

CSS voor gevorderden

6.1 Inleiding

De behandelde stof in hoofdstuk 1 tot en met 5 kan worden gezien als een goede basis voor het beheersen van HTML en CSS. In dit hoofdstuk komen een aantal onderwerpen aan de orde die meer door de gevorderde webdesigner kunnen worden toegepast. Dit kan gaan over kleine trucs, zoals het hr-element (6.2), of het toevoegen van schaduw (6.3), maar er staan ook lastige onderwerpen in zoals responsive design (6.4).



6.2 Het hr-element

ent

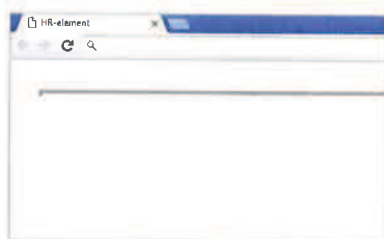
Je kunt met HTML-code een horizontale lijn weergeven op een webpagina. Dit is bijvoorbeeld handig als je een 'break' wilt tussen bepaalde stukken inhoud op een webpagina. Je kunt het ook gebruiken om het design van je webpagina te optimaliseren, zoals in dit voorbeeld; hier is onder de koptekst een horizontale lijn geplaatst om de koptekst duidelijker af te laten steken tegen het menu van de website:



Je voegt een horizontale lijn toe met het HTML-element `hr`. Net als het `br`-element en het `img`-element heeft een `hr`-element alleen een open-tag en geen sluit-tag.



Een `hr`-element is van zichzelf 100% breed, heeft een hoogte van 1 pixel en een schaduwrand aan de onderkant. Dit kun je goed zien als je inzoomt.

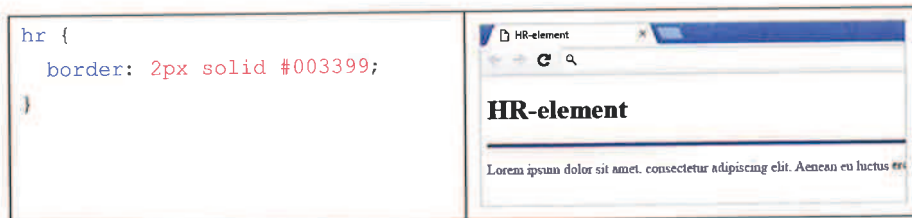


De CSS-opmaak van een `hr`-element is nagenoeg gelijk aan de opmaak van de border van bijvoorbeeld een tabel. Technisch gezien is een `hr`-element een 'losstaande' border.



I

Box-shadow

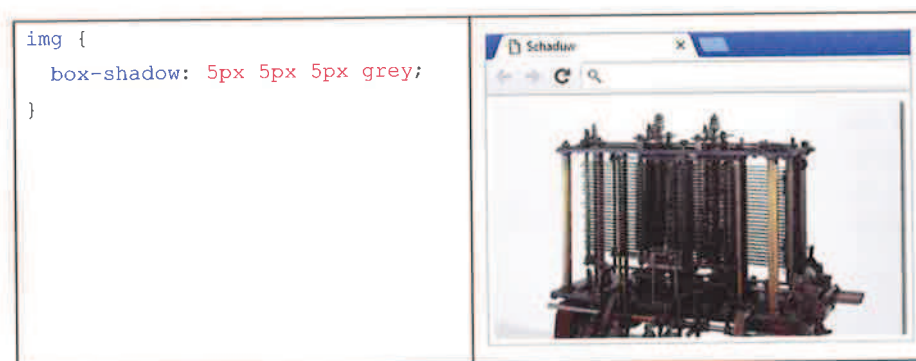


6.3 Schaduw

Met de CSS-code 'border' kun je de dikte, kleur en lijnstijl van een rand bepalen. Je kunt een rand ook schaduw geven. Dit kan bij verschillende HTML-elementen, zoals een division, een horizontale lijn of een afbeelding. Je kunt schaduw opgeven in CSS-code met de eigenschap box-shadow.

De eigenschap **box-shadow** heeft, net als de eigenschap border, verschillende waarden, namelijk voor:

h-shadow	horizontale schaduw (in px.)
v-shadow	verticale schaduw (in px.)
blur	de afmeting van de schaduw (in px.)
color	de kleur van de schaduw

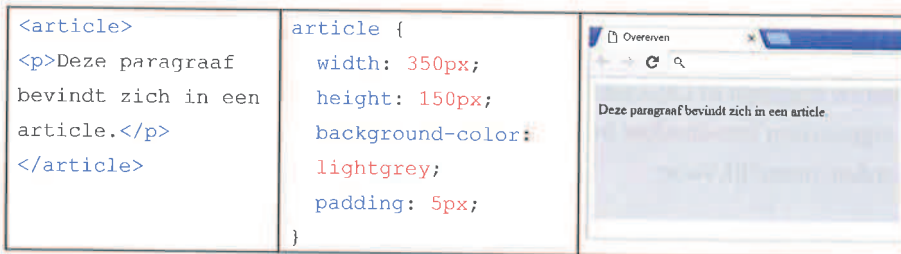


6.4 Overerving

6.4.1 Overerving

In paragraaf 5.4 heb je gezien hoe je met CSS-code opmaak kunt geven aan geneste HTML-elementen. Zo kun je bepaalde HTML-elementen een specifieke opmaak geven. Geneste HTML-elementen nemen automatisch de opmaak over van het HTML-element waarin ze zich bevinden. Dit noemen we **overerving**.

Kijk eens naar dit voorbeeld.



Je ziet dat in het article-element een paragraaf zit. Omdat het article-element een achtergrondkleur heeft (lichtgrijs), heeft de paragraaf automatisch ook een grijze achtergrond gekregen.

Je kunt de achtergrondkleur van de paragraaf natuurlijk ook aanpassen:

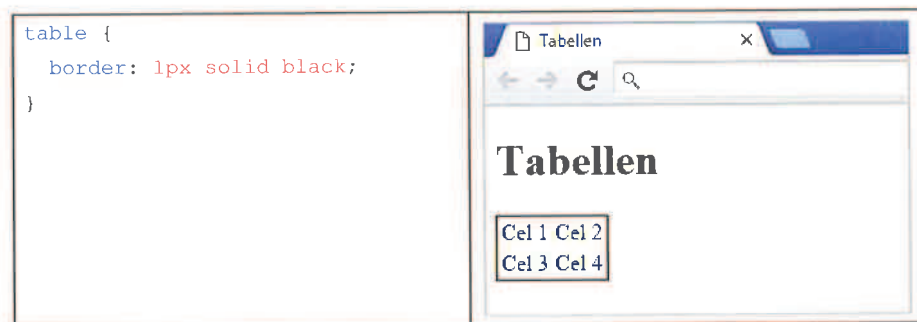


Je hebt eigenlijk al vaak gebruik gemaakt van overerving, zonder dat je wist wat het was. In de meeste gevallen is overerving precies wat je wilt: in de

bovenstaande situatie is het goed dat de paragraaf in het article automatisch dezelfde achtergrondkleur krijgt als het article zelf. Maar soms gaat dit overerven niet zoals je verwacht. Twee belangrijke voorbeelden hiervan zijn tabellen en hyperlinks. Daarover lees je hierna meer.

6.4.2 Tabellen

In hoofdstuk 4 is de opmaak van tabellen al aan de orde gekomen. Bekijk het volgende plaatje.



De CSS-code, geeft de tabel een doorgetrokken, zwarte rand met een dikte van 1 pixel.

Omdat de td-elementen genest zijn in het table-element, zou je verwachten dat ze de border-eigenschappen van table zouden overerven. Je zou dus verwachten dat de cellen én de tabel ook een doorgetrokken, zwarte rand met een dikte van 1 pixel zouden hebben. Maar dit gebeurt niet. Dat komt omdat een td-element van zichzelf al opmaak heeft voor de rand, namelijk dat er geen rand is.

De standaardopmaak die een HTML-element van zichzelf heeft, weegt zwaarder dan de opmaak die door overerving wordt verkregen.

Wil je de rand van de cellen opmaak geven met CSS-code, dan moet je die apart opgeven met een tag-selector voor td in de CSS-code.

```
table, td {  
  border: 1px solid black;  
}
```

6.4.3 Hyperlinks

Ook hyperlinks hebben van zichzelf al wat opmaak. Zo is de tekstkleur van een hyperlink standaard blauw en is een hyperlink onderstreept.

Bekijk het volgende voorbeeld:



De hyperlink bevindt zich in de paragraaf. Aan de paragraaf wordt door middel van CSS-code een zwarte tekstkleur gegeven. De hyperlink blijft echter een blauwe tekstkleur houden. Ook in dit geval weegt de standaardopmaak van de hyperlink zwaarder dan de opmaak die de hyperlink door middel van overerving krijgt.

Om de hyperlink toch een zwarte tekstkleur te geven, kan de selector 'article p' worden vervangen door:

```
article p, article a {  
  color: black;  
}
```

Hiermee is de CSS-code voor de zwarte tekstkleur van toepassing op een paragraaf in een article én voor een hyperlink in een article.

6.5 Id's en classes

6.5.1 Id's en classes

In hoofdstuk 5 leerde je om een 'id' toe te voegen aan een div-element om het een unieke naam te geven. Hierdoor kun je verschillende div-elementen een eigen style geven. Bekijk de onderstaande code:

<pre><article> <p>Rode article tekst</p> <p>Groene article tekst</p> </article> <article id="greentext"> <p>Groene article tekst</p> </article></pre>	<pre>article { color: red; } #greentext { color: green; }</pre>
--	--

In plaats van een div-element is er nu een article-element met een id. Hierdoor geef je alle tekst in het tweede article-element een groene kleur. De id heeft om die reden de naam 'greentext'.

Denk goed na over het kiezen van een juiste naam als id. De naam die je kiest moet iets te maken hebben met het doel van de stijlregel. Door het geven van namen kun je de opmaak per element laten afwijken.

Als je goed naar de code kijkt, zie je dat er iets niet klopt. In het eerste article-element staat namelijk nog een p-element waarvan de tekstkleur ook groen moet zijn. Met de code zoals die er nu staat, wordt deze tekst rood. Een oplossing hiervoor zou kunnen zijn om ook deze paragraaf het id 'greentext' te geven. Helaas is dat niet toegestaan in CSS. Als je eenzelfde naam meerdere keren wilt gebruiken, mag je geen gebruik maken van een id-selector. In plaats daarvan moet je een class-selector gebruiken:

<pre> article { color: red; } .greentext { color: green; } </pre>	<pre> <article> <p>Rode article tekst</p> <p class="greentext">Groene article tekst</p> </article> <article class="greentext"> <p>Groene article tekst</p> </article> </pre>
--	---

Een **class-selector** werkt nagenoeg hetzelfde als een **id-selector**. In de HTML-code moet je het woord 'id' aanpassen in 'class' en in de CSS-code verwijst je naar een class met een punt (.) in plaats van een #.

Houd in gedachten dat voor een class (en ook voor een id) overerving van toepassing blijft. Je kunt hierboven bijvoorbeeld zien dat de class 'greentext' in het tweede article-element zorgt voor groene tekst in de in article geneste elementen.

Een belangrijke regel bij het gebruik van id- en class-selectors is dat je een id-selector gebruikt als je zeker weet dat je de naam maar één keer gaat gebruiken. Een class-selector gebruik je voor stijlregels die vaker toegepast worden. Het is dan logisch dat een div-element eigenlijk altijd een id-selector heeft. Een selector om meerdere elementen een groene tekstkleur te geven, zoals in bovenstaande voorbeeld, moet een class-selector zijn.

6.5.2 Een element met meerdere id's en classes

Een element kan in HTML-code zowel een id als een class meekrijgen. Het is zelfs mogelijk om elementen meerdere id's en classes mee te geven. Het is belangrijk daarbij te onthouden dat de CSS-code die hoort bij de id-selector voorrang heeft boven de CSS-code die hoort bij de class-selector. In onderstaand voorbeeld zal de tekst een blauwe kleur krijgen. Je kunt je voorstellen dat dit soort opmaak niet erg netjes is. Als je zorgvuldig omgaat met de opmaak wil je voorkomen dat er tegenstrijdige stijlregels op één element worden toegepast. De class greentext is in dit voorbeeld overbodig en kan in de HTML-code worden verwijderd.

<pre><article id="bluetext" class="greentext"> <p>article tekst</p> </article></pre>	<pre>#bluetext { color: blue; } .greentext { color: green; }</pre>
--	---

Bekijk het volgende voorbeeld:

<pre><article class="blue"> <p>article tekst</p> </article> <article class="red"> <p>article tekst</p> </article> <article class="blue red"> <p>article tekst</p> </article></pre>	<pre>article { font-weight: bold; } .blue { background-color: aqua; } .red { color: red; }</pre>	<p>article tekst</p> <p>article tekst</p> <p>article tekst</p>
--	--	--

Je ziet hier een deel van een website waar veel elementen een aquablauwe achtergrondkleur of een rode tekstkleur krijgen en soms allebei. Om snel te kunnen wisselen tussen deze kleuren is er voor die twee stijlen een class aangemaakt. Door de classes te combineren kunnen bepaalde elementen ook beide stijlen krijgen. Dit is een voorbeeld van een goede manier om een element twee classes mee te geven.

6.5.3 Veelgebruikte classes

Naast het bovenstaande voorbeeld, waar twee soorten opmaak met weinig moeite vaak kunnen worden toegepast, zijn er nog twee veelgebruikte classes die erg handig kunnen zijn.

De eerste veelgebruikte class is 'focus'. Die is bedoeld om in een navigatiebalk aan te geven op welke pagina de gebruiker is. Hier zie je een voorbeeld:

	
<pre><header> <h1>SUPERCOMPUTERS</h1> </header> <nav> Home Meer informatie Tijdslijn </nav></pre>	<pre>.focus { color: #00c9fd; }</pre>

Webdesigners gebruiken vaak de class 'clearfloat'. Het doel van die class is het uitschakelen van de float (zie paragraaf 5.11). Het is eenvoudiger om in HTML aan alle elementen waarvan de float moet worden opgeheven dezelfde class toe te voegen, dan om in CSS bij al deze elementen apart de float op te heffen. Bij een kleine eenvoudige website is dit geen probleem, maar bij een grote, complexe website kost dit veel tijd. Voorbeeld in code:

<pre><article class="floatright">Tekst</article> <article class="floatright">Tekst</article> <article class="clearfloat">Tekst</article></pre>	<pre>.floatright { float: right; } .clearfloat { clear: both; }</pre>
--	---

6.6 Responsive webdesign

6.6.1 Responsive webdesign

Doordat mensen websites bezoeken met verschillende apparaten, zoals telefoons, tablets of laptops is er voor website-ontwerpers een extra uitdaging. Al deze apparaten hebben een verschillende schermafmeting; het scherm van een telefoon is kleiner dan dat van een laptop. Hier moet je als ontwerper op een handige manier mee omgaan.

Als je een website bezoekt met je telefoon wil je niet dat je veel (horizontaal) moet scrollen en juist wel dat het lettertype iets groter is dan als je een website met je laptop bezoekt.

6.6.2 Mediatypes

Met CSS versie 3 is het mogelijk om (delen van) CSS-code alleen in bepaalde gevallen te activeren. Zo kun je de printversie van je website een andere opmaak geven. Dit kan handig zijn, bijvoorbeeld als je wilt dat bij het printen bepaalde elementen, zoals het menu, niet zichtbaar zijn.

Om hiermee aan de slag te gaan, moeten we in de CSS-code aangeven voor welke situatie de stijlregels zijn bedoeld. Dit kan met de code '@media'. Achter @media kunnen verschillende waarden worden geplaatst. Afhankelijk van het apparaat waarop de website wordt bekeken, kiest de webbrowser dan zelf de CSS-code dat er gebruikt moet worden.

all	voor alles
print	voor printers
screen	voor schermen, computers, tablets, telefoons e.d.

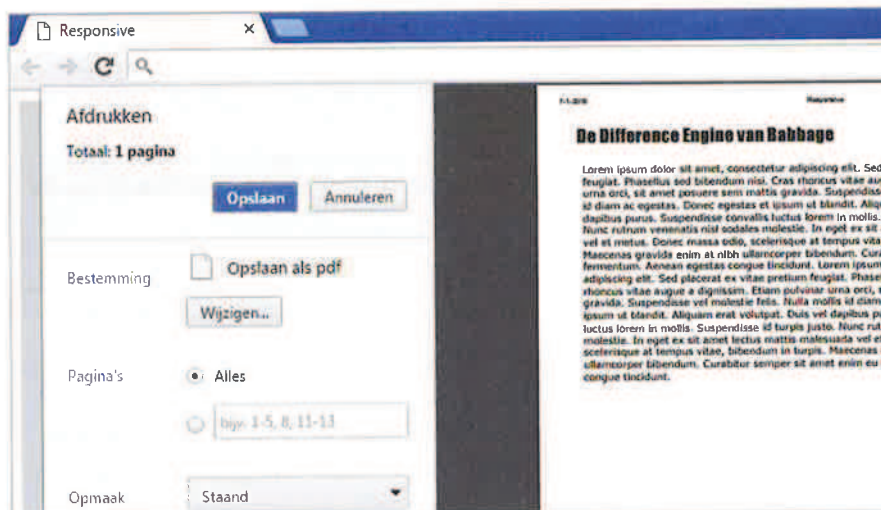
Als voorbeeld staat hier een deel van een webpagina die ook in hoofdstuk 5 gebruikt is.



De inhoud van de kop h1 op deze webpagina is gecentreerd. Op een webpagina is dit mooi, maar als geprinte pagina is het beter om de kop, net als de paragraaf, links uit te lijnen. De CSS-code hiervoor is als volgt:

```
@media all {  
  h1 {  
    margin: 0;  
    font-family: Impact;  
    padding: 10px;  
    font-size: 25pt;  
  }  
}  
  
@media screen {  
  h1 {  
    text-align: center;  
  }  
}  
  
@media print {  
  h1 {  
    text-align: left;  
  }  
}
```

Tijdens het opvragen van een afdrukweergave zie je dat de tekst van de kop h1 bij het afdrukken links zal worden uitgelijnd, terwijl bij het bekijken van de webpagina zelf de tekst van de kop h1 is gecentreerd.



In bovenstaande voorbeeld zie je dat bij het uitprinten van een website sommige opmaak automatisch wordt weggelaten, zoals de opmaak van de achtergrond van de webpagina of een division. Dit doen webbrowsers automatisch om inkt te besparen en de leesbaarheid te bevorderen. Het is mogelijk om delen van je webpagina onzichtbaar te maken. Dat kan met de CSS-eigenschap `display`. De waarde `'none'` maakt een element onzichtbaar. In onderstaande voorbeeld is de afbeelding onzichtbaar geworden.

<pre><article> <h1>De diff...</h1> <p>...</p> </article></pre>	<pre>img { display: none; }</pre>	
--	-------------------------------------	--

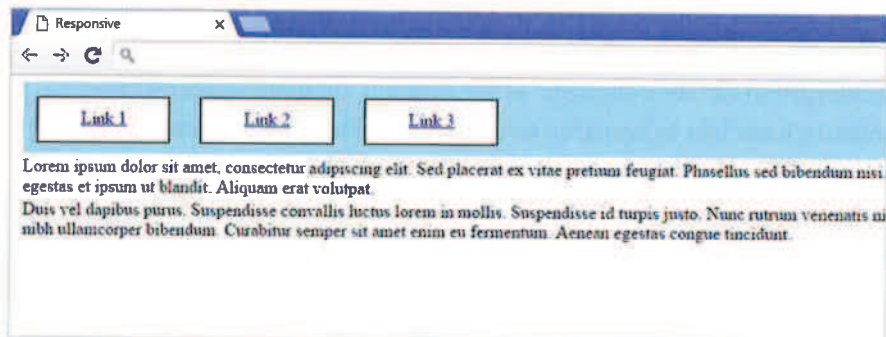
6.6.3 Media queries

es

Naast het opgeven of de opmaak in het CSS-bestand van toepassing moet zijn op de webpagina of de printversie, is het ook mogelijk om een aantal specifieke details over het scherm en het apparaat van de bezoeker te detecteren. Dit worden **media queries** genoemd. Je kunt hiermee onder andere de volgende informatie opvragen:

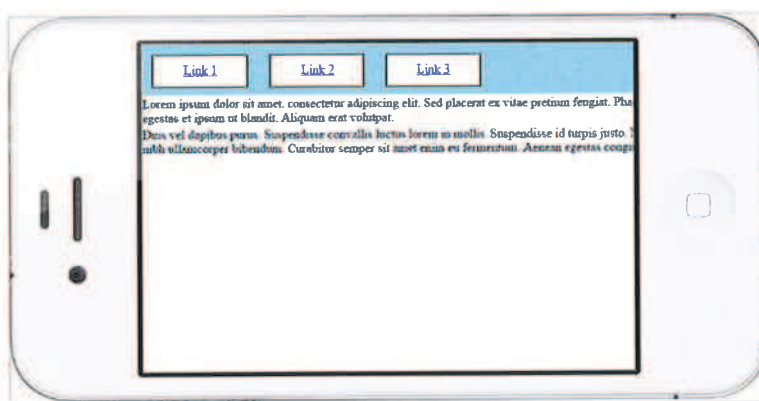
device-height	hoogte van het apparaat van de bezoeker, bijvoorbeeld van de telefoon (in px)
device-width	breedte van het apparaat van de bezoeker
orientation	of de bezoeker zijn telefoon/tablet horizontaal of verticaal houdt

Hieronder zie je een deel van een eenvoudige webpagina, met een horizontaal menu met knoppen en enkele paragrafen. De screenshot is genomen vanaf een computer:



<pre> <div id="wrapper"> <nav> Link 1 Link 2 Link 3 </nav> <p>Lorem ipsum...</p> <p>Duis vel ...</p> </div> </pre>	<pre> #wrapper { width: 1280px; } nav { width: 100%; height: 45px; background-color: lightblue; padding: 10px; } a { float: left; width: 100px; height: 20px; background-color: white; border: 1px solid black; padding: 10px; text-align: center; margin-right: 25px; } </pre>
---	---

Als je deze webpagina op een telefoon zou bekijken, in liggende houding, dan beslaat het menu bijna een kwart van het scherm. Als het menu aan de linkerkant verticaal weergegeven zou worden, blijft er meer ruimte voor de tekst over.



Met de CSS3 media queries is het mogelijk om de oriëntatie van het toestel op te vragen. Als de oriëntatie liggend is (landscape) is het mogelijk om andere CSS-code te gebruiken.

Een media query plaats je in het CSS-bestand als stijlregel en ziet er bijvoorbeeld als volgt uit:

```
@media screen and (orientation:landscape) {  
}
```

Binnen de accolades kun je de CSS-code plaatsen die alleen wordt gebruikt als het apparaat van de bezoeker 'landscape' is. Een computerscherm is altijd landscape, maar een telefoon of een tablet kan ook 'portrait' (rechttop) zijn. Door aparte code op te geven voor 'portrait' en 'landscape' wordt het menu altijd optimaal weergegeven. De volledige CSS-code is dan als volgt:

```
@media screen and (orientation:portrait) {  
  #wrapper {  
    width: 100%;  
  }  
  
  nav {  
    width: 100%;  
    height: 45px;  
    background-color: lightblue;  
    padding: 10px;  
  }  
  
  a {  
    float: left;  
    width: 100px;  
    height: 20px;  
    background-color: white;  
    border: 1px solid black;  
    padding: 10px;  
    text-align: center;  
    margin-right: 25px;  
  }  
}
```



```

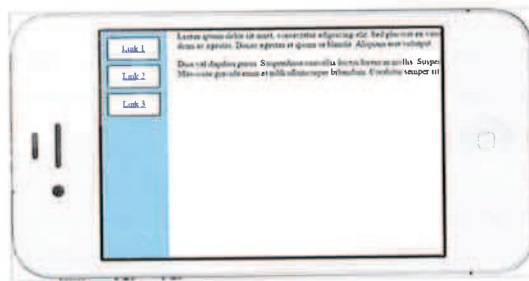
@media screen and (orientation:landscape) {
  #wrapper {
    width: 100%;
  }

  nav {
    width: 100px;
    height: 400px;
    background-color: lightblue;
    padding: 10px;
    float: left;
    margin-right: 20px;
  }

  a {
    float: left;
    width: 75px;
    height: 20px;
    background-color: white;
    border: 1px solid black;
    padding: 10px;
    text-align: center;
    margin-bottom: 10px;
  }
}

```

Door deze CSS-code te gebruiken wordt de webpagina op een telefoon of tablet altijd optimaal weergegeven.



6.7 Opdrachten

6.7.1 Opdracht 1

In deze opdracht gaan we verder met de website over Supercomputers (opdracht 2 van hoofdstuk 5). We maken de website compleet. Als de docent hier de mogelijkheid voor heeft, kan de website daarna geüpload worden. Ook in deze opdracht komen vragen voor. Deze vragen kun je toevoegen aan de antwoorden op de vragen van de vorige opdracht over Supercomputers.

Om te beginnen

- Open `index.html`, `meer_info.html`, `tijdljn.html` en `style.css`.

De opdracht

Het toevoegen van subpagina's.

We hebben in de vorige opdracht in de navigatiebalk drie knoppen gemaakt, maar als je erop klikte ging je naar een lege subpagina. Nu gaan we de navigatiebalk mooi werkend maken.

1. Kopieer de volledige inhoud van `index.html` in de andere twee html-documenten. Je hebt dan dus drie keer dezelfde pagina, maar onder een andere naam.
2. Verwijder uit `meer_info.html` en `tijdljn.html` twee van de drie `article`-elementen. Verander de `h1`-koppen van het nog overgebleven `article`-element op beide nieuwe pagina's in de woorden 'Meer informatie' en 'Tijdljn'. Als je het leuk vindt, kun je de afbeelding van de twee nieuwe pagina's ook veranderen.
3. Als het goed is, werkt de navigatiebalk nu al goed. Test dit uit.

Er is echter nog een probleem. Je kunt aan de navigatiebalk helemaal niet zien op welke pagina je bent! Dit is op veel websites wel het geval, en dit willen we graag toevoegen.

Dezelfde elementen anders opmaken

We gaan het volgende maken: als je op `index.html` bent, willen we dat het woord

'Home' blauw is in plaats van zwart en als je op meer_info.html bent, willen we dat 'Meer informatie' blauw is. De andere twee knoppen moeten zwart blijven.

4. Selecteer het bestand index.html. Voeg aan het a-element van 'Home' het volgende attribuut toe: class="focus". Kopieer het attribuut en voeg hem ook toe aan het a-element van 'Meer informatie' in meer_info.html, en aan het a-element van 'Tijdslijn' in tijdslijn.html. In alle drie de documenten heeft dan dus een ander a-element de class "focus".
5. Maak in CSS een class-selector aan. Dit doe je met een punt. Je verwijst dus in CSS met '.focus'. Geef .focus de tekstkleur #00c9fd. Als je nu op index.html bent, moet het woord 'Home' die kleur blauw zijn, en op Meer informatie moet 'Meer informatie' etc. die kleur hebben.

Je kunt nu meer_info.html en tijdslijn.html sluiten. We gaan index.html nog veel hipper maken! Kijk maar eens naar de uitwerking van deze opdracht.

Van #wrapper naar #headwrapper

Zoals je kunt zien zijn er elementen die de volledige breedte van de webbrowser gebruiken. Daarom moeten we de wrapper van opdracht 2 aanpassen.

6. Laat de wrapper niet afsluiten onder footer, maar onder nav, en verander de naam 'wrapper' in 'headwrapper'. Pas dit ook aan in de CSS-code.
7. Geef #headwrapper een breedte van 100% en een witte achtergrondkleur. Haal de opmaak margin:auto; weg, maar plaats die nu wel bij header en nav. Plaats vervolgens een hr-element tussen header en nav.

Vraag: Wat is een hr-element? Waarom is het hr-element breder dan header en nav?

8. Verwijder de background-color bij header, en geef header h1 een zwarte tekstkleur in plaats van witte tekstkleur. Verwijder ook de achtergrondkleur van nav. Zorg er verder voor dat de tekst in nav rechts wordt uitgelijnd.

Vraag: Waarom blijft de tekst in de knoppen gecentreerd?

9. Het `:hover` effect van `nav a` werkt nu niet meer. Wijzig het huidige effect in het veranderen van de tekstkleur in dezelfde kleur als `.focus`.

Een coverafbeelding met bijschrift

10. Het plaatje ziet er nu niet meer zo mooi uit. We gaan die ook uitrekken over de hele breedte.

Vraag: Waarom is het niet zo'n goed idee om het `img`-element 100% breed te maken?

11. Er is een mooie manier om grote afbeeldingen bij te snijden. Verwijder het `img`-element daarvoor uit het html bestand. Verwijder ook de bijbehorende stijlregel. Voeg in plaats daarvan extra opmaak toe aan `#bigimage`, namelijk: geef `bigimage` een hoogte van 550 pixels en een breedte van 100%.
12. Voeg daarna de afbeelding toe als achtergrond van de division `bigimage`. Dit doe je met de eigenschap `background-image`. Geef `background-size` de waarde `'cover'` en `background-position` de waarde `'center'`.
13. Het `p`-element in `#bigimage` staat nu niet mooi meer. Geef het `p`-element daarom een breedte van 350px.
14. Het `p`-element staat standaard links boven. We willen dit veranderen in rechts onder. Voeg daarvoor de volgende opmaak toe aan het `p`-element: `position: absolute, bottom: 30px, right: 30px`. Geef `#bigimage` `position: relative`.

Vraag: Wat gebeurt er als we de hoeveelheid tekst in `#bigimage p` verdubbelen?

Een moderne main

15. Zorg er eerst voor dat het `main`-element in het midden van de pagina komt te staan en als je dat nog niet had gedaan, dat het `main`-element een breedte heeft van 900 pixels.
16. Haal het `img`-element uit het tweede article en plaats die in een nieuwe division met de class `'bigleft'`. Plaats deze nieuwe division tussen het tweede en het derde article.
17. Geef het eerste article-element ook de class `'bigleft'` en geef het tweede article de class `'smallright'`.

18. Voeg de nieuwe classes ook toe aan de CSS-code. Geef `bigleft` een breedte van 590 pixels, en een rechtermarge van 10 pixels. Geef `smallright` een breedte van 290 pixels en een linkermarge van 10 pixels.

Hoewel er voldoende plaats is in het `main`-element, gaan de `article`-elementen helaas niet naast elkaar staan. Daarom moeten we een `float` instellen. Hierdoor gaan de elementen standaard naast elkaar staan in plaats van onder elkaar.

19. Geef `.bigleft` een `float` naar links en `.smallright` een `float` naar rechts.

De `divisions` gaan mooi naast elkaar staan, maar er gebeurt iets raars met het laatste `article`. Dit komt omdat een `float` ook altijd weer moet worden uitgeschakeld.

20. Dit doen we door aan het laatste `article`-element de regel `'clear: both;'` toe te voegen. Maak hiervoor weer een nieuwe class aan.

De `divisions` staan nu goed, maar de ruimte is nog niet mooi opgevuld. Waarschijnlijk staat het plaatje helemaal links, en is er nog een stuk witruimte tussen het plaatje. Omdat per website verschilt hoeveel tekst er wordt gebruikt, kunnen we er nooit precies voor zorgen dat alles helemaal goed aansluit. Als we dit wel willen, moeten we de `divisions` een hoogte meegeven. Dit betekent wel dat de hoeveelheid tekst per `division` beperkt is!

21. Geef het `img`-element in `bigleft` dezelfde breedte als de `bigleft` `division` zelf en geef `.bigleft` een hoogte van 515 pixels. Haal zoveel tekst uit `article` `.bigleft` weg dat de overige tekst er mooi in past.
22. Voeg in de `division` met het plaatje ook een `p`-element toe met een ondertekening. Centreer de tekst van het `p`-element.

Vraag: Waarom hebben we daar een nieuwe class voor nodig?

23. Pas de class met de `centreeer`-stijl toe op het juiste element, en geef de juiste opmaak mee.

Vraag: Wat is de totale lengte van de twee big left divisions inclusief margins?

24. Geef .smallright die lengte, zodat de divisions aansluiten. Pas ook de tekst hoeveelheid van .smallright aan op het resultaat.

Contact en footer

25. Maak de breedte van de elementen contact en footer ook schermvullend.

De link 'contact' in #contact willen we graag vervangen door iets met twee kolommen. In HTML gebruiken we daar een tabel voor.

26. Vervang alles wat in #contact staat door een tabel met twee rijen en twee kolommen. Vul de eerste cel in de eerste rij met de tekst 'Volg Supercomputers;' en de tweede cel van de eerste rij met een hyperlink: 'Ga naar de contactpagina'. Laat de hyperlink verwijzen naar contact.html. (Deze pagina bestaat niet, als je het leuk vindt, mag je hem erbij maken.)
27. Zorg er met marges voor dat de tabel in het midden van de division komt te staan en geef aan de boven- en onderkant 20 pixels marge op. Maak de tabel 900 pixels breed. Geef de tabel een padding van 20 pixels. Geef ook aan dat alle tekst in de tabel het lettertype 'Impact' moet hebben van 14 punten groot.
28. Geeft de hyperlink ook de class focus, en voeg aan de CSS-code toe dat alle elementen met de class focus geen onderlijning hebben. Stel ook in dat als je met de muis over de hyperlink in #contact heengaat de tekstkleur van de hyperlink zwart wordt.
29. In de tweede cel van de tweede rij komen vier klikbare plaatjes. Zet de plaatjes in de juiste map bij je andere bestanden en voeg ze in de HTML-code toe. Bij het href-attribuut mag je zelf iets bedenken. De laatste cel van de tabel blijft leeg.
30. Geef de vier plaatjes een hoogte en breedte van 50 pixels en een marge van 10 pixels aan alle zijden. Waarom is het handig dit in css te doen in plaats van html? Zorg ervoor dat de plaatjes naast elkaar staan in plaats van onder elkaar.
31. Nu de footer. Geef het p-element in de footer een breedte van 860pixels en zorg dat het element (en niet de tekst) in het midden komt te staan.

32. De website is nu compleet. Om het nog mooier te maken schakel je de grijze achtergronden van article en #contact uit.

Vraag: Waarom hebben we tot dit punt de achtergronden grijs gelaten?

Detaillering (niet verplicht)

33. Zorg ervoor dat de navigatiebalk niet uit het beeld kan worden gescrold:
Geef #headwrapper de opmaak `position:fixed, top:0px, z-index:1`.

Test de website. Wat gaat er mis?

34. Geef #headwrapper een witte achtergrond en een hoogte van 125 pixels.
Geef #bigimage een marge van 125 pixels aan de bovenkant.
35. Voor meer subtiliteit geven we het hr element een border van 1 pixel, kleur #eeeeee en maak hem solid. We passen bij #headwrapper de opmaak `'box-shadow: 0px 1px 10px grey;'` toe.
36. Geeft de border aan de bovenkant van #contact dezelfde eigenschappen als het hr element in #headerwrapper.
37. Geef de plaatjes in #contact als eigenschap `opacity :0.8` en `fliter:alpha(opacity=80)`. Stel ook in dat als je met de muis over zo'n plaatje heen gaat, de opacity 1.0 en de alpha 100 worden.

SUPERCOMPUTERS

[Home](#)

[Meer informatie](#)

[Tijdsje](#)

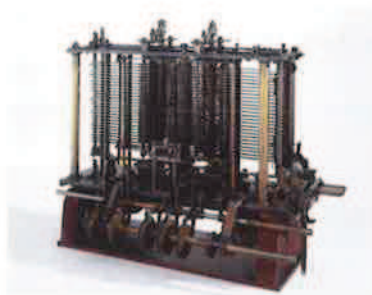


>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vehicula magna ac porta condimentum. Nulla tristique pulvinar leo. Donec ipsam lectos, molestie arcuansan hendrerit nec, vulputate et urna. Quisque euismod bibendum euismod.

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 portitor 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.



Babbage Difference Engine

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.

De tijd staat niet stil

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.

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.

Volg Supercomputers:



[Ga naar de contactpagina](#)

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

6.7.2 Opdracht 2

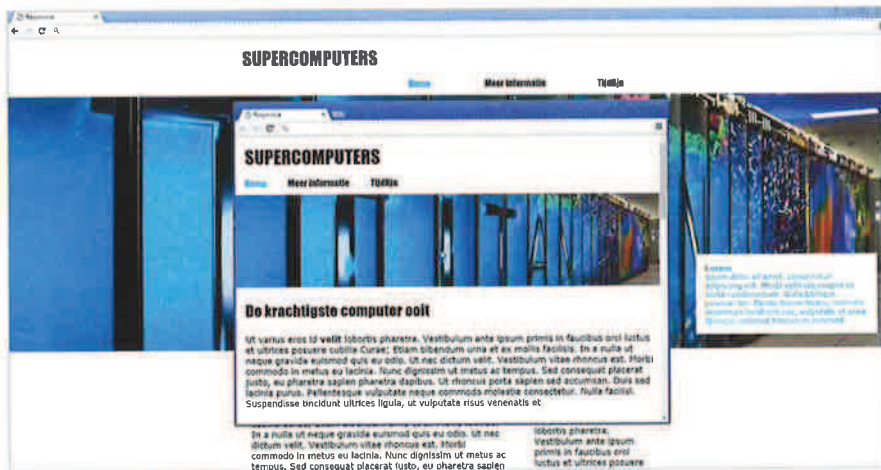
In deze opdracht ga je de website die je maakte in opdracht 1 responsive maken. Als je het resultaat van opdracht 1 nu met een smartphone of tablet zou bekijken, neemt de grote afbeelding van de Titan II bijna alle ruimte in beslag, moet je veel scrollen en zijn er nog enkele zaken die niet fijn werken.

We gaan in deze opdracht veel gebruik maken van de volgende CSS3 media query:

```
@media screen and (max-width: 1080px) {...}
```

Code die we tussen de accolades plaatsen is van toepassing op schermen die maximaal 1080 pixels breed zijn. In de praktijk vallen nagenoeg alle smartphones en tablet hieronder.

De nieuwe CSS-code kun je nog steeds op je computer testen. Als je de webbrowser kleiner maakt zal deze nieuwe code automatisch 'geactiveerd' worden.



Om te beginnen

- Open index.html en style.css.

De opdracht

1. We beginnen met het passend maken van de grote afbeelding van de Titan II voor een kleiner scherm. Voeg de bovengenoemde media query toe en plaats daarin code die de division #bigimage een hoogte geeft van 200 pixels. De nieuwe CSS-code is dan als volgt:

```
@media screen and (max-width: 1080px) {  
  #bigimage {  
    height: 200px;  
  }  
}
```



Let op dat de 'oude' code van #bigimage blijft staan. Die blijft namelijk van toepassing op schermen die breder zijn dan 1080 pixels.

2. Het bijschrift bij de grote afbeelding staat nu niet mooi meer: het neemt teveel ruimte in beslag. Daarom laten we in de responsive versie van de website dit bijschrift weg. Gebruik hiervoor de CSS-eigenschap display met de waarde 'none'.
3. Nu gaan we de header met de tekst 'SUPERCOMPUTERS' aanpassen. De header heeft nu een breedte van 900 pixels en een automatische marge. Voeg nieuwe code toe aan de media query die de header een breedte van 100 procent geeft en alleen aan de linkerkant een padding van 15 pixels.

Het instellen van een breedte van 100% is belangrijk. De header wordt dan namelijk in alle gevallen even breed is als het scherm. Als je een breedte op zou geven van 900 pixels, dan ontstaat er een probleem: er komt een horizontale

scrolbalk op schermen die kleiner zijn dan 900 pixels.

De padding aan de linkerkant zorgt er voor dat de tekst 'SUPERCOMPUTERS' niet strak tegen de linkerkant van het scherm komt te staan. Dit zelfde resultaat kan ook verkregen worden door de linkerpadding van de h1 in de header 15 pixels groter te maken.

4. De knoppen in de menubalk staan rechts uitgelijnd en staan best ver uit elkaar. Dat is handig als je een groot scherm hebt, maar bij een kleiner scherm is het beter als ze links zijn uitgelijnd en minder tussenruimte hebben.

Geef het nav-element daarom, net als de header, een breedte van 100 procent en lijn de tekst moet links uit. De automatische marge is in de responsive versie niet meer nodig - verwijder die. Geef de a-elementen in het nav-element een automatische breedte, een padding van 20 pixels aan de linker- en rechterkant en lijn ze links uit. Het resultaat hiervan is dat de knoppen in de menubalk aan de linkerkant staan, met minder tussenruimte.



5. De bovenkant van de website is nu goed toegankelijk voor bezoekers met een smartphone of tablet. Het gedeelte onder de coverafbeelding, de articles, gaan we nu aanpassen. Om te beginnen geven we het main-element dezelfde opmaak als het header- en het nav-element, namelijk een breedte van 100 procent. Geef het main-element ook een linkermarge van 20 pixels, zodat de inhoud van het main-element wat los komt van de linkerkant van het scherm.

6. Door de aanpassing van de linkermarge van het main-element ontstaat er nu een probleem. In het main-element staan (in de articles) paragrafen. Aan het begin van de opdracht hebben we alle paragrafen een padding van 20 pixels gegeven. Hierdoor is de ruimte aan de linkerkant van een paragraaf in het main-element nu 40 pixels (20 pixels linkermarge van het main-element + 20 pixels paragraaf-padding). Los dit op door de linker-padding van de paragrafen 0 pixels te maken.
7. In totaal zijn er drie articles in het main-element. Deze krijgen hun positie door middel van de classes `bigleft` en `smallright`. In de responsive versie van de website is het mooier om alle articles onder elkaar weer te geven. Als eerste gaan we de article-elementen zelf aanpassen; geef ze een breedte van 100 procent, een boven- en ondermarge van 20 pixels en een linker- en rechtermarge van 0 pixels.
8. Lijn de tekst in het `h1`-element van een article aan de linkerkant uit en geef dit element een linkerpadding van 0 pixels.
9. Geef voor zowel de classes `.bigleft`, `.smallright` en `.clear` de volgende code op:
 - `width: 100%;`
 - `display: block;`
 - `clear: both;`
 - `height: auto;`
 - `margin-left: 0px;`Hiermee zorg je er voor dat alle articles de volledige breedte van het scherm innemen en dat ze als 'blokken' worden weergegeven, onder elkaar.
10. De tekst die onder de afbeelding van de Difference Engine staat is gecentreerd. Aangezien veel elementen van onze website nu vanaf links zijn uitgelijnd, is het goed om ook deze tekst links uit te lijnen. Pas hiervoor de class `.center` aan met een uitlijning aan de linkerkant.
11. De responsive website is nu bijna klaar. Er moeten nog twee aanpassingen gedaan worden. Zowel de tabel in de division `contact` als de paragraaf in het footer-element hebben nog een breedte van ongeveer 900 pixels. Dit moet in de responsive versie, net als veel andere elementen, 100 procent worden.
12. De tekst in de paragraaf van de footer staat strak tegen de linkerkant van het scherm. Geef deze paragraaf, net als de paragrafen in de articles, een linkerpadding van 20 pixels.

SUPERCOMPUTERS

[Home](#) [Meer informatie](#) [Tijdslijn](#)



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 donec molestie sed eros sed laoreet. Nunc euismod nulla tortor, sed pulvinar justo cursus sit amet. Praesent id varius nisi.

De tijd staat niet stil

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.

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.



Babbage Difference Engine

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.

Volg Supercomputers:

[Ga naar de contactpagina](#)



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

