

Werken met blokken

5.1 Inleiding

Nu je geleerd hebt om te werken met CSS, kun je aan de slag om nog mooiere websites te gaan maken. In dit hoofdstuk ga je leren hoe je je website opbouwt uit blokken en hoe je die blokken een goede opmaak geeft.



ontwerp
blokken

Als je een website wilt maken, begin je niet direct met het schrijven van code in HTML en CSS, maar met een goed **ontwerp**. In het ontwerpproces denk je niet alleen na over de inhoud van de website, de afbeeldingen en over de kleuren die je gaat gebruiken. Een van de belangrijkste onderdelen van het ontwerpen is het nadenken over de structuur van elke pagina. Om een goede structuur te krijgen, moet je leren 'denken in blokken'.

Websites zijn opgebouwd uit **blokken**: er is een blok voor het logo, voor de navigatie, voor het belangrijkste onderwerp, voor tekst, enzovoorts. Om je daarvan een goed beeld te geven, zie je hier een paar voorbeelden van websites die een prijs hebben gewonnen voor het ontwerp. Let eens goed op de manier waarop bij deze websites alle onderdelen geplaatst zijn en probeer in gedachten alle elementen die bij elkaar horen als aparte blokken te zien.



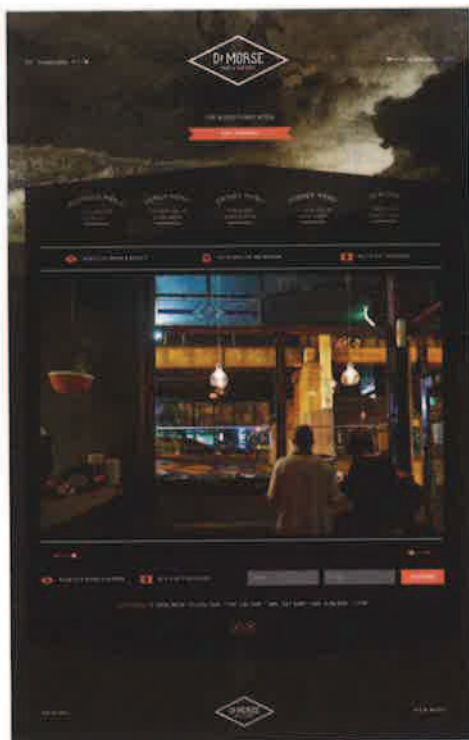
rudys.paris



inlayinsights.com



source-tree.com



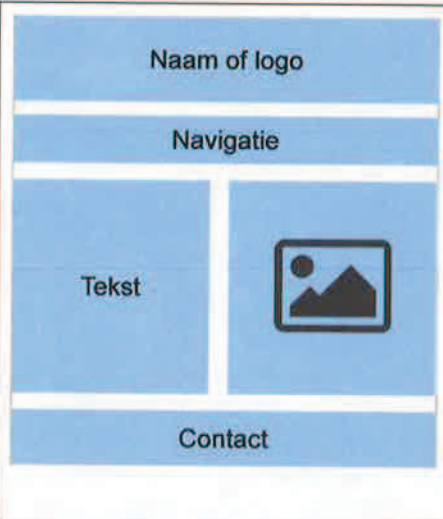
dr-morse.com.au

Hoe verschillend de websites ook zijn, je ziet steeds een verzameling blokken met elk een eigen functie. Wil je meer weten over webdesign? Kijk dan eens in hoofdstuk 7.

5.3 Standaardblokken en divisions

HTML is erop gemaakt om een webpagina op te bouwen uit **blokken**. De blokken vormen de basis voor de structuur en bepalen daarom voor een belangrijk deel hoe overzichtelijk je website is. Ga je een website ontwerpen, bedenk dan eerst hoeveel en welke blokken je wilt. Die blokken teken je vervolgens en daarna beschrijf je waarvoor elk blok dient. Als je ontwerp af is, neem je altijd als eerste de blokstructuur over in HTML. Dat doe je met structuurelementen, die we voor het gemak blokelementen zullen noemen. Hoe ziet dat eruit?

Hieronder zie je een eenvoudig webpagina-ontwerp en de bijbehorende HTML-code uit de body, naast elkaar.

<pre><header> <h1>Naam of logo</h1> </header> <nav> <p>Navigatie</p> </nav> <article> <p>Tekst</p> </article> <div id="afbeelding"> </div> <footer> <p>Contact</p> </footer></pre>	 A diagram of a web page layout with five distinct blocks. The top block is a blue header containing the text 'Naam of logo'. Below it is a white navigation bar with the text 'Navigatie'. The main content area is divided into two columns: the left column contains the text 'Tekst', and the right column contains an image icon representing a landscape with a house and a sun. At the bottom is a blue footer containing the text 'Contact'.
--	--

Je ziet in het ontwerp vijf blokken en hun functie. Sommige van die blokken worden zo veel gebruikt, dat deze een vaste naam hebben gekregen in HTML. Dat zijn (van boven naar beneden):



- **header**, om de naam, het logo of de titel van de website weer te geven.
- **nav**, om de navigatie (het menu) van je website weer te geven.
- **article**, om een stuk tekst neer te zetten.
- **footer**, om onderaan contact-informatie en de sitemap weer te geven.

Niet voor elke soort blok is er een vaste naam. Als je een blok hebt dat niet tot één van de standaardblokelementen behoort, dan kun je een blok maken die je zelf een naam geeft. Zo'n blok heet een **division**. Dat doe je met het `div`-element. In de code wordt er een `div`-element aangemaakt met de naam 'afbeelding'. Als je een `div`-element gebruikt, moet je een `id` opgeven, dus bijvoorbeeld:

```
<div id="afbeelding">
```

De opmaak van blokelementen in CSS gaat als volgt:

<pre>article { background-color: #cccccc; width: 300px; }</pre>	<pre>#afbeelding { width: 300px; height: 200px; }</pre>
---	---

Een stijlregel met een hekje (#) maak je voor een element met een `id`. Je noemt dat een `id-selector`. Bij de standaardblokken maak je een CSS-regel aan met simpelweg de naam van dat blok en een accolade openen en sluiten (we gebruiken dus een `tag-selector`). Bij een blokelement dat je zelf een naam (`id`) gegeven hebt (een `div`-element dus), maak je een stijlregel aan door een # neer te zetten met daarachter de door jouw gekozen naam van het blok. Je kunt voor elk blok een hoogte en breedte opgeven, zoals je ziet in de bovenstaande CSS-code. Dat doe je met de eigenschappen `width` en `height`. Je kunt trouwens voor elk `html`-element een breedte en hoogte opgeven, dus bijvoorbeeld ook voor een `a`-element. Dat is handig als je een menuknop wilt maken. Hoe je dat doet, leer je in opdracht 2 van dit hoofdstuk.

Heade

Nav

Article

Footer

Divisi

5.4 Opmaak van geneste elementen

st
ent

Op een webpagina komen dezelfde html-elementen meestal meer dan één keer voor. Zo is er bij het voorbeeld in paragraaf 5.3 een p-element in article, maar ook in footer. We zeggen dat p genest is in article of footer.

in

Stel nu dat je het p-element in article een andere opmaak wilt geven dan het p-element in footer. Je hebt dan niets aan de stijlregel voor p, omdat die voor alle p-elementen geldt. Als je een stijlregel voor p in een **genest element** wilt aanmaken, dan noteer je in CSS dat geneste element vóór p. Dat gaat zo:

ling

er

<pre>#afbeelding { width: 300px; height: 200px; }</pre>	<pre>footer p { color: blue; }</pre>
---	--

De paragraaf in het blok article heeft nu een rode kleur en de paragraaf in de footer een blauwe.

5.5 Margin, padding en border van een blok

Je weet inmiddels dat je zelf de hoogte en breedte van blokken kunt bepalen. Maar er is nog veel meer dat je kunt instellen. Zo kun je bijvoorbeeld ruimte om het blok heen maken. We noemen dat **margin**. Je kunt ook de ruimte tussen de rand van het blok en de inhoud daarvan (tekst bijvoorbeeld) instellen. Dat noemen we **padding**. Tot slot kun je de dikte, de kleur en de stijl van de rand van het blok instellen, de **border**.



Hieronder zie je voorbeeldcode van hoe je margin, padding en border kunt instellen.

```
article {
  padding: 10px;
  margin: 50px 0px 10px 10px;
  border: 1px solid black;
}
```

Met de regel: **padding: 10px;** stel je een padding in van 10px aan alle vier de kanten van het blok. In dit voorbeeld zijn er voor de margin vier verschillende waarden opgegeven, voor elke kant van het blok een waarde. De volgorde daarvan is: bovenmarge, rechtermarge, ondermarge, linkermarge. Je kunt het onthouden met dit geheugensteuntje:

```
padding: Noord Oost Zuid West;
margin: 12 uur 3 uur 6 uur 9 uur;
```

Zoals je al uit het vorige hoofdstuk weet is de regel **border: 1px solid black;** een manier om drie regels in één te schrijven:

```
border-width: 1px;
border-style: solid;
border-color: black;
```

Bij padding en margin gebeurt weer hetzelfde. Nu zijn de regels margin-top, margin-right, margin-bottom en margin-left samengevoegd. Behalve `margin: 10px`; en `margin: 50px 0px 10px 10px`; kun je ook `margin: 50px 0px`; schrijven. De eerste waarde is dan voor boven en onder, en de tweede waarde voor links en rechts.

5.6 Het CSS boxmodel

Het gebeurt regelmatig dat je blokelementen nest, zoals in deze afbeelding.



De precieze plaats van de inhoud van het binnenste blok hangt af van de margin, padding en border-width van zowel het binnenste als het buitenste blok! Daar moet je rekening mee houden als je de blokken positioneert.

5.7 De echte afmetingen van een blok

Er is iets bijzonders aan de hand met de afmetingen van blokken. Zoals je weet geef je die op met width en height. In de onderstaande code is de breedte van de article-elementen opgegeven.


```
article {  
  width: 300px;  
  padding: 10px;  
  margin: 20px;  
  border: 5px solid black;  
}
```

Toch is de breedte van de article-elementen in de praktijk niet 300px. Dat komt omdat de opgegeven padding, margin en border elk het article-element breder maken. De werkelijk breedte is:

300px (width)
+ 20px (padding links en rechts)
+ 40px (margin links en rechts)
+ 10px (border links en rechts)
= 370px

De werkelijke hoogte kun je op een vergelijkbare manier berekenen.

5.8 Margin, padding en border bij andere elementen

Om te voorkomen dat je veel rekenwerk moet doen bij het maken van je website, is het aan te raden om bij blokken zorgvuldig met padding om te gaan. Als je ruimte tussen de rand en de inhoud van een blok wilt, kun je het beste padding toepassen op geneste elementen, zoals p-elementen. Het is namelijk mogelijk om niet alleen voor blokken, maar voor elk willekeurig html-element de margin, padding en border op te geven. Het is zelfs zo dat een aantal elementen automatisch een bepaalde margin hebben. Dat is bijvoorbeeld het geval bij koppen en het p-element.

Dit voordeel heeft ook een nadeel: je kunt te maken krijgen met de vervelende situatie dat je de afmetingen van bepaalde blokken op een webpagina niet krijgt zoals je wilt, door een (onbekende) standaardmargin van één van de html-elementen. Daarom is het goed om de standaardmarge uit te zetten door bovenin je CSS-bestand de volgende stijlregel op te nemen:

```
* {  
  margin: 0;  
}
```

De asterisk (*) geeft aan dat deze regel geldt voor alle html-elementen.

5.9 De wrapper

Meestal is je scherm breder dan de website die je er op bekijkt. In dat geval wil je natuurlijk dat de website gecentreerd op je scherm staat. Om dat te realiseren, moet je alle html-element in één overkoepeld blok zetten. Dat blok wordt meestal **wrapper** genoemd. Wrapper betekent 'wikkel' of 'omslag'. Hier zie je een voorbeeld:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>...</title>  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  <body>  
    <div id="wrapper">  
      <header>...</header>  
      <nav>...</nav>  
      <main>...</main>  
      <footer>...</footer>  
    </div>  
  </body>  
</html>
```

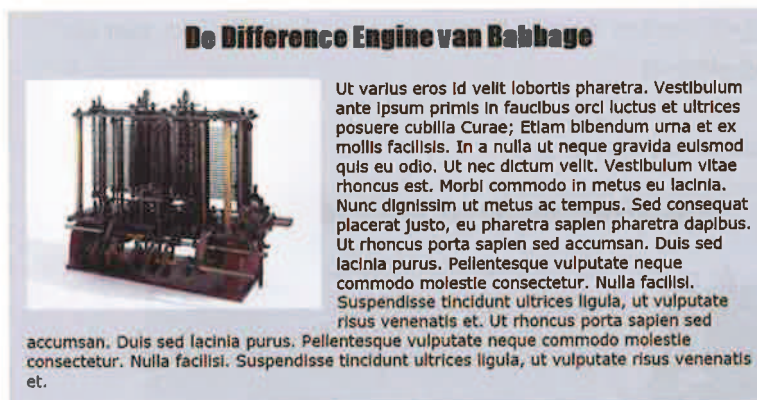
Om de website te centreren, moet je vervolgens twee dingen doen in CSS: de breedte opgeven van de wrapper en `margin: auto;` instellen. De automatische marge zorgt ervoor dat je webpagina links en rechts evenveel marge krijgt en dus gecentreerd wordt.

```
#wrapper {
  width: 900px;
  margin: auto;
}
```

De breedte van de wrapper is nu ook de breedte van je hele website. In paragraaf 5.11 kun je lezen welke grote voordelen dat oplevert als je de blokken van je website goed wilt positioneren.

5.10 Floating

Tekst en afbeeldingen wil je vaak naast elkaar plaatsen, op zo'n manier dat de tekst om de afbeelding heen staat, bijvoorbeeld zo:



De HTML-code voor dit blok (een article) is eenvoudig te maken: eerst een kop h1, dan een img-element en tot slot een paragraaf met de tekst:

```
<article>
  <h1>De Difference Engine van Babbage</h1>
  
  <p>Lorem ipsum dolor sit amet, ...</p>
</article>
```

In de CSS-code geven we de breedte en achtergrondkleur van het blok article op.

```
article {  
  background-color: #cccccc;  
  width: 900px;  
}  
  
article img {  
  width: 348px;  
  margin: 20px;  
}
```

De code voor de kop h1 en de paragraaf zijn hier weggelaten, die is hier even niet belangrijk. De tweede stijlregel die er staat is wel belangrijk: daar wordt de breedte van het plaatje opgegeven en ook een margin, zodat er 20 pixels ruimte om het plaatje vrij blijft. Let er op dat er geen 'img' staat, maar 'article img'. Zoals je inmiddels weet uit paragraaf 5.4, betekent dit dat de stijlregel alleen voor img-elementen in een article-element geldt en dus niet voor elk willekeurig img-element.

Wat is het resultaat van deze code?



Je ziet dat alle elementen onder elkaar worden weergegeven. Dat is niet wat we willen. Om toe te staan dat er iets naast het plaatje mag staan, moeten we een **float** instellen:

```

article img {
    width: 348px;
    margin: 20px;
    float: left;
}

```

We krijgen nu een goed resultaat.

De Difference Engine van Babbage



Ut varius eros id velit lobortis pharetra. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilla Curae; Etiam bibendum urna et ex mollis facilisis. In a nulla ut neque gravida euismod quis eu odio. Ut nec dictum velit. Vestibulum vitae rhoncus est. Morbi commodo in metus eu lacinia. Nunc dignissim ut metus ac tempus. Sed consequat placerat justo, eu pharetra sapien pharetra dapibus. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et.

Het woord left achter float betekent dat het plaatje links wordt uitgelijnd. Als je

```
float: right;
```

opgeeft in plaats van een float: left;, wordt het plaatje rechts uitgelijnd.

De Difference Engine van Babbage

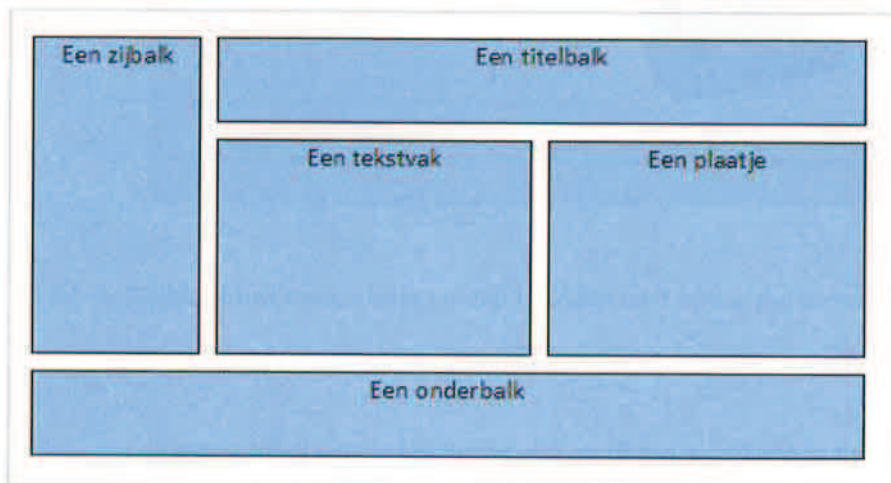
Ut varius eros id velit lobortis pharetra. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilla Curae; Etiam bibendum urna et ex mollis facilisis. In a nulla ut neque gravida euismod quis eu odio. Ut nec dictum velit. Vestibulum vitae rhoncus est. Morbi commodo in metus eu lacinia. Nunc dignissim ut metus ac tempus. Sed consequat placerat justo, eu pharetra sapien pharetra dapibus. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et.



Index Met floating kun je nog veel meer dan tekst om plaatjes heen zetten. Een van de belangrijkste toepassingen is dat je er blokken mee op de juiste plek kunt zetten. Hoe dat gaat, leer je in de volgende paragraaf. Daar wordt ook uitgelegd wat floating precies is.

5.11 De blokken op de juiste plaats

Blokken wil je vaak naast elkaar **positioneren** en niet alleen onder elkaar. Dat gaat niet vanzelf. Om uit te leggen hoe je dat moet doen, bekijken we onderstaand ontwerp.

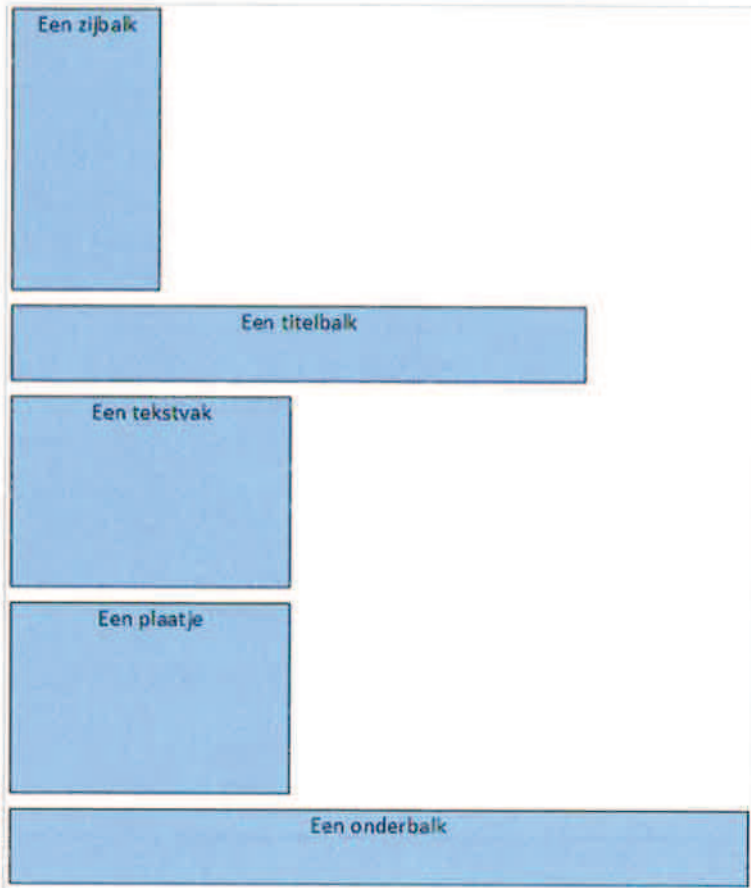


We gaan dit ontwerp realiseren met HTML- en CSS-code en we beginnen met de code in HTML waarin we de blokken neerzetten:

<pre> <div id="sidebar">Een zijbalk</div> <header>Een titelbalk</header> <article>Een tekstvak</article> <div id="picture">Een plaatje</div> <footer>Een onderbalk</footer> </pre>	<pre> #sidebar { width: 100px; height: 191px; } header { width: 392px; height: 50px; } article { width: 190px; height: 129px; } #picture { width: 190px; height: 129px; } footer { width: 504px; height: 50px; } </pre>
--	---

Verder geven we van alle blokken in CSS een breedte en hoogte op. De blokken hebben ook een achtergrondkleur en border, maar de CSS-regels daarvoor laten we even weg voor de overzichtelijkheid.

Het resultaat van deze HTML- en CSS-code zie je hieronder.



Zoals je kunt zien, staan alle blokken onder elkaar. HTML-elementen worden altijd automatisch onder elkaar geplaatst. Om ze toch naast elkaar te zetten, gebruiken we, net als in de vorige paragraaf, floating. Float is Engels voor 'drijven'. Door de regel:

```
float: left;
```

toe te voegen aan de stijlregel van een blok, 'drijft' dat blok naar linksboven tot het tegen een ander drijvend blok 'botst' of tegen de linkerbovenhoek van het blok aan komt waar in het genest is.

We gaan voor #sidebar, header, article en #picture een float opgeven.


```

#sidebar {
  width: 100px;
  height: 191px;
  float: left;
}

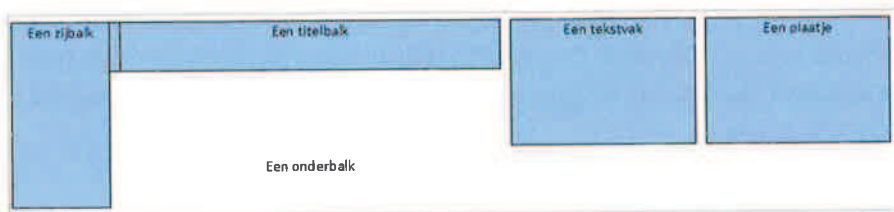
header {
  width: 392px;
  height: 50px;
  float: left;
}

article {
  width: 190px;
  height: 129px;
  float: left;
}

#picture {
  width: 190px;
  height: 129px;
  float: left;
}

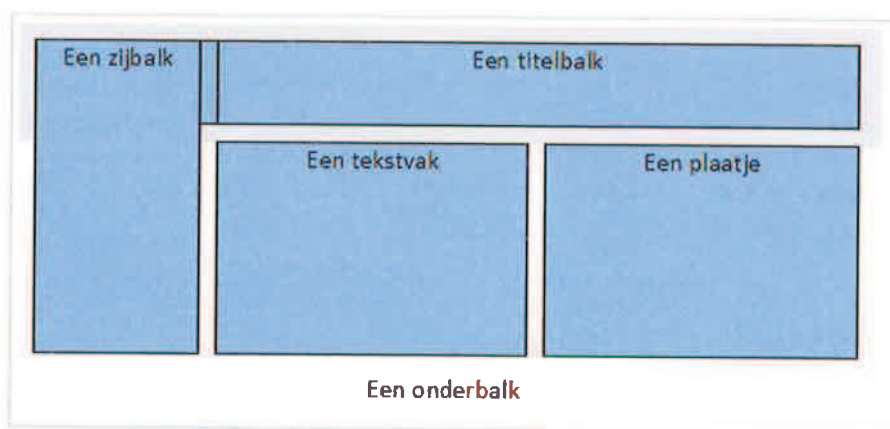
```

Het resultaat is dit:



De blokken komen nu naast elkaar te staan in plaats van onder elkaar, maar het ziet er nog niet uit zoals we willen. We hebben namelijk geen grens opgegeven voor de breedte van de website. Daarom kunnen de blokken onbeperkt naast elkaar drijven. We kunnen dat probleem eenvoudig oplossen door alle blokken in een wrapper te plaatsen waarvan we de breedte opgeven. We geven de wrapper ook tijdelijk een achtergrondkleur, zodat we de positie van wrapper zelf ook goed kunnen zien.

<pre><div id="wrapper"> <div id="sidebar">Een zijbalk</div> <header>Een titelbalk</header> <article>Een tekstvak </article> <div id="picture">Een plaatje</div> <footer>Een onderbalk</footer></pre>	<pre>#wrapper { width: 520px; background-color: #ddddd; } header { ... }</pre>
--	---



We zien nu dat vier van de vijf blokken al goed staan. Maar we zien ook iets bijzonders: het lijkt wel of er drie blokken niet helemaal in de wrapper zitten en we zien de onderbalk ónder de titel- en zijbalk. De reden hiervoor is dat de **drijvende blokken** 'bovenop' de niet-drijvende blokken terecht komen. De tekst van de footer staat wél op de goede plaats, maar de footer is onder de titelbalk en de zijbalk beland.

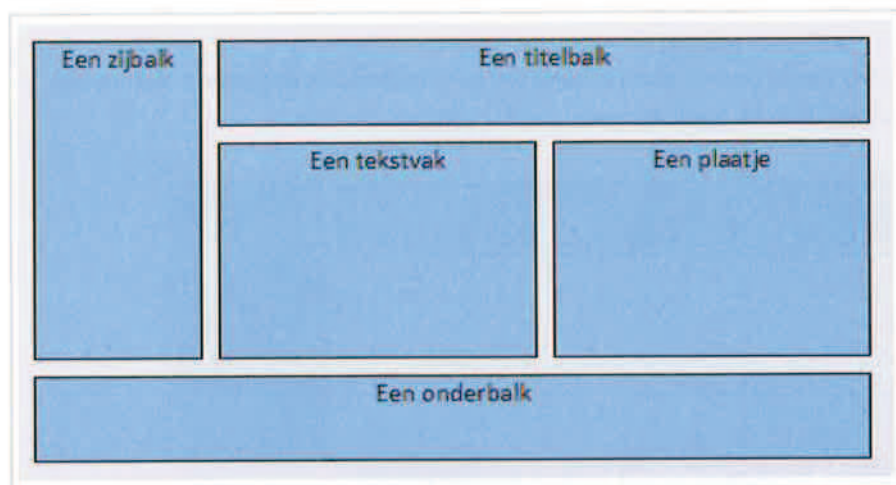
Je zou dit probleem kunnen oplossen door bij footer ook `float: left; op te geven`. Toch gaan we dat niet doen. Na het blok `#picture` hebben we geen float meer nodig. De onderbalk beslaat namelijk de hele breedte van deze website, er hoeft niets naast. In dit geval kun je het beste het effect van het drijven uitschakelen voor de onderbalk (en eventuele blokken daaronder). Dat doe je door de regel:

```
clear: both;
```

toe te voegen aan de stijlregel voor footer. Als je voor één of meerdere blokken float instelt, moet er ergens in je HTML-code een (blok-)element zijn waarin je de float weer opheft.

```
footer {  
  width: 504px;  
  height: 50px;  
  clear: both;  
}
```

De website is nu precies geworden zoals we wilden:



5.12 Left en right floating

Door `float: left` in te stellen, worden de drijvende blokken links uitgelijnd. Als je wilt dat een blok rechts wordt uitgelijnd, geef dan `float: right`; op.

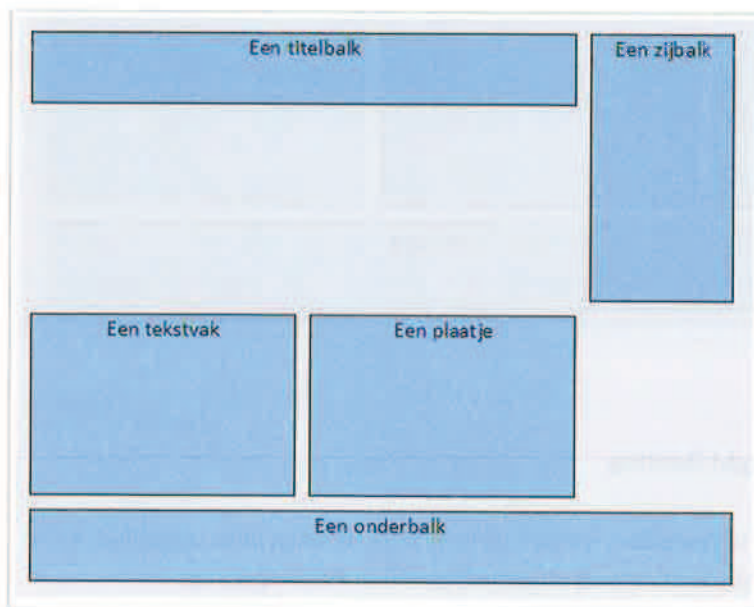
Met `clear: left`; en `clear: right`; kun je links en rechts uitgelijnde float afzonderlijk uitschakelen.

5.13 Een volgorde voor de blokken kiezen

We gaan terug naar de HTML-code van het voorbeeld van de drijvende blokken. Stel nu eens dat we niet #sidebar, maar header als eerste neerzetten, dus zo:

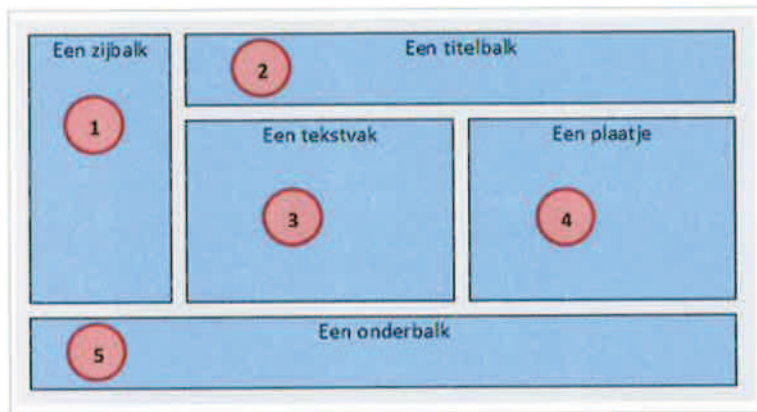
```
<div id="wrapper">  
  <header>Een titelbalk</header>  
  <div id="sidebar">Een zijbalk</div>  
  <article>Een tekstvak</article>  
  <div id="picture">Een plaatje</div>  
  <footer>Een onderbalk</footer>  
</div>
```

Het ligt eigenlijk ook voor de hand om een titelbalk eerder neer te zetten dan een zijbalk. Toch geeft dat geen goed resultaat:



De volgorde van de blokken in je HTML-code is dus heel belangrijk voor het resultaat! Hoe kun je nu de juiste volgorde bepalen?

Als je een float: left instelt, komen de blokken zover als mogelijk linksboven terecht. De goede volgorde van de blokken in HTML is in dat geval: van links naar rechts en van boven naar beneden.



Opmerking

Door het opheffen van de float in footer, vervalt ook de margin-top van dat blok. Als je bijvoorbeeld:

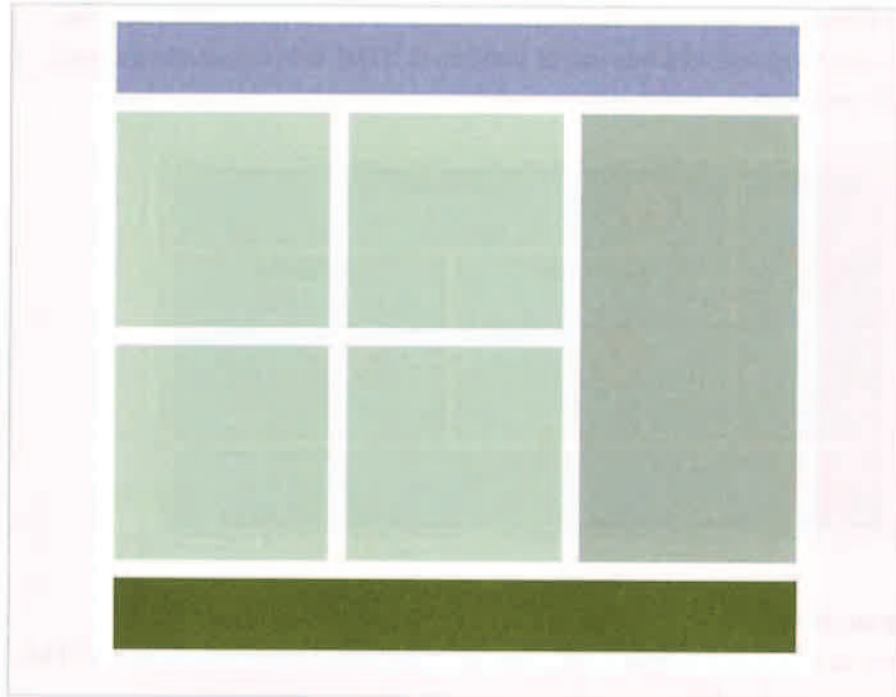
```
margin-top: 100px;
```

op zou geven in footer, verandert er helemaal niets. Om toch ruimte tussen de blokken te maken, kun je de margin-bottom van één van de blokken erboven vergroten.

5.14 Opdrachten

5.14.1 Opdracht 1

In deze opdracht ga je oefenen met het positioneren van blokken. Je gaat de volgende (tekstloze) webpagina maken:



1. Maak een nieuw HTML-bestand en een nieuw CSS-bestand aan. Zorg ervoor dat het CSS-bestand gekoppeld is aan het HTML-bestand.
2. Zet in de body een division neer met de naam "wrapper"
3. Plaats in de division wrapper de volgende blokken:
 - een header
 - vier article-elementen
 - een division met de naam "sidebar"
 - een footer
4. Neem de volgende CSS over in je CSS-bestand:

```
* {  
  margin: 0;  
}  
  
body{  
  background-color: #ffeae3;  
}
```

```
#wrapper {
  background-color: #fcfdff;
  padding: 10px 0 10px 10px;
  margin: auto;
}

header {
  background-color: #9fa2b2;
  width: 380px;
  height: 40px;
  margin: 0 10px 10px 0;
}

article {
  background-color: #c0ccb5;
  width: 120px;
  height: 120px;
  margin: 0 10px 10px 0;
  cursor: pointer;
}

article:hover {
  background-color: #ffeae3;
}

#sidebar {
  background-color: #a9b2a1;
  width: 120px;
  height: 250px;
  margin: 0 10px 10px 0;
}

footer {
  background-color: #505810;
  width: 380px;
  height: 40px;
  margin: 0 10px 0px 0;
}
```

5. Zorg er nu voor dat de blokken op de juiste plaats komen te staan. Doe dat door:
 - in de HTML-code de juiste volgorde voor de blokken te kiezen.
 - in de CSS-code float in te stellen op de juiste plaats (vergeet niet om ook clear:both te gebruiken op een goede plaats).
 - een goede breedte te kiezen van de wrapper.

5.14.2 Opdracht 2

Dit is een grote opdracht waarin je een complete website gaat maken. De opdracht bestaat daarom uit veel stappen. Lees elke stap rustig door en voer het uit. Bij een aantal opdrachten staan vragen. Beantwoord deze vragen in een apart document of in je schrift. Als je alle stappen goed hebt doorlopen, heb je een mooie website gebouwd. Je bent dan ook een aantal bekende problemen tegengekomen en je hebt geleerd hoe je die moet oplossen. In het volgende hoofdstuk kun je deze website uitbreiden en nog mooier maken.

Het resultaat zal er zo uit zien:

SUPERCOMPUTERS

Home

Meer informatie

Tijdlijn



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vehicula magna ac porta condimentum. Nulla tristique pulvinar feui. Donec ipsum lectus, molestie accumsan hendicil, net, vulputate et trinis. Quisque euismod bibendum euismod.

De krachtigste computer ooit

Ut varius eros id velit lobortis pharetra. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam bibendum urna et ex mollis facilisis. In a nulla ut neque gravida euismod quis eu odio. Ut nec dictum velit. Vestibulum vitae rhoncus est. Morbi commodo in metus eu lacinia. Nunc dignissim ut metus ac tempus. Sed consequat placerat justo, eu pharetra sapien pharetra dapibus. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et.

Nam gravida risus at euismod rhoncus. Integer lorem ex, dapibus a magna sed, sodales porttitor massa. Maecenas nec lacus dolor. Etiam nec justo justo. Donec molestie sed eros sed laoreet. Nunc euismod nulla tortor, sed pulvinar justo cursus sit amet. Praesent id varius nisi. Nullam ullamcorper quam sit amet sodales egestas. Nunc mattis neque eget volutpat vestibulum. Vestibulum nec suscipit mi.

De Difference Engine van Babbage

Ut varius eros id velit lobortis pharetra. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam bibendum urna et ex mollis facilisis. In a nulla ut neque gravida euismod quis eu odio. Ut nec dictum velit. Vestibulum vitae rhoncus est. Morbi commodo in metus eu lacinia. Nunc dignissim ut metus ac tempus. Sed consequat placerat justo, eu pharetra sapien pharetra dapibus. Ut rhoncus porta sapien sed accumsan. Duis sed lacinia purus. Pellentesque vulputate neque commodo molestie consectetur. Nulla facilisi. Suspendisse tincidunt ultrices ligula, ut vulputate risus venenatis et.



Tot slot

Ut varius eros id velit lobortis pharetra. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam bibendum urna et ex mollis facilisis. In a nulla ut neque gravida euismod quis eu odio.

Contact

© 2015 N.N. | Publicatie zonder toestemming van de auteur is wettelijk verboden.

Om te beginnen

- Je hebt bij deze opdracht twee afbeeldingen nodig: één van de supercomputer Titan 2 (<https://commons.wikimedia.org/wiki/File:Titan2.jpg>) en één van de Analytical Engine van Charles Babbage (https://de.wikipedia.org/wiki/Analytical_Engine). Vraag je docent om deze afbeeldingen.
- Maak in HTML Kit drie HTML-bestanden en sla ze op in een aparte map (bijvoorbeeld '/informatica/opdracht2css/'). Sla ze op met de volgende namen: index.html, meer_info.html en tijdlijn.html. Vergeet niet deze HTML-bestanden het HTML 5 DOCTYPE mee te geven.
- Maak één CSS-bestand en zorg ervoor dat de drie HTML-bestanden gekoppeld zijn met dit CSS-bestand. Als je dit gedaan hebt, sluit je de bestanden informatie.html en tijdlijn.html, zodat je aan het werk kunt in index.html en style.css.

De opdracht

We gaan een webpagina maken met een header, een navigatiebalk, een tekstvak en een footer.

1. Zorg in CSS-code ervoor dat het lettertype in het hele HTML-bestand 'Verdana' is en de lettergrootte 14pt.

De hoofdstructuur van de webpagina

2. We gaan eerst aan de slag met de hoofdstructuur in HTML. Plaats hiervoor onder elkaar de volgende blokken: header, nav, main, footer.
3. Plaats tussen nav en main een division (<div>) en geeft deze division de id 'bigimage' mee.

Vraag: waarom moeten we voor de bigimage-division een algemeen div-element gebruiken en voor de andere divisions niet?

4. Plaats nog een algemeen div-element tussen main en footer. Geef dit element de id 'contact'
5. Plaats in main drie article-elementen. We hebben nu de hoofdstructuur in HTML compleet.

Het opmaken van de divisions

6. Geef het header-element de achtergrondkleur #00c9fd. Zet in de header de tekst 'SUPERCOMPUTERS'. Gebruik hiervoor de kop h1.
7. Stel de breedte van de header in op 900 pixels.

Vragen:

- a. Verander de tekst in h1 in 'SUPERCOMPUTERS SUPERCOMPUTERS' en open index.html in je webbrowser. Verklein de webbrowser zó dat de tekst niet meer past. Wat gebeurt er met de tekst?
 - b. Stel de breedte van het element 'header' nu in op 60% en experimenteer met verschillende venstergroottes van de webbrowser. Wat is er veranderd?
 - c. Stel tot slot de breedte van de division eerst in op 150% en vervolgens op 3084 pixels. Wat is het verschil?
 - d. Bedenk een voordeel en een nadeel van het gebruik van percentages in plaats van een vaste breedte van een division.
 - e. Verander de tekst in h1 weer terug in 'SUPERCOMPUTERS' en stel de breedte weer in op 900 pixels.
8. Plaats alle elementen die we hebben gemaakt in een overkoepelend div-element met de naam 'wrapper'. Geef de wrapper een breedte van 900 pixels en de achtergrondkleur #ff3333.

Vraag: waarom is de achtergrondkleur van de wrapper niet te zien?

9. Geef de navigatiebalk een achtergrondkleur #cccccc en een breedte van 900 pixels. Zet er ook wat tekst in, bijvoorbeeld 'Hier komt de navigatiebalk', anders wordt de division niet weergegeven op het scherm.
10. Geef article en #contact de achtergrondkleur #cccccc, en footer de achtergrondkleur #00c9fd. Plaats in #bigimage, alle articles, #contact en de footer wat tekst, zodat de divisions zichtbaar worden. Geef alle divisions de breedte van 900 pixels.

Vraag: waarom heeft #bigimage een rode achtergrond?

Margin en padding

11. Je kunt zien dat alle divisions tegen elkaar aan geplakt zitten. Behalve de header. Daar is iets vreemds mee aan de hand. Dit komt door de margin van het h1-element. Zoals je hebt kunnen lezen in de tekst van het hoofdstuk is de standaard margin van elementen onvoorspelbaar en kan het voor vervelende situaties zorgen. Schakel de margin op alle elementen standaard uit door deze regel toe te voegen:

```
* {  
  margin: 0;  
}
```

12. Elementen kunnen ook padding hebben. Geef 'header h1' een padding van 10 pixels.

Vragen:

- Wat is het verschil tussen padding en margin?
- Geef 'header' ook een padding van 10px. Waarom is dit niet zo handig?
- Haal de padding weer weg. We zullen op verschillende plaatsen margin en padding toevoegen.

Details header en nav

13. We gaan de website nu per division mooi maken. We beginnen bij header: stel de kleur van de kop h1 in op `rgb(255,255,255)` en maak de lettergrootte 30 punten. De kop moet het lettertype 'Impact' krijgen en niet vetgedrukt zijn. Let op: pas de opmaak van de kop h1 alleen aan voor het element 'header', niet in het algemeen! Verander als laatste ook nog de padding. In plaats van 10 pixels aan alle zijden moet de kop h1 20 pixels padding aan de bovenkant, en 5 pixels aan de overige drie zijden.

Vraag: wat is `rgb(255,255,255)` voor kleur? En `rgb(255,0,0)`?

14. Vervang de tekst in het nav-element door drie a-elementen. De eerste is Home (verwijst naar `index.html`), de tweede Meer informatie (linkt naar `meer_info.html`) en de derde Tijdlijn (verwijst naar `tijdlijn.html`).

15. Geef de a-elementen in nav(!) het lettertype 'Impact', en verander verder: lettergrootte - 15 punten, tekstkleur - zwart en haal de onderstreping weg.
16. We gaan er echte knoppen van maken. Geef de a-elementen de volgende opmaak: display: inline-block; Nu worden a-elementen opgemaakt zoals een division. Geef ze daarom de afmeting 200 bij 50 pixels (dus width: 200px; en height: 50px;). Lijn de tekst uit in het midden.
17. We willen de tekst ook verticaal centreren. Dit doen we met de eigenschap 'line-height'. Geef de a-elementen evenveel line-height als de hoogte van de knoppen (50 pixels).
18. Stel in dat de a-elementen in nav een witte achtergrondkleur hebben als je er met de muis overheen gaat.

Details #bigimage

19. Voeg het plaatje van supercomputer Titan toe aan het HTML-bestand. Zorg dat de breedte van het plaatje gelijk is aan de breedte van de division. Voeg ook een p-element toe aan #bigimage met daarin drie regels Lorem ipsum tekst, die je van het internet haalt.
20. Geef het p-element in #bigimage de tekstkleur #00c9fd en de achtergrondkleur #ffffff. Misschien zie je nu een dun lijntje zichtbaar tussen het plaatje en het p-element. Dit is een bug in sommige webbrowsers. Je kunt dit verwijderen door img-elementen 'display: block;' als opmaak mee te geven.
21. Geef het p-element in #bigimage een lettergrootte van 12 punten. Maak het eerste woord in het p-element van #bigimage dikgedrukt en plaats na dit woord een enter.
22. Geef alle p-elementen in het hele document een padding van 20 pixels.

Details van main en article

23. In alle article-elementen moet het volgende worden toegevoegd: Een h1-element en twee p-elementen. Bedenk bij de drie h1-elementen zelf goede titels en vul alle p-elementen met minimaal 10 regels lorem ipsum tekst.
24. Geef de h1-elementen in article een padding van 10 pixels, lettertype Impact en lettergrootte 25 punten. Centreer de tekst in h1. Geef het element article aan de boven- en onderkant een margin van 20 pixels.

25. Plaats het plaatje van de Analytical Engine in het middelste article. Geef in HTML alleen de breedte op: 348 pixels. Geef alle plaatjes in de article-elementen de opmaak 'float: right;' en een margin van 20 pixels aan alle zijden.
26. Kies in het bovenste article-element een woord uit, en zorg ervoor dat dit woord een hyperlink wordt. De hyperlink moet verwijzen naar de Wikipedia-pagina over supercomputers.
27. Geef alle hyperlinks die onderdeel zijn van een paragraaf (dus alle a-elementen die in een p-element zijn genest) de volgende opmaak: zorg dat de woorden dikgedrukt zijn, de kleur #00c9fd hebben en niet zijn onderstreept. Codeer ook dat de lettergrootte 5% groter en zwart wordt, als je over het woord heengaat met de muis.

Details van #contact en footer

28. Vervang de tekst in #contact door het woord 'Contact'. Maak van dit woord een hyperlink in een p-element. Laat de hyperlink verwijzen naar de nog niet bestaande pagina contact.html.

Vraag: welke drie regels opmaak zijn op het woord 'Contact' van toepassing?

29. Zet in de footer een p-element met daarin de regel '© 2015 [jouw naam] | Publicatie zonder toestemming van de auteur is wettelijk verboden'. Maak de tekst in p wit.

Vraag: moeten we ook een padding toevoegen? Waarom wel/niet?

30. Geef aan dat de hoogte van footer 150 pixels moet zijn.

Vraag: waarom zouden we deze hoogte toevoegen?

31. Verwijder de rode achtergrondkleur van de wrapper door er commentaar van te maken (verwijder dus de opmaak, en niet de regel uit het css-document!) en voeg in plaats daarvan de regel 'margin: auto' toe. Nu is het een mooie website geworden.

Vragen:

- a. Waarom hebben we tijdens de hele opdracht de achtergrond van de wrapper rood gemaakt, terwijl bij het eindresultaat deze kleur weer weg moet?
 - b. Waarom is een wrapper zo handig als je een website in zijn geheel wilt centreren?
-
32. Check of je resultaat hetzelfde eruit zien als de uitwerking op het plaatje. Bedenk een bijpassende titel voor het title-element in de head en voeg die toe.
 33. Controleer je complete html document met F9, en je CSS-bestand eventueel met een online validator. Bijvoorbeeld op https://validator.w3.org/#validate_by_input.

