stukje code van hierboven kan er dus ook zó uitzien:

```
<style>
p {
   color: blue;
   text-indent: 24px;
   }
</style>
```

3. Ten slotte kun je de stijlkenmerken van een element ook nog vastleggen in een extern stijlblad (**external stylesheet**). De meeste elementen zullen op alle pagina's van je website dezelfde layout moeten krijgen. Het is dan onhandig als je in de `head` van elke pagina een stijlblok zou moeten neerzetten. Een extern stylesheet is dan veel eenvoudiger. En als je op een enkele pagina een bepaald element eens een afwijkend uiterlijk wilt geven, dan overschrijf je de eigenschappen uit het externe stylesheet gewoon door nieuwe eigenschappen voor dit element op te geven in een stijlblok in de `head`, of direct in de openings-*tag* van het element zelf. Want de instructies van een i*nline style* hebben altijd voorrang op de instructies van een *internal stylesheet*, en die hebben op hun beurt weer voorrang op de instructies van een *external stylesheet*.

Wat je verder nog moet weten over stylesheets, is dat browsers ook nog hun eigen, interne stylesheets hebben. Zo komt het dat de tekst die je tussen `h1`-*tags* hebt neergezet altijd een stuk groter en vetter wordt weergegeven dan tekst die je tussen `p`-*tags* had gezet – ook al had je daar niks over aangegeven. Dat doet het browser-stylesheet dus. Maar de regels die jij zelf opgeeft (op welke manier dan ook), overschrijven altijd eventuele browser-regels.

Vormgeving met een external stylesheet

In de voorbeelden die ik verderop in dit e-book geef, ga ik ervan uit dat je werkt met een *external stylesheet*. In de `head` van elke webpagina moet je aan de browsers doorgeven in welk stylesheet ze de regels kunnen vinden op basis waarvan ze de elementen op die webpagina moeten vormgeven. Dat doe je met deze code:

```
<link href="css/basis.css" rel="stylesheet" type="text/css" media="screen">
```

Wat je hiermee in feite zegt, is dit: Maak voor de vormgeving van de verschillende elementen op deze webpagina gebruik van een stylesheet dat `basis.css` heet, en dat te vinden is in een map met de naam ‘css’.

Natuurlijk kun je je stylesheet ook gerust anders noemen: `style.css` bijvoorbeeld, of `stijlblad.css`. Het bestand moet alleen wel de extensie `.css` hebben, en het moet te vinden zijn op de plaats die jij in de `head` hebt aangegeven.

In het stijlblad ga je nu stuk voor stuk de regels opschrijven die de browsers moeten toepassen voor de vormgeving. Dat gaat weer op dezelfde manier als ik heb aangegeven op pag. 18 en 19. Alleen de twee `<style>`-tags kun je nu weglaten. Je kunt direct beginnen met de *selector*. Dus stel dat je de koppen die je hebt gemarkeerd met `h1`-*tags* blauw wil laten weergeven, dan doe je dat zó:

```
h1 { color: blue; }
```

Je noemt dus eerst het element (de *selector*), daarachter zet je een spatie met een set accolades, en binnen die accolades zet je de declaratie (eigenschap en waarde) neer die je wilt laten toepassen op dit element.

Wil je diezelfde declaratie laten toepassen op meerdere elementen, dan kun je ze gewoon achter elkaar neerzetten, met een komma ertussen. Dus wil je álle koppen op je pagina blauw laten weergeven (ook alle subkoppen), dan wordt dit de declaratie:

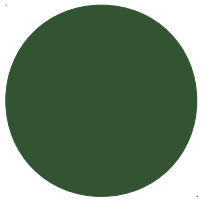
```
h1, h2, h3, h4, h5, h6 { color: blue; }
```

En je kunt de selectie ook vernauwen als je wilt. Dus wil je niet álle koppen die je met `h1`-tags hebt gemarkeerd blauw laten weergeven, maar alleen de koppen die binnen het `article`-element staan (en dus niet eventuele `h1`-koppen binnen het `header`-, `footer`- of `aside`-element), dan noem je eerst dit `article`-element en daarachter het `h1`-element (met een spatie tussen beide elementen):

```
article h1 { color: blue; }
```

Met deze bovenstaande declaratie worden dus alleen de koppen blauw die binnen het `article`-element staan, en die je tussen `h1`-*tags* had gezet.

Alles wat hier buiten de set accolades staat, heet in CSS-taal een **selector**: dat is datgene waarop je de stijlregel(s) binnen de accolades wilt laten toepassen. Het gedeelte vóór de dubbele punt is de **eigenschap**, en het gedeelte erachter de **waarde**. De *combinatie van eigenschap en waarde* heet **declaratie**.



4. Overzicht van de belangrijkste CSS-eigenschappen voor tekst

color

Achter de eigenschap `color` zet je natuurlijk neer welke kleur de *selector* (het gedeelte dat je hebt aangegeven buiten de accolades) moet krijgen. Wil je dat álle tekst op je pagina dezelfde kleur krijgt, dan kun je deze eigenschap het beste laten toepassen op het `body`-element. Want dat is het allereerste element op een pagina. Die eigenschap wordt dan vanzelf toegepast op alle elementen die bínnen het `body`-element staan (en dat zijn altijd alle andere elementen).

```
body { color: darkslategrey; }
```

Kleuren kun je op verschillende manieren aangeven; in het voorbeeld hierboven heb ik een kleurnaam gebruikt, maar er zijn ook andere manieren om een kleur vast te leggen. Meer over dit onderwerp lees je in het artikel Kleuren met html-codes.

Tekst waarvan je een link hebt gemaakt, wordt altijd blauw (en onderstreept) weergegeven, ook al heb je dat nergens aangegeven. Dat is zo in de browser-stylesheets vastgelegd. Wil je dus dat je links een andere kleur krijgen, dan moet je die waarde overschrijven in je eigen stylesheet:

```
a { color: darklategrey; }
```

Met deze regel worden natuurlijk álle links op je pagina grijs van kleur. Als je bijvoorbeeld alleen de kleur van de links in je menubalk wilt veranderen, dan moet je de *selector* uitbreiden:

```
nav a { color: darkslategrey; }
```

Zo worden alleen de link-elementen (`a`) die binnen het navigatie-element (`nav`) staan grijs van kleur.

font-family

```
font-family: Verdana, Helvetica, Arial, sans-serif;
```

Achter deze eigenschap zet je neer welk lettertype de *selector* moet krijgen. Je kunt hier meerdere lettertypes achter elkaar neerzetten, gescheiden door een komma. De browser zal eerst proberen de *selector* in het eerstgenoemde lettertype weer te geven. Als degene die jouw webpagina bekijkt dat lettertype niet op zijn computer heeft staan, dan zal de browser het tweede lettertype proberen dat je hebt genoemd, enzovoort.

Want jij kunt bijvoorbeeld wel zeggen dat je wilt dat de koppen op je pagina in AvantGarde worden weergegeven, maar als op het apparaat van degene die jouw webpagina bekijkt dat lettertype niet geïnstalleerd is, dan zal dat niet gaan. Dus daarom kun je beter meerdere lettertypes opgeven.

Je eindigt je opsomming altijd met een lettergroep in plaats van een specifiek lettertype (in het voorbeeld hierboven is de lettergroep `sans-serif`). Want op elke computer is altijd wel één lettertype uit een bepaalde groep geïnstalleerd. Dus als op het apparaat van degene die jouw webpagina bekijkt geen van de lettertypes is geïnstalleerd die je hebt genoemd, dan wordt je pagina in elk geval wel vertoond in een lettertype dat erop lijkt. Lettergroepen waaruit je kunt kiezen, zijn: serif, sans-serif, `monospace`, cursive, en *fantasy*.

Wat ook kan - als je per se wilt dat een *selector* in een bepaald lettertype wordt vertoond, is dat je dit lettertype als het ware 'meelevert' met je webpagina. Hoe dat moet, lees je in het artikel Wat te doen als je bijzondere website-lettertypes gebruiken wilt.

font-size

```
font-size: 3em;
```

Met deze eigenschap stel je de lettergrootte van de *selector* in. Achter de dubbele punt kun je waarden neerzetten in verschillende eenheden: pixels (`px`), percentages (`%`), `em`'s, `rem`'s, `vw`'s en ook bepaalde keywords: `larger`, `smaller`, `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large` en `xx-large`.

De waarden `em` en `vw` zijn waarden die je waarschijnlijk niet zoveel zeggen. De waarde `1em` is niet meer dan een referentiepunt. Als je nergens iets hebt opgegeven voor de lettergroottes op je webpagina zal `1em` meestal hetzelfde zijn als `16px`. De letters `vw` staan voor *viewport width* (beeldschermbreedte). Een waarde van `10vw` betekent 10% van de beeldschermbreedte.

Als je waarden opgeeft voor de `font-size`, dan komt bij sommige eenheden natuurlijk de vraag op hoe ze worden berekend. Een pixel is een absolute waarde, en wat de *viewport width* is, is ook duidelijk, maar als je opgeeft dat de `font-size` van je alinea's `90%` moet zijn, of `smaller`, dan wil je weten: 90% van wat, of kleiner dan wat?

Soms worden de waarden die je opgeeft, berekend aan de hand van de waarde die geldt voor de pagina als geheel (het `html`-element). Dat is het geval bij de eenheden `rem`, `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large` en `xx-large`.

In de andere gevallen (`em`, `larger` en `smaller`) worden de waarden berekend aan de hand van de waarde die geldt voor het element waarbínnen de *selector* zich bevindt. Stel dat de grootte van de letters op de webpagina `1em` is, en binnen het `article`-element `.75em`. Geef je vervolgens op dat de koppen van het eerste niveau binnen dit element (`article h1`) op `200%` moeten weergegeven, dan is dat `200%` van `.75em`, en niet `200%` van `1em`.

font-style

```
font-style: italic;
```

Met deze eigenschap kun je instellen of een stukje tekst cursief *(italic)* of misschien nog wat schuiner *(oblique)* weergegeven moet worden. Je kunt kiezen uit drie waarden: `normal`, `italic` en `oblique`.

font-variant

```
font-variant: small-caps;
```

Deze eigenschap wordt gebruikt als je een *selector* in KLEINKAPITALEN wilt laten weergeven.

font-weight

```
font-weight: bold;
```

Wil je een tekstgedeelte, een woord of een letter vet maken, dan kun je dat doen door die tussen `b`- of `strong`-*tags* te zetten, maar soms is het handiger om een regel in het stylesheet op te nemen (bijvoorbeeld als je van die *selector* nog meer eigenschappen wilt instellen).

Achter de `font-weight` eigenschap kun je verschillende waarden neerzetten: `bold` (zoals in de voorbeeld-regel hierboven), maar ook getallen: `100` (dun), `200` (zeer licht), `300` (licht), `400` (normaal), `500` (medium), `600` (halfvet), `700` (vet; is dus hetzelfde als `bold`), `800` (extra vet) en `900` (zeer vet).

Daarnaast bestaan er ook nog de waarden `bolder` en `lighter`, waarmee je een *selector* één stapje lichter of vetter laat weergeven dan het element waarvan het deel uitmaakt.

line-height

```
line-height: 130%;
```

De `line-height` is de regelafstand. Deze hangt samen met de lettergrootte: hoe groter het lettertype waarin je een tekst laat weergeven, hoe groter ook de `line-height` zal zijn. Maar een tekst op het scherm wordt vaak wat gemakkelijker leesbaar als je de standaardwaarde van de `line-height` iets verhoogt.

Dat kan je op verschillende manieren doen: door een percentage te gebruiken, zoals in het voorbeeld hierboven, door een waarde in `em`'s op te geven, of door een waarde op te geven zonder maateenheid erachter, bijvoorbeeld: `line-height: 2.5`. In dat geval wordt de `line-height` berekend door het getal dat je opgeeft te vermenigvuldigen met de lettergrootte.

font

Hiervóór staan een groot aantal CSS-eigenschappen genoemd die allemaal betrekking hebben op de letterweergave. Als je meerdere van die eigenschappen wilt opgeven voor een bepaalde *selector* dan zou je dus een heleboel van die regels onder elkaar krijgen. In dat geval is het handiger ze te combineren in een enkele regel, achter de CSS-eigenschap `font`:

```
font: italic small-caps bold 2em/3 sans-serif;
```

Je moet de opsomming van de verschillende eigenschappen wel in een bepaalde volgorde neerzetten: `font-style`, `font-variant`, `font-weight`, `font-size/line-height` and `font-family`.

Je hóeft ze natuurlijk niet allemaal te gebruiken; je kunt achter de `font`-eigenschap ook bijvoorbeeld alleen de `font-style` en de `font-size` neerzetten:

```
font: italic, 2.5em;
```

De overige eigenschappen houden dan gewoon hun standaardwaarde.

letter-spacing

```
letter-spacing: 2em;
```

De waarden achter `letter-spacing` kun je aangeven in `pixels` of in `em`'s. Hiermee worden de letters van je *selector* wat v e r d e r u i t e l k a a r geplaatst.

text-align

```
text-align: center;
```

De eigenschap `text-align` kun je alleen toepassen op zogenaamde *block elementen*. Een alinea (`p`) is bijvoorbeeld een **block element**, een woord dat je cursief hebt gemaakt met behulp van een set `i`-*tags* is dat niet; dat is een **inline element**.

Met de waarde achter `text-align` geef je aan hoe de inhoud binnen het block element uitgelijnd moet worden: tegen de linker- (`left`) of rechterrand aan (`right`), gecentreerd (`center`), uitgelijnd behalve de laatste regel (`justify`) of volledig uitgelijnd, dus ook de laatste regel (`justify-all`).

De naam van deze eigenschap doet het niet vermoeden, maar je kunt `text-align` ook gebruiken om `img`-elementen (foto's of afbeeldingen) uit te lijnen. Het `img`-element is namelijk ook een *inline* element. Dus zet je meerdere `img`-elementen naast elkaar en geef je het element waarbinnen ze staan de declaratie `text-`

`align: center`; dan worden deze afbeeldingen netjes gecentreerd. (Dit werkt niet met het `figure`-element, omdat dat een *block* element is.)

text-decoration

```
text-decoration: none;
```

De bovenstaande regel zul je zo af en toe vast willen gebruiken om het standaarduiterlijk van hyperlinks te veranderen. Browsers geven een link namelijk standaard blauw en onderstreept weer. Vaak is dat prima, maar in andere gevallen (bijvoorbeeld in je menubalk) wil je dat links een ander kleurtje krijgen, en wil je die onderstreping ook weghalen. De kleur veranderen doe je natuurlijk met de CSS-eigenschap `color`, maar voor het weghalen van de onderstreping heb je de eigenschap `text-decoration` nodig. Met de waarde `none` verdwijnt deze onderstreping onder links.

Behalve none kun je nog meer waarden kiezen: `underline` (lijntje onder de tekst), `overline` (lijntje erboven) en `line-through` (lijn in het midden erdoorheen).

Als je wilt, kun je ook nog een kleur voor het lijntje aangeven:

```
text-decoration: underline red;
```

En ten slotte kun je nog het type lijn kiezen. De standaardwaarde is `solid` (een ononderbroken lijn), maar wil je iets anders, dan kun je ook nog kiezen uit `double` (dubbele lijn), `dotted` (stippellijn), `dashed` (lijn met streepjes) en `wavy` (golflijntje):

```
text-decoration: wavy underline blue;
```

text-indent

```
text-indent: 3em;
```

Hiermee laat je de eerste regel van een tekst een stukje inspringen. De waarde geef je op in pixels, `em`'s, `rem`'s of percentages.

text-shadow

```
text-shadow: 3px 2px 5px black;
```

Door een stukje tekst of een paar letters een schaduw te geven, kun je heel leuke effecten bereiken. Met deze eigenschap is het bijvoorbeeld mogelijk je tekst een 3D-uiterlijk te geven.

De eerste waarde die je achter deze eigenschap neerzet is de schaduw naar rechts (de X-as), en de tweede naar beneden (de Y-as). Ook negatieve waarden gebruiken mag: dan krijg je een schaduw naar links en naar

boven. De derde waarde is de mate van vervaging: in de voorbeeldcode hiervóór vervaagt de schaduw over 5 pixels. En ten slotte geef je nog een kleur op voor de schaduw.

text-transform

```
text-transform: uppercase;
```

Heb je per ongeluk een kop (of een ander tekstgedeelte) in kleine letters getypt, en wil je die eigenlijk in hoofdletters laten weergeven, dan kun je daarvoor de eigenschap `text-transform` gebruiken. Als je als waarde voor die eigenschap `uppercase` opgeeft, dan zal die tekst in hoofdletters verschijnen. Andersom kan ook, dan gebruik je natuurlijk `lowercase`. En met de eigenschap `capitalize` wordt de eerste letter van elk woord weergegeven als hoofdletter. Op deze manier kun je bijvoorbeeld heel makkelijk in één keer de letters van de links in je menubalk veranderen.

hyphens

```
hyphens: auto;
```

Met bovenstaande regel geef je aan dat de tekst van de *selector* automatisch moet worden afgebroken als hij niet op een regel past. Nu zijn afbreekregels per taal verschillend, dus je moet tegelijkertijd ook even aangeven in welke taal je pagina is geschreven. Dat doe je in het `html`-element: `<html lang="nl">`.

Behalve de waarde `auto`, kun je ook de waarde `manual` instellen. Dan werkt woordafbreking ook, maar moet je de afbreekpositie zelf aangeven met behulp van een hard (-) of zacht (`&shy;`) afbreekteken.

Een andere manier om te zorgen dat je webteksten worden afgebroken, is door gebruik te maken van een script. In het artikel Automatisch woorden afbreken in html lees je hoe dat in zijn werk gaat.

padding, margin

```
padding: 20px 10px 15px 10px;
margin: 20px 10px 15px 10px;
```

De eigenschappen `padding` en `margin` hebben betrekking op de ruimte om de *selector* heen. Achter deze beide eigenschappen staan in dit voorbeeld hierboven vier waarden: een voor de ruimte aan de bovenkant (`20px`), een voor de ruimte aan de rechterkant (`10px`), een voor de ruimte aan de onderkant (`15px`), en ten slotte een voor de ruimte aan de linkerkant (`10px`).

Als de ruimtes voor boven/onder en voor links/rechts nu gelijk zouden zijn (bijvoorbeeld boven en onder een ruimte van `20px` en links en rechts een ruimte van `10px`), dan zou er ook dit kunnen staan:

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

En als aan alle vier de zijden dezelfde hoeveelheid ruimte zou moeten komen (bijv. `15px`) , dan zou het volgende voldoende zijn:

```
padding: 15px;
margin: 15px;
```

Met de eigenschappen voor zowel `padding` als `margin` komt er ruimte om een element. Maar er zit wel een subtiel verschil tussen deze beide eigenschappen.

Een element kun je zien als een velletje papier. De rand van het papier is de `border`. (Van veel elementen zie je de `border` niet omdat er geen kleur en dikte voor is ingesteld, maar hij is er wel degelijk.) De tekst die op het velletje papier staat, staat nooit strak tegen de randen aan, maar er is altijd wat ruimte aan alle kanten. Wij zouden dat een marge noemen, maar in html-taal spreek je van een `padding`. Leg je twee velletjes papier naast elkaar, dan kun je die strak tegen elkaar aan leggen, maar je kunt er ook wat ruimte tussen laten. En dat is de `margin`.

Tot slot

Met al deze voorbeelden en codes heb je aardig wat materiaal in handen om ervoor te zorgen dat de teksten op je webpagina's er goed uitzien en prettig lezen. Kijk op de artikelenpagina van Mijn-eigen-website.nl voor meer voorbeelden om je teksten vorm te geven.