

Web Technologie

Hallo, HTML!

```
1
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <h1>Hello World!</h1>
12
13   </body>
14 </html>
15
```

<http://bit.ly/1OTpo2e>

Tags versus Content

Elk HTML document telt 2 types content:

- Tekst (content)
- Tags (bijvoorbeeld: <html>, <head>, ...)

Tags zijn een vorm van metadata. Data over data!
In dit geval worden ze gebruikt om de inhoud van een pagina te structureren

Je browser toont enkel de content, de tags worden gebruikt om betekenis te kunnen afleiden

```
1
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <h1>Hello World!</h1>
12
13   </body>
14 </html>
15
```

P van Paragraaf

Typische workflow voor web development:

Aanpassen in je tekst editor en heen en weer checken in je browser (bijvoorbeeld Chrome) door te refreshen

```
1
2 <!DOCTYPE html>
3
4 <html>
5   <head>
6     <title>Mijn eerste HTML pagina</title>
7
8   </head>
9
10  <body>
11    <h1>Hello World!</h1>
12    <p>
13      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
14      Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
15      Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
16      Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
17    </p>
18
19  </body>
20 </html>
21
```

<http://lipsum.com/>

Commentaar

Handige manier om HTML te verklaren

Steeds nuttig om de meer complexe stukken code te voorzien van commentaar

Goed voor anderen die de code moeten interpreteren, of voor jezelf later zodat je (of andere developers) begrijpt waarom een bepaalde keuze werd gemaakt

```
1
2 <!DOCTYPE html>
3
4 <html>
5   <head>
6     <title>Mijn eerste HTML pagina</title>
7
8   </head>
9
10  <body>
11    <h1>Hello World!</h1>
12    <!-- TODO: Onderstaande tekst moet nog aangepast worden -->
13    <p>
14      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
15      Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
16      Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
17      Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
18    </p>
19
20  </body>
21 </html>
```

Hoofdingen H1..6

Hoofdingen zijn gereserveerd voor belangrijke content op de site

We kunnen op verschillende niveaus worden aangebracht:

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

De belangrijkste hoofding krijgt <h1>

```
1
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11
12     <h1>De belangrijkste hoofding</h1>
13     <p> Een belangrijke paragraaf</p>
14     <h2>een minder belangrijke hoofding</h2>
15     <p>Dit is minder belangrijke content</p>
16
17   </body>
18 </html>
19
```

Links: A-Tags

<a>-tag staat voor anchor en wordt gebruikt om links te maken

Unieke eigenschap van hypertext want ze kunnen linken naar andere informatie (op dezelfde pagina, of op een andere pagina)

Anchor tags hebben een href attribuut in de openings-tag die aangeeft waar de browser naartoe moet gaan wanneer de link wordt aangeklikt.

Tekst in de tag wordt onderlijnd en is aanklikbaar

```
1
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <p>Hier: een <a href="http://www.google.be">link</a>naar Google</p>
12     <p><a href="#">En nog een link..voor development</a></p>
13     <p><a href="http://www.facebook.com">Een link naar facebook.com</a></p>
14
15   </body>
16 </html>
```

Lijsten ul

De -tag staat voor niet-geordende lijsten of unordered lists

 staat voor list items

Gebruik het als de volgorde niet uitmaakt

(Wordt dikwijls gebruikt om een navigatie systeem te bouwen)

```
1
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <h1>Lijst voorbeeld</h1>
12
13     <ul>
14       <li> een list item met een link <a href="http://www.yahoo.com">yahoo</a></li>
15       <li> een list item</li>
16       <li> nog een list item</li>
17     </ul>
18
19
20   </body>
21 </html>
22
```


Lijsten ol

Niet geordende lijsten krijgen bullets in de browser

Geordende lijsten krijgen een nummering

```
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <h1>Lijst voorbeeld</h1>
12
13     <ol>
14       <li>een list item met een link <a href="http://www.yahoo.com">yahoo</a></li>
15       <li>een list item</li>
16       <li>nog een list item</li>
17     </ol>
18
19
20   </body>
21 </html>
22
```



Lijst voorbeeld

1. een list item met een link [yahoo](http://www.yahoo.com)
2. een list item
3. nog een list item

Document Object Model (DOM) & boomstructuren

HTML tags definiëren een hiërarchische structuur die het Document Object Model wordt genoemd of kortweg de DOM

Met de DOM kan geïnterageerd worden met een scripting taal als JavaScript.

HTML elementen definiëren DOM-elementen. Dit zijn entiteiten die leven binnen de DOM

Een voorbeeld

We werken met insprongen, maar voor sommige tags wordt dit niet gedaan, bijvoorbeeld de `` tags

De code vertaalt zich in een boomdiagram van de DOM.

We spreken over:

- Afstammelingen (descendants)
- Kind-Ouder-relatie (parents-children)

```
2  <!DOCTYPE html>
3
4  <html>
5    <head>
6      <title>Mijn eerste HTML pagina</title>
7
8    </head>
9
10   <body>
11     <h1>Hallo</h1>
12     <div>
13       <ol>
14         <li>list item</li>
15         <li>list item</li>
16         <li>list item</li>
17       </ol>
18       <p>Dit is een paragraaf</p>
19       <p> en dit een <span>tweede</span></p>
20     </div>
21
22     <ul>
23       <li>list item <span>1</span></li>
24       <li>list item <span>2</span></li>
25       <li>list item <span>3</span></li>
26     </ul>
27
28
29   </body>
30 </html>
31
```

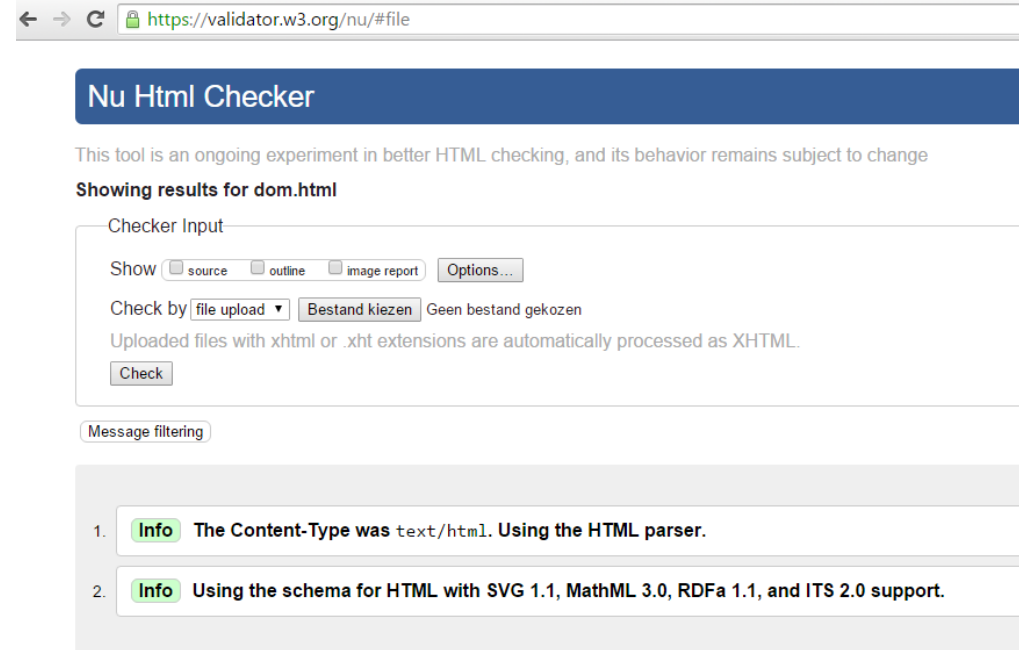

HTML validatie

Elk element ligt besloten tussen 2 tags

Kan bij lange documenten soms fout lopen

Daarom is een HTML validator een handige tool

W3C heeft een Markup Validation Service



The screenshot shows the Nu Html Checker interface in a web browser. The address bar displays <https://validator.w3.org/nu/#file>. The page has a dark blue header with the text "Nu Html Checker". Below the header, a message states: "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". The main section is titled "Showing results for dom.html". Under "Checker Input", there are options to "Show" (source, outline, image report) and an "Options..." button. The "Check by" section shows "file upload" selected, with a "Bestand kiezen" button and the text "Geen bestand gekozen". A note mentions that uploaded files with .xhtml or .xht extensions are processed as XHTML. A "Check" button is present. Below the input section is a "Message filtering" button. The results area shows two messages: 1. "Info The Content-Type was text/html. Using the HTML parser." and 2. "Info Using the schema for HTML with SVG 1.1, MathML 3.0, RDFa 1.1, and ITS 2.0 support."

Oefening

We gaan een nieuwe pagina bouwen voor Jeroom, de cartoonist (jeroom.be)

Onderweg leren we een paar belangrijke nieuwe HTML tags en waarvoor ze staan

We gaan oefenen met de frontpagina van de site tot we HTML in onze vingers krijgen

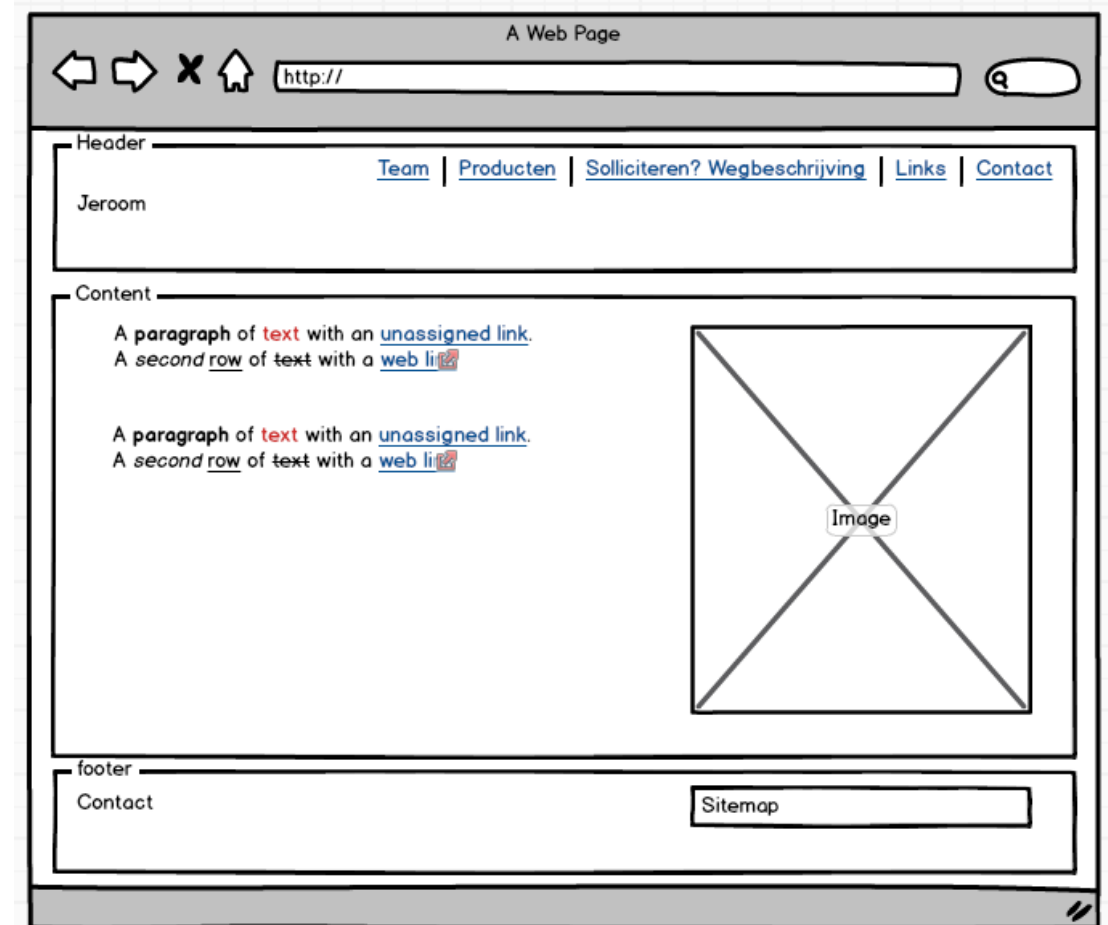
Stijlen doen we later met CSS!



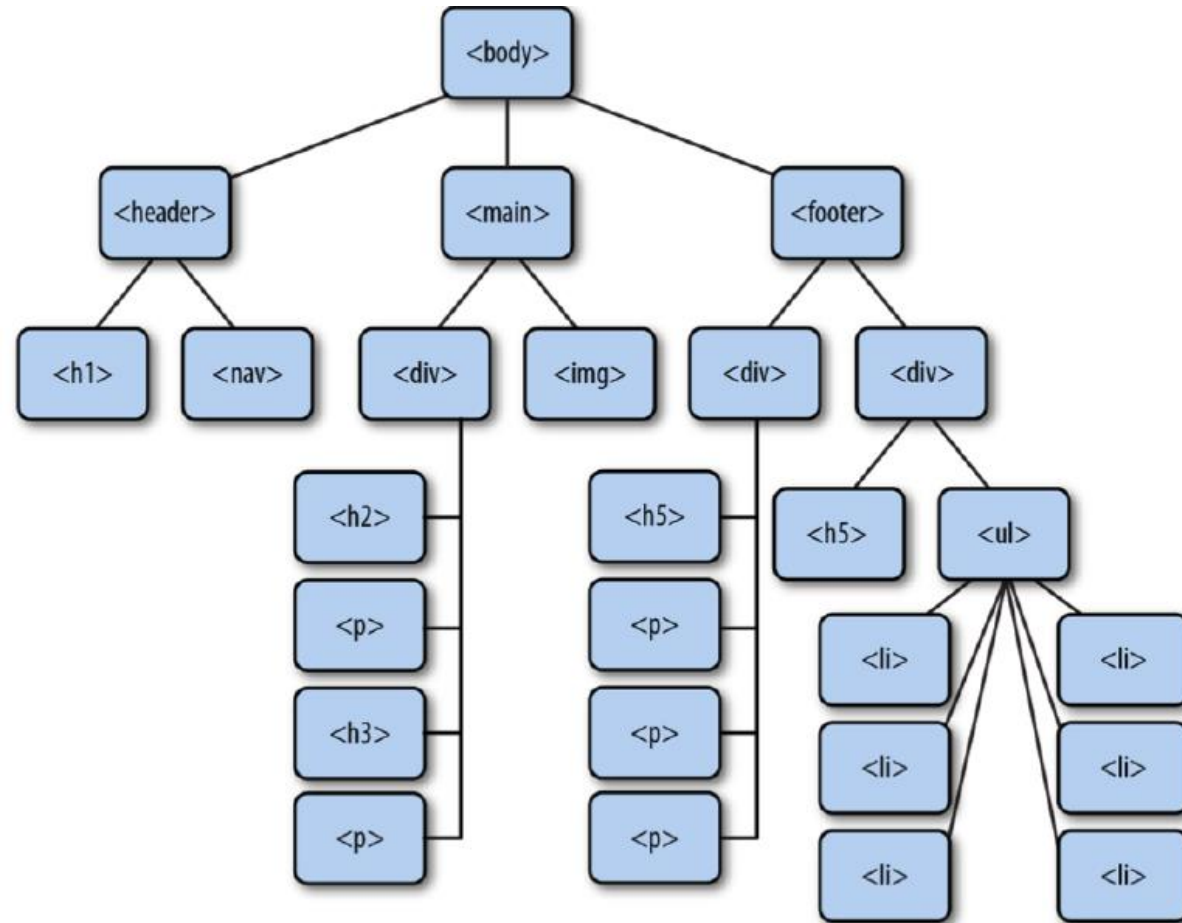
Oefening

We maken een schets (wireframing / balsamiq) over hoe het eindresultaat er moet uitzien.

Het helpt ons om de verschillende delen van de structuur te identificeren



Boomschema voor de structuur



Oefening

Eens de boomstructuur getekend is en de tags kennen wordt het relatief eenvoudig om de HTML te coderen

We beginnen met een skeletversie van de pagina

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>jerom</title>
5
6    </head>
7
8    <body>
9      <h1>Jerom</h1>
10
11    </body>
12  </html>
```

Resultaat in de browser

← → ↻ 📄 127.0.0.1:8020/Web%20technology%201/Les2/jeroom.html

Jeroom

Oefening

Nu we de structuur vastgelegd hebben kijken we naar het boomschema.

We zien een header, content (main) en footer

Ze zijn alle 3 “kinderen” van het body element

HTML heeft een tag beschikbaar voor alle drie: header en footer voor de elementen bovenaan en onderaan een document. Main voor de belangrijkste content van een document

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>jeroom</title>
5
6    </head>
7
8    <body>
9      <header>
10        <h1>Jeroom</h1>
11
12      </header>
13
14      <main>
15        ~~~~~ dit is de main ~~~~~
16      </main>
17
18      <footer>
19        dit is de footer
20      </footer>
21
22    </body>
23  </html>
```

De Header

We voegen een navigatiesectie toe aan de pagina. Met behulp van CSS zullen we ze later in de rechterbovenhoek plaatsen en stijl geven.

HTML heeft een tag die een navigatie element ondersteunt, 'nav'

We voegen het element toe aan onze header tag

De links binnen het nav element worden in een a-tag gestopt

Omdat we in development mode zitten, en nog geen links hebben gebruiken we een tijdelijke placeholder #

```
8- <body>
9-   <header>
10-     <h1>Jerome</h1>
11-     <nav>
12-       <a href="#">Team</a>
13-       <a href="#">Producten</a>
14-       <a href="#">Solliciteren</a>
15-       <a href="#">Wegbeschrijving</a>
16-       <a href="#">Links</a>
17-       <a href="#">Contact</a>
18-     </nav>
19-   </header>
20-
```

De MAIN content

De main content is onderverdeeld in 2 gedeeltes:

- Content links
- Afbeelding rechts

Een img-tag hoeft niet gesloten te worden. (HTML5 “void” elementen hebben geen content en hoeven niet gesloten te worden)

Het alt attribuut is verplicht en bevat een tekstuele beschrijving van de afbeelding

De foto plaatsen we in dezelfde folder als ons html bestand

```
20 <main>
21 <div>
22 <h2>Leverancier van cartoons sinds 1998</h2>
23 <p>
24 Welkom bij Jeroom, gespecialiseerd in het
25 maken van cartoons voor gans Europa. Sinds 1998 levert dit bedrijf
26 diensten aan allerlei dagbladen, reclamebureau's en particulieren.
27 We zijn klein begonnen maar door de jaren heen is Jeroom uitgegroeid
28 tot een vaste waarde in de cartoonwereld. Kijk gerust rond op onze site,
29 maak kennis met onze producten en het toffe team
30 dat dit bedrijf
31 dagelijks runt.</p>
32 <h3>Hiervan kan je Jeroom kennen</h3>
33 <ul>
34 <li>Rudy het boek (2002)</li>
35 <li>In het hol van de reet (2003)</li>
36 <li>Beerschot (2006)</li>
37 </ul>
38 </div>
39 
40
41 </main>
```

Het resultaat

← → ↻ 127.0.0.1:8020/Web%20technology%201/Les2/jeroom.html

Jeroom

[Team](#) [Producten](#) [Solliciteren](#) [Wegbeschrijving](#) [Links](#) [Contact](#)

Leverancier van cartoons sinds 1998

Welkom bij Jeroom, gespecialiseerd in het maken van cartoons voor gans Europa. Sinds 1998 levert dit bedrijf diensten aan allerlei dagbladen, reclamebureau's en particulieren. We zijn klein beg gerust rond op onze site, maak kennis met onze producten en het toffe team dat dit bedrijf dagelijks runt.

Hiervan kan je Jeroom kennen

- Rudy het boek (2002)
- In het hol van de reet (2003)
- Beerschot (2006)

Jeroom Snelders, directeur



Niet zo, maar zo!

dit is de footer

De footer

De footer bevat 2 secties:

- Contactinformatie
- Links die we sitemap noemen

HTML heeft geen element dat rechtstreeks verwijst naar contact informatie

Maar geeft ons wel 2 generieke tags die hiervoor kunnen gebruikt worden:

- <div>
-

We opteren voor div elementen met elk een eigen **class attribuut**

We gebruiken ook een ul-tag voor onze sitemap

```
43<
44<
45
46<
47      Jeroom Snelders
48    </p>
49    <p>
50      Algemeen directeur
51    </p>
52    <p>
53      jeroomtet@hotmail.com
54    </p>
55  </div>
56  <div class="sitemap">
57    <h5>Sitemap</h5>
58    <ul>
59      <li><a href="#">Home</a></li>
60      <li><a href="#">Team</a></li>
61      <li><a href="#">Producten</a></li>
62      <li><a href="#">Solliciteren</a></li>
63      <li><a href="#">Wegbeschrijving</a></li>
64      <li><a href="#">Links</a></li>
65      <li><a href="#">Contact</a></li>
66    </ul>
67  </div>
68</footer>
```

Besproken HTML tags

Tag	Beschrijving
<html>	Hoofdcontainer voor een html-document
<head>	Meta informatie over het document
<body>	Bevat informatie die in de browser getoond wordt (de content)
<h1>	Belangrijkste hoofding
<h2>	Tweede belangrijkste hoofding
<h3>	Derde belangrijkste hoofding
<a>	Anchor, link naar een ander document of een aanklikbaar element
	Lijst van niet-geordende elementen
	Lijst van geordende elementen
	Element uit een lijst

Besproken HTML tags

Tag	Beschrijving
<main>	Het voornaamste content-gebied
<footer>	De footer content van je pagina
<div>	Een container voor substructuur
<nav>	Het navigatie element van je pagina
<header>	

Oefening 2

Maak een boodschappenlijstje
(boodschappen.html)

Maak zeker ook een paar links in het lijstje
naar zaken die je online wil gaan kopen

Shopping List

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____

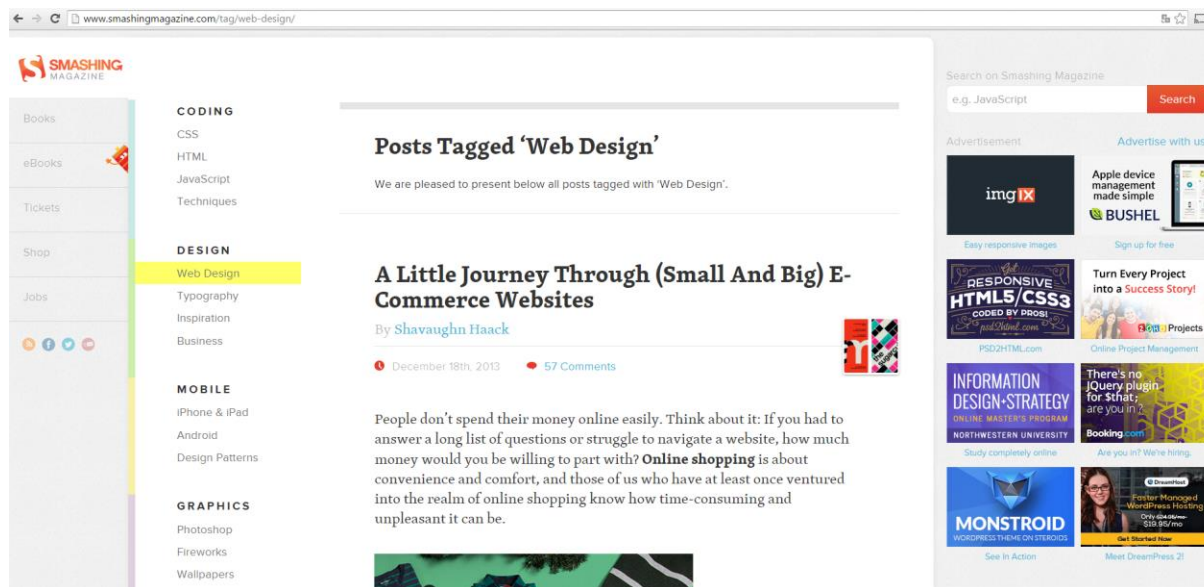
©Vitronic Printed in U.S.A.

Inleiding CSS

We hebben geleerd de inhoud van een HTML document te structureren

In termen van design en stijl laten deze pagina's nog veel te wensen over

Een CSS file beschrijft in essentie hoe specifieke elementen van HTML moeten getoond worden in de browser



Wat is CSS

CSS = Cascading Style Sheets

- Stylen en layouten van HTML elementen en structuur van pagina
- CSS controleert de styling of de presentatie van een pagina

CSS is een presentatietaal

De CSS is in ons voorbeeld toegevoegd aan de HTML pagina in de head. In de praktijk gebeurt dit meestal in een aparte file.

```
6  <head>
7    <title>css1</title>
8
9    <style>
10     body {
11         color: red;
12     }
13
14     h1 {
15         font-family: Arial, sans-serif;
16         font-size: 32px;
17         color: black;
18     }
19    </style>
20 </head>
21
22 <body>
23     <h1>Start</h1>
24 </body>
25 </html>
```

CSS

Dit voorbeeld heeft een nieuwe tag in de <head> tag.

De <link> tag linkt naar een externe stylesheet die beschrijft hoe het document dient getoond te worden.

```
6  <head>
7    <title>css2</title>
8    <link href="css2.css" rel="stylesheet" type="text/css">
9  </head>
10
11 <body>
12   <h1>Hello, World</h1>
13   <p>Dit is een paragraaf</p>
14
15 </body>
16 </html>
```

CSS bestand

Dit is een eenvoudig voorbeeld van een CSS bestand

Het zet de achtergrond van een body element naar een lichtblauw cyaan

De tekst binnen de h1 tag is kastanjebruin (maroon)

We hebben ook de body tekst een breedte gegeven van 800 pixels en de margin op auto gezet zodat de body gecentreerd staat op de pagina.

We geven de tekst binnen het p element een grijze kleur en een dunne rand errond

```
1 body{
2     background-color: lightcyan;
3     width: 800px;
4     margin: auto;
5 }
6
7 h1{
8     color:maroon;
9     text-align:center;
10 }
11
12 p{
13     color:gray;
14     border:1px solid gray;
15     padding:10px;
16 }
17 }
```

Regelsets

Een CSS file is een verzameling van regelsets.
Een regelset is een verzameling van stijlregels die toegepast worden op een aantal elementen in de DOM.

Een regelset bestaat uit een selector met daarna – tussen accolades – een lijst van regels.

Elke regel bestaat uit een specifieke eigenschap, gevolgd door een dubbelpunt, gevolgd door een waarde (of een lijst van waardes gescheiden door spaties), gevolgd door een puntkomma

Selector

```
{  
    property:value;  
}
```

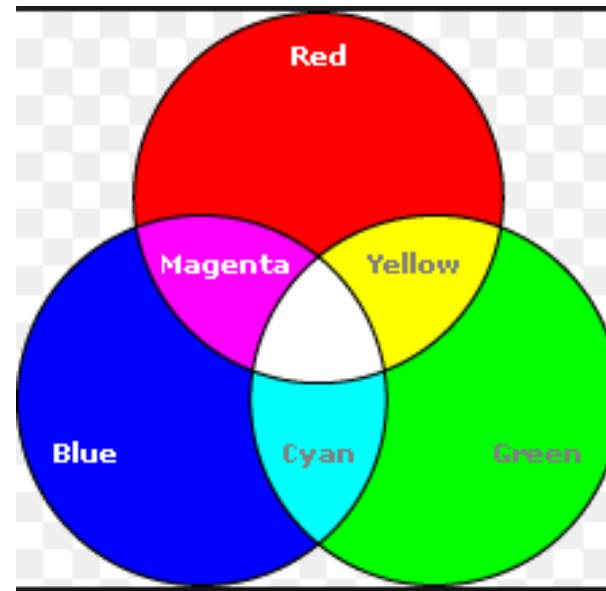
```
1 body{  
2     background-color: lightcyan;  
3     width: 800px;  
4     margin: auto;  
5 }
```

Kleuren

CSS heeft 2 manieren om kleuren te specificëren

- De CSS kleurnamen
- De hexadecimale code. 6 hexadecimale getallen.
1^{ste} paar is de hoeveelheid rood, 2^{de} paar de hoeveelheid groen, 3^{de} paar de hoeveelheid blauw

```
body{  
  background: lightcyan;  
  color:#FF0000;  
}
```



Commentaar

Net zoals bij HTML kan onze CSS code ook comments bevatten.

Het zijn code annotaties die door de browser volledig genegeerd worden.

Wanneer comments? Als het echt nodig is (de beste commentaar is goed geschreven code!)

- Bijvoorbeeld wanneer je een keuze maakt die minder evident is

```
2  /* Hier gaan we de body stylen */
3  body{
4      width:800px; /* De breedte van body op 800 pixels */
5      background: lightcyan; /* achtergrond lichtblauw */
6      color:#FF0000; /* voorgrond in het rood */
7  }
```

Inline elementen

Bijna steeds worden HTML elementen getoond op 1 van 2 manieren. De eerste is INLINE.

Voorbeelden hiervan zijn `<a>` of ``.

Dit betekent onder andere dat de content op dezelfde lijn als de omgevende content zal staan

```
<div>  
  Dit is een paragraaf en dit <span>woord</span> verschijnt inline.  
  Deze <a href="http://www.voorbeeld.com">link</a> zal ook inline getoond worden.  
</div>
```

Block elementen

De meeste HTML elementen zijn BLOCK elementen.

Dit betekent dat de content binnen dit element op een nieuwe lijn zal staan buiten de normale flow van de tekst

Voorbeelden van block elementen: p, nav, main, div, ..

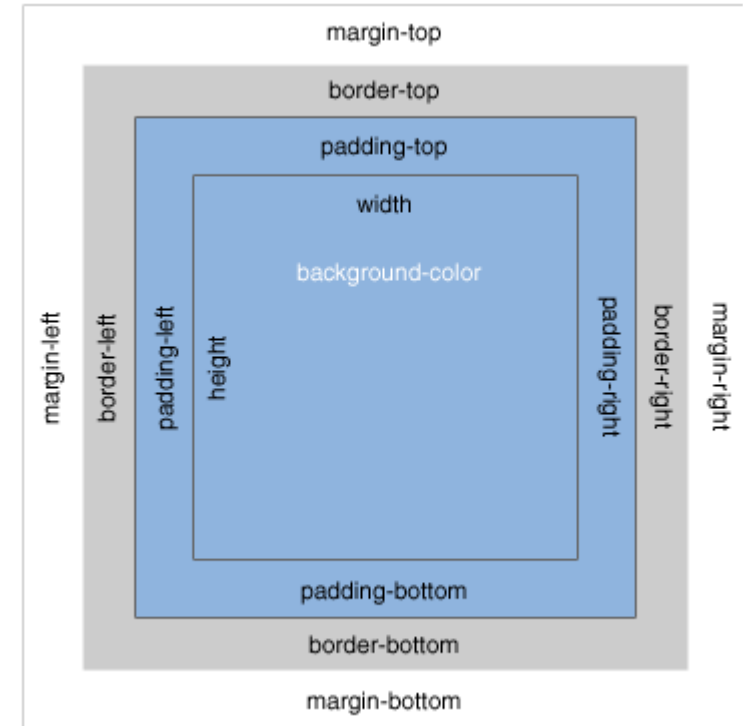
```
<div>  
  Dit is een paragraaf en dit <div>woord</div> verschijnt op een andere regel.  
  Deze <a href="http://www.voorbeeld.com">link</a> zal inline getoond worden.  
</div>
```

Padding, border, margin

Zowel block als inline elementen hebben background en color eigenschappen.

Maar block elementen hebben 3 andere eigenschappen die zeer handig zijn wanneer we de pagina stylen met custom layouts:

- Padding: de ruimte tussen de content en de border
- Margin: de ruimte tussen het element en het omgevende element
- Border: de ruimte tussen de padding en de margin



Padding, border, margin voorbeeld

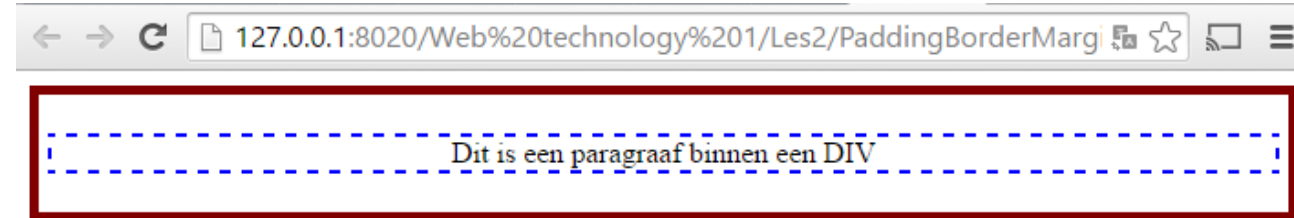
```
<style>
  div{
    border:5px solid maroon;
    text-align:center;
    padding:5px;
  }

  p{
    border:2px dashed blue;
  }

</style>
</head>

<body>

  <div>
    <p>Dit is een paragraaf binnen een DIV</p>
  </div>
```



Oefening

Experimenteer: verander in CSS padding, border en margin eigenschappen van elk element en bekijk

Selectors

Het belangrijkste aspect van CSS komt van het effectief gebruiken van selectors

We hebben een basis type selector gezien die de tagnaam gebruikt om alle elementen die deze naam hebben te selecteren

De eerste regelset heeft betrekking op het h1 element, het tweede op beide p elementen

Soms is dit wat we willen, een andere keer willen we die beide elementen elk afzonderlijk aanpakken

```
20- <style>
21-   h1{
22-       background: black;
23-       color:white;
24-   }
25-
26-   p{
27-       margin:10px;
28-       padding:20px;|
29-   }
30- </style>
31- </head>
32-
33- <body>
34-
35-     <h1> Hello World! </h1>
36-     <p> Dit is een paragraaf</p>
37-     <p> Dit is een tweede paragraaf</p>
38-
39- </body>
```

Classes

We hadden al gezien dat we een klasse kunnen toevoegen aan een `<div>` tag. Dit kunnen we met elk element doen!

Op deze manier kunnen we bijvoorbeeld een `p` element met een eigen naam benoemen

Als `.eerste` niet ook voor andere DOM elementen wordt gebruikt kan je `p` zelfs weglaten

```
20 <style>
21   h1{
22     background: black;
23     color:white;
24   }
25
26   p.eerste{
27     margin:10px;
28     padding:20px;
29   }
30
31   .eerste{
32     margin:10px;
33     padding:20px;
34   }
35 </style>
36 </head>
37
38 <body>
39
40   <h1> Hello World! </h1>
41   <p class="eerste"> Dit is een paragraaf</p>
42   <p class="tweede"> Dit is een tweede paragraaf</p>
43
44 </body>
```


Pseudoclasses

In de vorige les zagen we dat we aanklikbare DOM elementen kunnen maken door gebruik van de <a> tag

Het a-element kan net als elk ander DOM element gestyled worden via CSS

Bijvoorbeeld de kleur van de link kan gewijzigd worden

Het is vaak nuttig een verschillende kleur te gebruiken wanneer de gebruiker over de link gaat

```
20 <style>
21
22     a{
23         color:cornflowerblue;
24     }
25
26     a:hover{
27         color:tomato;
28     }
29 </style>
30 </head>
31
32 <body>
33     <ul>
34         <li>
35             <a href="#">Link 1</a>
36         </li>
37         <li>
38             <a href="#">Link 2</a>
39         </li>
40         <li>
41             <a href="#">Link 3</a>
42         </li>
43     </ul>
44 </body>
--
```

Nog selectors

Wanneer het boomdiagram voor onze DOM complexer wordt, wordt het ook noodzakelijk complexere selectors te gebruiken

Ons voorbeeld bevat 2 lijsten, verschillende paragrafen en verschillende lijstelementen

```
11 <body>
12   <h1>Hello World</h1>
13   <div class="content">
14
15     <ol>
16       <li>
17         List item<span class="number">één</span>
18       </li>
19       <li>
20         List item<span class="number">twee</span>
21       </li>
22       <li>
23         List item<span class="number">drie</span>
24       </li>
25     </ol>
26     <p>
27       Dit is de <span>eerste</span> paragraaf
28     </p>
29     <p>
30       Dit is de <span>tweede</span> paragraaf
31     </p>
32   </div>
33   <ul>
34     <li>
35       List item<span class="number">één</span>
36     </li>
37     <li>
38       List item<span class="number">twee</span>
39     </li>
40     <li>
41       List item<span class="number">drie</span>
42     </li>
43   </ul>
44
45 </body>
46 </html>
```

Nog selectors

We kunnen makkelijk algemene elementen selecteren en van stijl voorzien zoals we al eerder gezien hebben

Bijvoorbeeld indien we een afgeronde border willen aanbrengen rond de geordende lijst ``, dan kunnen we de regelset hiernaast aanbrengen

Indien we de lijstelementen in deze lijst geel willen maken, dan zou onze eerste gedachte die hiernaast zijn

Maar dat zal alle lijstelementen aanpakken: in de `` en ook in de ``

```
<style>

  ol{
    border:5px solid blue;
    border-radius:10px;
  }

  li{
    color:yellow;|
  }

</style>
```

Nog selectors TOT HIER

We kunnen specifieker gaan stylen door enkel de 's aan te pakken in de ol

Als er meerdere geordende lijsten op de pagina zijn willen we mogelijk nog specifieker worden en selecteren we enkel de lijstelementen die nakomelingen zijn van de content <div> elementen

```
9
10
11
12
13
14
15
16
17
18
19

<style>

    ol li{
        color:brown;
    }

    .content li{
        color:yellow;
    }

</style>
```

Nog selectors

Indien we enkel de achtergrond van de eerste elementen in beide lijsten geel willen maken?

Er bestaat een pseudo-klasse voor elementen die het “eerste” kind zijn van hun “ouders”

Op net dezelfde manier kunnen we elementen selecteren die het tweede, derde, vierde, ... kind zijn door gebruik te maken van de nth child pseudo klasse

```
9
10
11
12
13
14
15
16
17
18
19
20
```

```
<style>

    li:first-child{
        background:green;
    }

    li:nth-child(2){
        background:yellow;
    }

</style>
```

Cascade regels

Wat gebeurt er wanneer 2 verschillende rulesets selectors gebruiken voor hetzelfde element

Er zijn cascade regels die gaan bepalen wat er met dat element gebeurt

De meer specifieke regel (hier de klasse) neemt de bovenhand

In het tweede voorbeeld neemt de regelset die het laatst wordt toegekend de bovenhand. De paragraaf zal groen zijn en indien we de volgorde veranderen in geel

```
<p class="greeting">Hello, CSS rules!</p>
```

```
9  <style>
10
11  p{
12      color:yellow;
13  }
14
15  p.greeting{
16      color:green;
17  }
18
19
20
21  p{
22      color:yellow;
23  }
24
25  p{
26      color:green
27  }
28
29  </style>
--
```

Overerving

Nakomelingen erven de eigenschappen van hun ouders

Alle nakomelingen erven de eigenschappen, tenzij een andere meer specifieke ruleset dit verhindert

Indien we de kleur veranderen van de body, dan zullen alle nakomelingen van de body die kleur erven

Dit is een essentieel onderdeel van CSS en daarom is het zo nuttig om een visualisatie van de DOM hiërarchie in je hoofd te hebben terwijl je elementen van styling voorziet

```
body{  
    background:yellow;  
}  
  
/*h1 is een nakomeling van body, dus gele achtergrond*/  
h1{  
    color:red;  
}  
  
/*h2 is ook nakomeling van body, door zelf achtergrond  
* te specificeren blokkeren we overerving*/  
h2{  
    background:green;  
}
```

Overerving

Hoewel de meeste CSS eigenschappen met overerving werken is dit niet altijd het geval

De belangrijkste niet-overerfbare eigenschappen zijn gerelateerd aan de block-style elementen

De margin, padding en border regels zijn niet overerfbaar van voorouders

```
11- body{
12     margin:0;
13     padding:0;
14 }
15- /*
16     * h1 zal niet de margin en de padding overnemen van de body,
17     * zelfs niet indien we geen alternatief speciëren
18     */
19- h1{
20
21 }
22
23
24 |
25 </style>
--
```


Layout met floats

We hebben eigenschappen gezien die de basis styling van DOM elementen bepaalt

Maar er zijn andere meer algemene eigenschappen die de algemene layout van de pagina gaan bepalen

Deze eigenschappen geven de ontwikkelaar controle over waar objecten zullen geplaatst worden

Eén daarvan is de float-eigenschap

Deze flexibele eigenschap laat ons toe complexe layouts te maken

```
26 <body>
27
28 <main>
29 <nav>
30 <p><a href="#">link1</a></p>
31 <p><a href="#">link2</a></p>
32 <p><a href="#">link3</a></p>
33 <p><a href="#">link4</a></p>
34 </nav>
35 <p class="content">
36 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
37 ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
38 laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
39 voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
40 non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
41 </p>
42 </main>
43
44 </body>
```

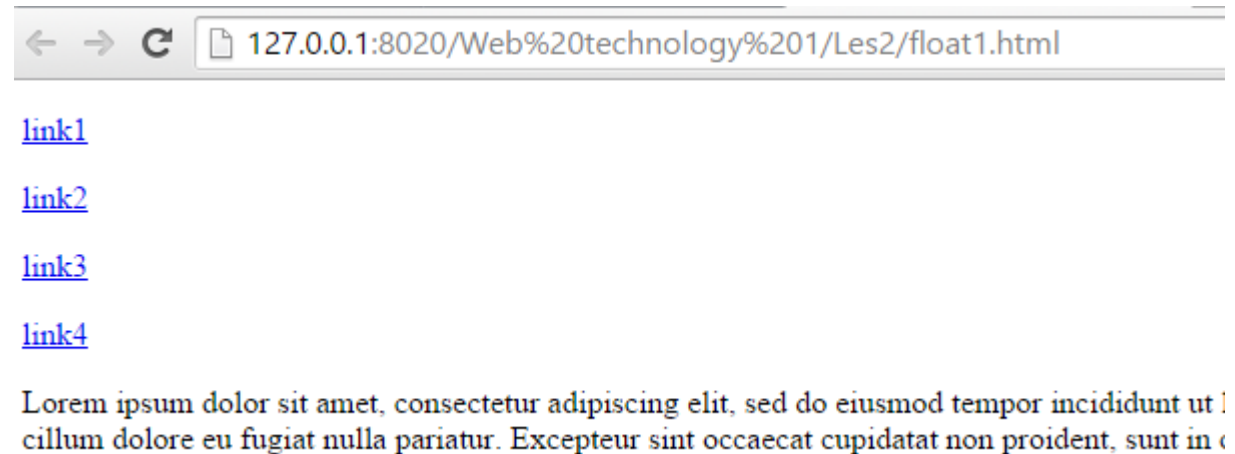
Layout met floats

De float eigenschap kan rechts of links gezet worden

Dit neemt het DOM element uit de normale flow en duwt het helemaal links of rechts binnen het omvattende element, voor zover er voldoende ruimte is!

In dit voorbeeld zitten een nav (met p elementen) en een p element besloten in een main element

Zonder styling worden al deze elementen gestapeld



Layout met floats

Als we volgende styling aanbrengen

- Nav: 200px breed en float naar rechts om een zijbalk layout te krijgen
- De inhoud van p vloeit dan rond de nav div

```
9  <style>
10
11  main{
12      width:500px;
13      margin:auto;
14      background:gray;
15  }
16
17  nav{
18      width:200px;
19      float:right;
20  }
21
22  p.content{
23      margin:0;
24  }
25
26  </style>
```

Layout met floats

Werkt goed voor afbeeldingen en andere inline elementen

Bij layouts willen we vaak een kolom effect

In dit geval willen we ook het p-element floaten naar links

```
9  <style>
10
11  main{
12      width:500px;
13      margin:auto;
14      background:gray;
15  }
16
17  nav{
18      width:100px;
19      float:right;
20  }
21
22  p.content{
23      margin:0;
24      float:left;
25      width:400px;
26  }
27
28  </style>
```

Layout met floats

Je merkt bij de vorige sample dat de grijze achtergrond verdwenen is!

Als alle elementen in een element floaten dan wordt de hoogte van een bevattend element 0

Voor een fix: overflow op auto

```
main{  
  width:500px;  
  margin:auto;  
  background:gray;  
  overflow:auto;  
}
```

Layout met floats

De som van alle pixels in het main element is groter dan de breedte

We kunnen dat oplossen door 2x de padding af te trekken van de breedte van het p-element

Zelfde truuk moet ook worden toegepast telkens we een border of margin toekennen aan elementen

```
9  <style>
10
11  main{
12      width:500px;
13      margin:auto;
14      background:gray;
15      overflow:auto;
16  }
17
18  nav{
19      width:100px;
20      float:right;
21  }
22
23  p.content{
24      margin:0;
25      float:left;
26      width:380px; /* 400- 2*10 = 380 */
27      padding:10px;
28  }
29
30  </style>
```

Clear property

Een interessant fenomeen kan de kop opsteken wanneer we floats gebruiken

Nu willen we een navigatie links met een footer onderaan beide kolommen

```
33<body>
34
35<main>
36<nav>
37<p><a href="#">link1</a></p>
38<p><a href="#">link2</a></p>
39<p><a href="#">link3</a></p>
40<p><a href="#">link4</a></p>
41</nav>
42<p class="content">
43Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
44ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
45laboreis nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
46voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
47non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
48</p>
49</main>
50<footer>
51<p>Dit is een footer</p>
52</footer>
53
54</body>
```

Clear property

Nav float naar links

Universele selector (alle DOM elementen) zet marge en padding op 0

Overal verschillende soorten grijs als achtergrond

```
9  <style>
10
11  *{
12      margin:0;
13      padding:0;
14  }
15
16  nav{
17      float:left;
18      background:darkgray;
19      width:200px;
20  }
21
22  main{
23
24      background:lightgray;
25      width:500px;
26      float:left;
27
28  }
29
30  footer{
31      background:gray;
32  }
33
--
```

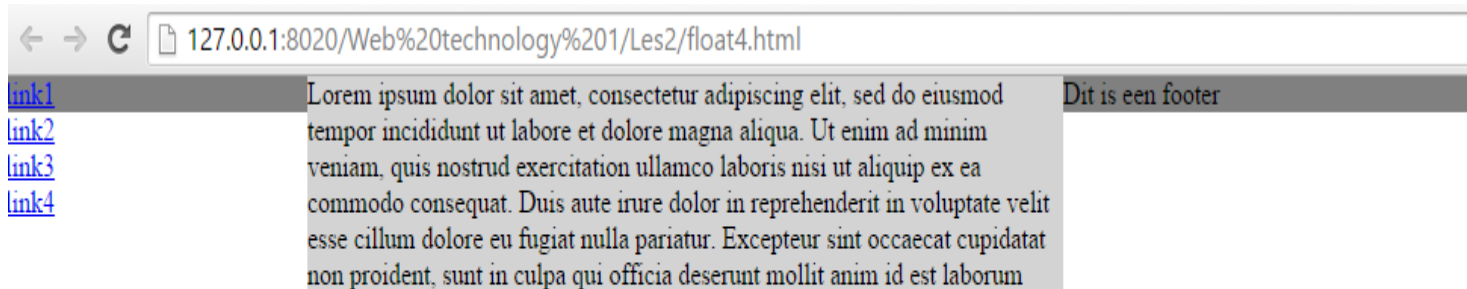

Clear property

Resultaat zie je rechts

Footer staat rechts, maar we willen deze onder beide elementen

Daar zorgt clear voor

Daarmee forceren we het footer element onder een floating element te komen



```
9 <style>
10
11 *{
12     margin:0;
13     padding:0;
14 }
15
16 nav{
17     float:left;
18     background:darkgray;
19     width:200px;
20 }
21
22 main{
23
24     background:lightgray;
25     width:500px;
26     float:left;
27
28 }
29
30 footer{
31     background:gray;
32 }
33
34
35 </style>
```

Clear property

We willen de footer onder beide elementen

Daar zorgt clear voor

Daarmee forceren we het footer element onder een floating element te komen

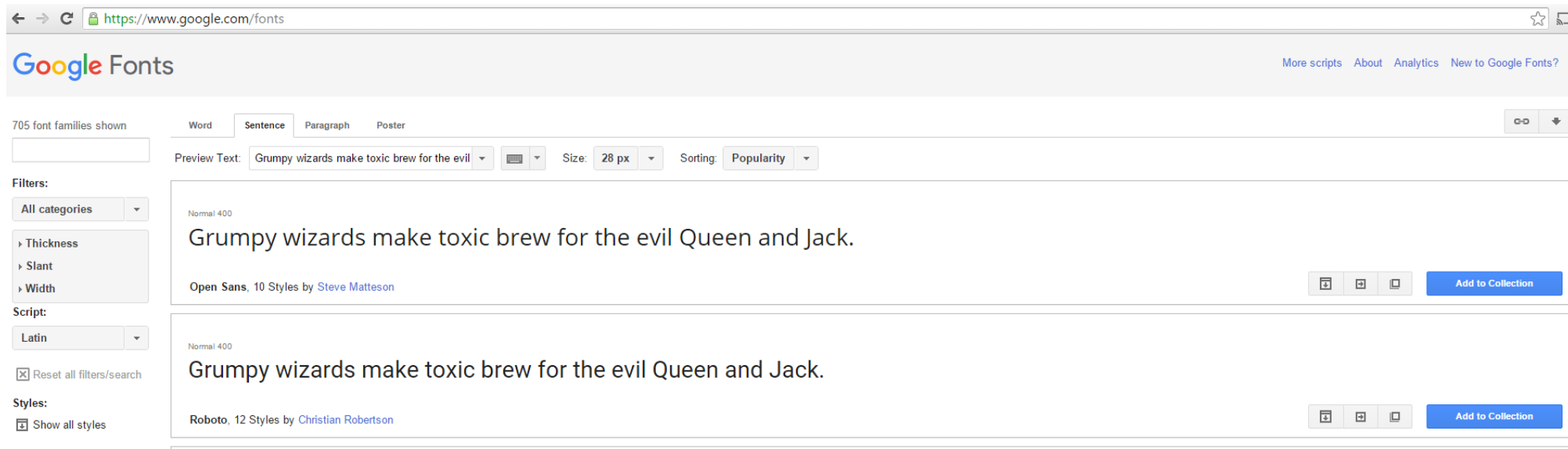
```
9- <style>
10
11-  *{
12      margin:0;
13      padding:0;
14  }
15
16-  nav{
17      float:left;
18      background:darkgray;
19      width:200px;
20  }
21
22-  main{
23
24      background:lightgray;
25      width:500px;
26      float:left;
27
28  }
29
30-  footer{
31      background:gray;
32      clear:both;
33  }
34
35 </style>
--
```

Werken met fonts

Web fonts is een geweldige uitvinding

Bijvoorbeeld google hosts een groot aantal fonts en hen gebruiken is zeer eenvoudig

Ga zelf op zoek: op de site van Google zie je een lijst van lettertypes waaruit je kan kiezen



Fonts

```
35- body{
36   font-family: 'Open Sans', sans-serif;|
37 }
38 </style>
39
40 <link href='https://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
41 </head>
42
43- <body>
44
45-   <nav>
46-     <p>
47       <a href="#">link1</a>
48     </p>
```

Standard

@import

Javascript

3. Add this code to your website:

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
```

Browser “resetten”

Een tool om alles op 0 te zetten

Zo vertrek je bij alle browsers op gelijke voet!

Verschillende reset-stylesheets

Bekendste dit van Eric Meyer

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