

CSS

Met CSS kan je je webpagina's vormgeven. Je kan elementen een plaats geven en hun uitzicht bepalen.

In CSS maken we een onderscheid tussen **2 soorten elementen in een webpagina**:

1. Sommige elementen gedragen zich als een “**doos**” (box, block) en kan je een hoogte en een breedte geven. Net zoals een “doos” een hoogte of een breedte heeft.

2. Sommige elementen kan je geen hoogte of breedte meegeven nl. **tekstelementen**. Je kan niet zeggen dat een titel 100 pixels hoog moet zijn, want de hoogte van een titel wordt bepaald door zijn lettergrootte.

Starten met CSS

Tekstelementen zoals titels, alinea's... kan je dus geen hoogte of breedte geven, maar je kan wel een heleboel andere eigenschappen opmaken:

1. lettertype
2. lettergrootte
3. marge (links, boven, rechts, onder... of allemaal tegelijk)
4. kleur

Uiteraard gebruiken we voor CSS ook vaste benamingen in het Engels. Je kan niet zo maar wat gaan schrijven.

lettertype	font-family
lettergrootte	font-size
marge (links, boven, rechts, onder... of allemaal tegelijk)	margin (margin-left, margin-top, margin-right, margin-bottom...)
kleur	color
uitlijning	text-align

“Doos”elementen kan je ook nog andere eigenschappen meegeven:

breedte	width
hoogte	height
positie (een absolute plaats of een relatieve plaats) tov. andere elementen	position
plaats op de pagina (links, rechts...)	float
stapelen (elementen boven elkaar plaatsen)	z-index
afronding	border-radius
achtergrondkleur	background-color
achtergrondafbeelding	Background-image
marge (links, boven, rechts, onder... of allemaal tegelijk): de afstand van dit element tov andere elementen daar buiten	margin (margin-left, margin-top, margin-right, margin-bottom...)
padding (links, boven, rechts, onder... of allemaal tegelijk): de afstand van dit element tov andere elementen daar binnen	padding (padding-left, padding-top, padding-right, padding-bottom...)
rand	border (border-left, border-top, border-right, border-bottom...)

Hiermee hebben we nog lang niet alle mogelijkheden opgesomd, maar je kan er al mee aan de slag.

Hoe noteren we CSS?

In het bestand “**bouweenwebpagina.pdf**” leer je hoe je elementen in een webpagina kan selecteren om ze vorm te geven met CSS.

Voor de duidelijkheid: we plaatsen al onze CSS-code in een apart bestand in de map “style” van de websitemap. We bewaren dit bestand met de naam “style.css”.

Vermits dit bestand via een “link” is gekoppeld aan al onze webpagina's (zie coderegel hieronder), zal elke wijziging aan het bestand “style.css” een onmiddellijk effect hebben op de weergave van *al onze pagina's*.

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

Eerst bepalen we de algemene instellingen die voor de hele pagina gelden. Heel het visuele gedeelte van de website staat in het onderdeel `<body>`.

Om alle ergerlijke marges en ruimtes te vermijden zetten we alle marges op 0.

We selecteren het `<body>`-element.

```
body{
    margin:0;
    padding:0;
}
```

We willen dat heel de inhoud van de pagina netjes centraal zal staan. Voor oude versies van Internet Explorer moet je daarom één regeltje toevoegen aan dit stukje code.

```
body{
    margin:0;
    padding:0;
    text-align:center;
}
```

Een beetje vreemd eigenlijk, want hiervoor dient `text-align` eigenlijk niet, maar plaats je het niet, dan doet IE helemaal niet wat je verwacht. IE redeneert: “Alle elementen gaan netjes in het midden staan. IE beveelt zoals een schoolmeester tegen zijn kindjes.” Andere browser doen dit op een andere manier.

Het eerste element dat we in onze `<body>` tegenkomen is een `<div>`-element. Div-elementen gedragen zich letterlijk als dozen, daarom zie je ze ook zo vaak opduiken in de HTML-code van webpagina's.

Opgelet: als je nu het `<div>`-element zou selecteren, zouden je CSS-instructies gelden voor alle `<div>`-elementen op de pagina.

Dus als je dit zou schrijven:

```
div{
    border: 1px solid red;
}
```

dan zou niet alleen de eerste `<div>` een rood randje krijgen, maar alle div's op de pagina.

Hoe lossen we dit op?

Vergelijk het met een leraar die straf geeft. Hij geeft geen straf aan alle jongens, maar aan één specifieke jongen. In onze HTML-code hebben we aan alle `<div>`-elementen een identiteit gegeven door de parameter `id=""`. De eerste `<div>` draagt als id de naam “`container`”.

Een id kan je selecteren op de volgende manier:

```
#container
```

Zo'n beetje zoals op Twitter: een hekje gevolgd door de “id”. Om deze container ook netjes te centreren in alle andere browsers (buiten IE), stellen we de marges in op automatisch. De “container” weet hierdoor dat hij zelf in het midden moet gaan staan, hij heeft geen “bevel” nodig van het body-element zoals in IE.

```
#container{
    margin:auto;
}
```

Door de extra instructie voor IE, zien we als resultaat dat alle inhoud in het midden gaan staan. Immers: IE vroeg `text-align="center"`. Het gevolg is dat effectief alles gecentreerd wordt.

Dit lossen we snel op:

```
#container{
    margin:auto;
    text-align:left;
}
```

We geven de container ook een breedte. Je kan een vaste breedte kiezen in pixels (standaard is 960px omdat de webpagina dan op alle platformen kan weergegeven worden) of een procentuele waarde.

Vaste breedte in pixels:

```
#container{
    margin:auto;
    text-align:left;
    width:960px;
}
```

Een breedte in procenten (fluid layout):

```
#container{
    margin:auto;
    text-align:left;
    width:90%;
}
```

In dit laatste geval, vult de container 90% van de beschikbare breedte van het browservenster. Zowel links, als rechts blijft er dan een marge van 5% over.

Je merkt meteen het verschil tussen beide als je je browservenster begint te *resizen*.

Je kan zowel voor de body als voor de container een aparte achtergrondkleur instellen:

```
body{
    margin:0;
    padding:0;
    text-align:center;
    background-color:black;
}

#container{
    margin:auto;
    text-align:left;
    width:90%;
    background-color:white;
}
```

Vervolgens komen we in de `<div id="container">` terecht. In deze “doos” zitten nog andere dozen die we eerst en vooral op de juiste plaats moeten zetten en de juiste afmetingen geven. Eerst en vooral stoten we op `<div id="header">`. Hierin komt onze bannerafbeelding en een slogan.

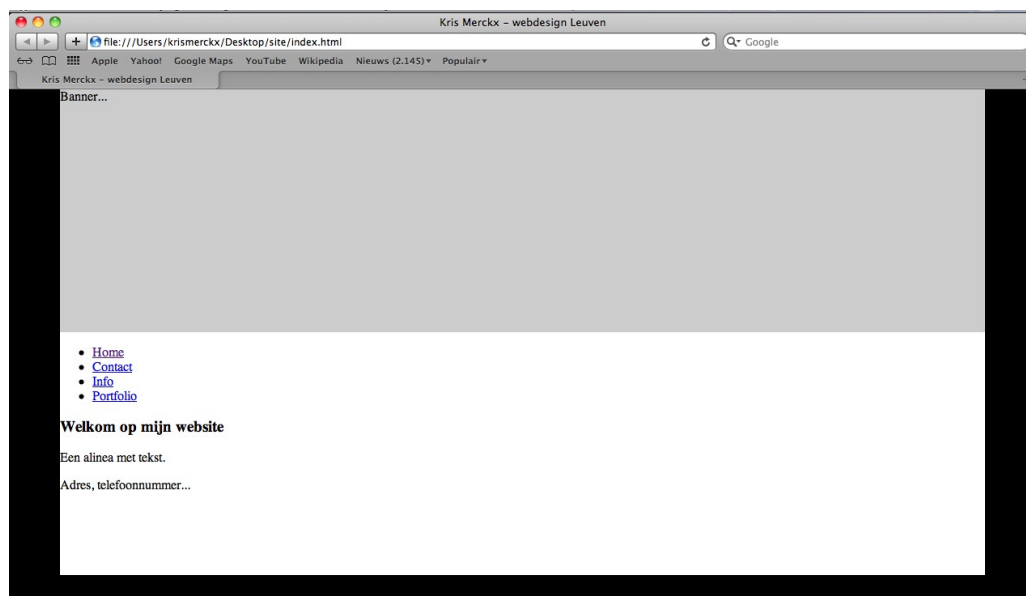
```
#header{
    height:300px;
}
```

De bedoeling is dat we in onze header een achtergrondaafbeelding plaatsen, maar daar wachten we nog even mee tot de rest van de pagina in orde is. Wel kunnen we al een achtergrondkleur toekennen zodat we duidelijk het verschil zien tussen de “header” en de rest van de pagina.

```
#header{
    height:300px;
    background-color:#cccccc;
}
```

Je merkt dat we hier de waarde `#cccccc` toegevoegd hebben. Dat is de RGB-kleurwaarde voor lichtgrijs. We komen op de kleurwaardes later nog terug.

Onze pagina ziet er nu ongeveer uit zoals in het voorbeeld hieronder:



Na de “header” vinden we de volgende onderdelen:

```
<div id="pagecontent">
  <div id="navigation">
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="contact.html">Contact</a></li>
      <li><a href="info.html">Info</a></li>
      <li><a href="portfolio.html">Portfolio</a></li>
    </ul>
  </div>
  <div id="content">
    <section>
      <article>
        <h1>Welkom op mijn website</h1>
        <p>Een alinea met tekst.</p>
      </article>
    </section>
  </div>
</div>
```

De doos “pagecontent” bevat nog twee ingesloten dozen. Onder de “header” staat dus een doos die gevuld is met twee andere dozen: de `<div id="navigation">` en de `<div id="content">`

De buitenste “doos” `<div id="pagecontent">` zorgt ervoor dat de rest netjes op zijn plaats blijft staan. Binnen die “div” verdelen we de pagina in twee “kolommen”. Eentje voor het navigatiemenu (dit plaatsen we aan de linkerkant), eentje voor de eigenlijke paginainhoud (“content”).

Voor de buitenste doos (`<div id="pagecontent">`) hoeven we dus niet echt “opmaak” te regelen. Toch merken we een onhandigheidje op. Als er zo goed als geen inhoud in onze pagina staat, plakt de “footer” (het onderste deel van de webpagina) meteen tegen de “inhoud” en de “navigatie” aan. We lossen dit even op door een hoogte te geven aan `<div id="pagecontent">`. Naargelang de inhoud die je in de pagina plaatst, wordt de pagina automatisch hoger. Hij lijkt een beetje op een doos met een rubberen onderkant. Hij kan oneindig uitrekken.

Je zal merken dat een “procentuele” hoogte (bijvoorbeeld: `height:100%`) over het algemeen niet zal werken. Een browservenster heeft immers geen vaste “hoogte”. Als de tekst meer ruimte inneemt dan de schermhoogte of de hoogte van het browservenster, verschijnen er schuifbalken (scrollbars).

Een hoogte kan je niet “procentueel” instellen omdat een webpagina nooit een “vaste” hoogte heeft, je kan hem in principe oneindig lang maken. We kunnen dit deel van de webpagina echter wel een minimumhoogte (min-height) geven.

```
#pagecontent{  
    min-height: 600px;  
}
```

We gaan nu naar `<div id="navigation">`.

```
#navigation{  
}
```

De navigatiebalk verschijnt links en laten we ongeveer 200 pixels ruimte innemen van de “container”.

```
#navigation{  
    float:left;  
    width:200px;  
}
```

Dit zal in de meeste browsers meteen het gewenste effect krijgen. De inhoud gaat automatisch rechts staan, maar als je meer tekst toevoegt, kan het fout lopen.



We moeten er voor zorgen dat de pagina-inhoud niet plots aan de linkerkant uitlijnt als hij langer wordt dan het navigatiemenu. Het navigatiemenu is 200 pixels breed. Als we de pagina-inhoud een linkermarge geven van 210 pixels, blijft die altijd rechts staan en plakt die niet meteen tegen het navigatiemenu aan.

```
#content{  
    margin-left:210px;  
}
```

Je merkt dat de “footer” ook automatisch op zijn plaats staat. Onze pagina is echter nogal “kort” omdat er op dit moment niet echt veel tekst in de pagina staat.

We richten onze pijlen nu op het navigatiemenu. Voor onze links staan nog zwarte bolletjes, de links zijn standaard “blauw” en onderstreept. Dat moet en kan anders.

We moeten nu goed opletten wat we doen. De links (`<a>`) staan stuk per stuk in een “lijstitem” (``), die ``-elementen zijn een onderdeel van een ongeordende lijst (``), en die lijst staat op zijn beurt in een `<div>` met als id “`navigation`”.

Of omgekeerd

`DIV#navigation => UL => LI => A`

We kijken eerst naar het diepste element, de links. We moeten opletten dat we onze stijl enkel toepassen op de links in het navigatiemenu en niet op alle links in de pagina.

We schrijven dus niet

```
a{  
}
```

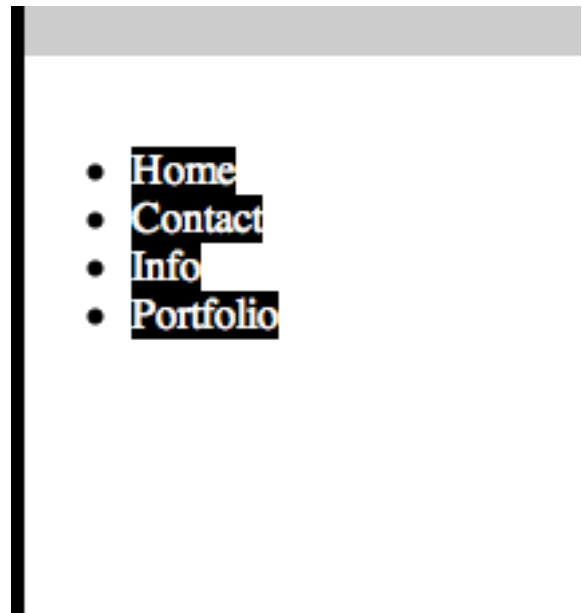
maar

```
#navigation ul li a{  
}
```

We willen onze links standaard in het wit, de achtergrond in het zwart en de lijn eronder moet weg:

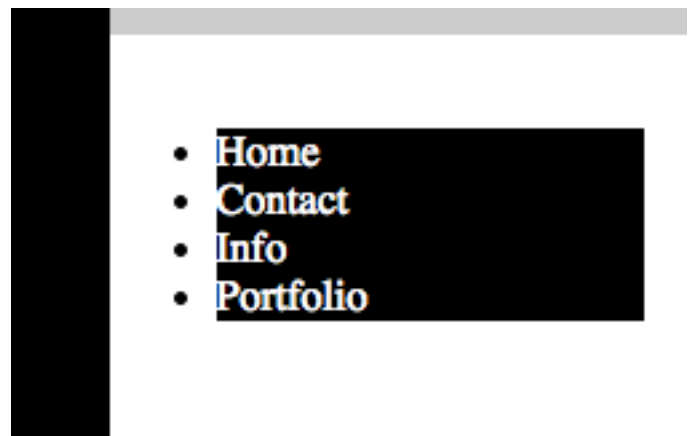
```
#navigation ul li a{  
    color:white;  
    background-color:black;  
    text-decoration:none;  
}
```

Je merkt meteen dat dit niet het gewenste resultaat oplevert. De stijl is wel aangepast, maar de zwarte achtergrond is slechts zo breed als het opschrift van de link en de link is enkel “op de tekst” aanklikbaar. *De link gedraagt zich niet als een knop.*



We moeten aan onze browser vertellen dat hij de links moet behandelen als een doos, als één “blok”.

```
#navigation ul li a{  
    color:white;  
    background-color:black;  
    text-decoration:none;  
    display:block;  
}
```



Dit ziet er al wat beter uit, maar nog niet helemaal zoals we wensen:

1. Er staan vervelende “bolletjes” voor de links.
2. Het navigatiemenu staat erg naar rechts.
3. Het opschrift van de link plakt helemaal tegen het zwarte kader aan.

Het probleem met de bolletjes heeft niets te maken met de links zelf. De “bolletjes” zijn een eigenschap van de list-items (``) en de lijst (``).

Om ze te laten verdwijnen, moeten we de lijst selecteren.

```
#navigation ul{  
    list-style-type:none;  
}
```

De vervelende bolletjes zijn nu verdwenen, maar de lijst plakt nog steeds niet tegen de linkerkant van de container aan. Om dit doel te bereiken, plaatsen we alle marges op 0.

```
#navigation ul{  
    padding:0;  
    margin:0;  
    list-style-type:none;  
}
```



Dat ziet er al aardig beter uit. Nu moeten we de afstand van de opschriften tot de rand nog aanpassen. Hiervoor moet je niet “margin”, maar de “padding” (de marges aan de binnenkant van een element) vergroten.

```
#navigation ul li a{  
    color:white;  
    background-color:black;  
    text-decoration:none;  
    display:block;  
    padding:13px;  
}
```

Het zou leuk zijn om de links een ander uitzicht te geven, als je er met de muis overheen beweegt. Daarvoor gebruiken een bijzondere CSS-techniek:

We selecteren opnieuw de links, maar vertellen nu wat er moet gebeuren als de gebruiker met zijn muis “over” een “hyperlink” (:hover) een link beweegt. Dan veranderen we de kleur van het lettertype en de achtergrondkleur.

```
#navigation ul li a:hover{
    color:black;
    background-color:white;
}
```

Nu ontbreekt er nog een lettertype. Dit hoef je maar één keer in te stellen nl. in het “body”-element.

```
body{
    margin:0;
    padding:0;
    text-align:center;
    background-color:black;
    font-family: helvetica, arial, sans-serif;
}
```

Je merkt dat er niet één, maar meerdere lettertypes zijn ingegeven. Waarom? Niet op alle computersystemen staan dezelfde lettertypes. Je moet dus voorzien dat wanneer één lettertype ontbreekt, de browser automatisch op zoek gaat naar een ander. Als “helvetica” niet aanwezig is, zoekt hij naar “arial”. Als ook dat niet aanwezig is, neemt hij het eerste lettertype zonder “voetjes”.

Zie zo... onze eerste CSS-layout is klaar. Tijd om te tweaken, te finetunen... aan te passen aan je eigen smaak.

Hulpmiddelen:

HTML-kleuren	http://www.w3schools.com/tags/ref_colorpicker.asp
Kleurpaletten	http://www.paletton.com/
Lettertypes	https://www.google.com/fonts
CSS3-effecten met browsercontrole	http://www.css3maker.com