

## HTML week 1

### 1. Het internet

- DNS: *Domain Name Server* (koppelt een URL aan een IP-adres.)
- IP: *Internet Protocol* (denk ook aan IP-adres.)
- HTTP: *HyperText Transfer Protocol*
- Client/server: *Browser/server (hosting, zie ook punt 7.)*

### 2. HTML

- HTML: *HyperText Markup Language*
- Structuur: Een HTML-pagina is gestructureerd, net als een krant of een verzekeringsformulier.

### 3. Tags

- Tags: HTML-elementen (*tags*) zijn te herkennen aan de punthaken.
- `<html> </html>`: Alles tussen deze tags is HTML-code.
- `<body> </body>`: Alles tussen deze tags wordt getoond in het browservenster.
- `<head> </head>`: Alles tussen deze tags bestaat uit zogenaamde *metagegevens*, zoals een paginatitel.
- `<title> </title>`: Tussen deze tags wordt de titel van de webpagina bepaald.
- `<h1> </h1>`: Woorden tussen deze tags zijn *main-headings* (zoals hoofdstukaanduidingen).
- `<h2> </h2>`: Woorden tussen deze tags zijn *sub-headings* (zoals paragraafaanduidingen).  
Zie ook `<h3>`, `<h4>`, `<h5>` en `<h6>` op pagina 43 van het boek.
- `<p> </p>`: Alles tussen deze tags is tekst binnen een paragraaf.
- `<b> </b>`: Alles tussen deze tags wordt **vetgedrukt** weergegeven.
- `<i> </i>`: Alles tussen deze tags wordt *cursief* weergegeven.
- `<sup> </sup>`: Alles tussen deze tags is <sup>superscript</sup>.
- `<sub> </sub>`: Alles tussen deze tags is <sub>subscript</sub>.
- `<br />`: De *line break-tag* laat de erop volgende tekst op een nieuwe regel beginnen.
- `<hr />`: De *horizontal rule-tag* zorgt voor een horizontale lijn.

### 4. Semantische tags (zie ook attributen bij punt 5 en in hoofdstuk 2 van het boek)

- `<strong> </strong>`: Alles tussen deze tags is **extra belangrijke informatie**.
- `<em> </em>`: Alles tussen deze tags wordt *benadrukt*.
- `<blockquote> </blockquote>`: Alles tussen deze tags is tekst een (lang) citaat en wordt ingesprongen.
- `<q> </q>`: Alles tussen deze tags is tekst een “kort citaat”.
- `<abbr> </abbr>`: Alles tussen deze tags is een afkorting of een acroniem (zoals NASA).
- `<cite> </cite>`: Alles tussen deze tags is een *verwijzing* (bijv. naar een boektitel).
- `<del> </del>`: Alles tussen deze tags is wordt ~~vervangen~~, door;
- `<ins> </ins>`: De tekst die tussen deze tags wordt geplaatst.

- `<!-- -->`: Alles tussen deze tags is commentaar bij HTML-code.

## 5. Attributen

- Attributen: Attributen vertellen ‘iets’ over een tag. (Ze geven er een eigenschap aan mee.)  
Voorbeeld: `<p lang="en-us">Paragraph in English</p>`
- Attribuutnaam: De naam van een attribuut, zoals **lang** (taal) in bovenstaand voorbeeld.
- Attribuutwaarde: De waarde van een attribuut, zoals **en-us** in bovenstaand voorbeeld.

## 6. De geschiedenis van de HTML-standaard

- HTML-4: HTML-4 werd in 1997 in gebruik genomen.
- XHTML-1.0: XHTML-1.0 werd in 2000 in gebruik genomen en was strikter (‘strenger’) dan HTML-4.
- HTML-5: HTML-5 neemt veel van HTML-4 over en is nog in ontwikkeling.
- Document-type: Omdat er meerdere versies van HTML zijn, moet een HTML-document starten met de declaratie van het type. In HTML-5 is dat eenvoudig: `<!DOCTYPE html>` (zie ook pagina 181 van het boek).

## 7. Domeinnamen, hosting en FTP

- Domeinnaam: Het webadres, zoals *google.com* of *nu.nl*. Een DNS is een soort telefoonboek en bevat de koppeling tussen domeinnaam en bijbehorend IP-adres.
- Hosting: Een (*web-*) *server* is een computer met een IP-adres waarop bijvoorbeeld de bestanden van een website staan. *Hosting* betekent het opslaan en toegankelijk maken van die bestanden.
- FTP: *File Transfer Protocol*. Een protocol om gemakkelijk bestanden op bijvoorbeeld een webserver te plaatsen.

## 8. Syntaxis en semantiek (recapitulatie)

- Syntaxis: De vorm (soms ‘welgevormdheid’) van een tag (of grotere eenheid). Een voorbeeld:  
De naam `<b>Peter</b>` is belangrijk.      De naam `<b>Peter</b>` is belangrijk.  
Syntactisch incorrect      Syntactisch correct  
(*niet welgevormd*)      (*welgevormd*)
- Semantiek: De betekenis die aan een tag (of grotere eenheid) wordt toegekend.  
De naam `<b>Peter</b>` is belangrijk.      De naam `<strong>Peter</strong>` is belangrijk.  
Semantisch ontoereikend (incorrect?)      Semantisch correct  
(*niet informatief*)      (*informatief*)

## 9. Lijsten (*lists*)

- `<ol> </ol>` Geordende lijst. De items (`<li> </li>`) in een geordende lijst voldoen aan een volgorde of rangorde, zoals:  

1.	Item 1	a.	Item 1
2.	Item 2	b.	Item 2
3.	Item 3	c.	Item 3
- `<ul> </ul>`: Ongeordende lijst. De items (`<li> </li>`) in een ongeordende lijst voldoen niet aan een volgorde of rangorde, zoals:
  - Item 1
  - Item 2
  - Item 3
- `<li> </li>`: De tekst tussen deze tags vormt een lijstitem (*list item*).
- `<dl> </dl>`: Definitielijst. De definitietermen (`<dt> </dt>`) in een definitielijst vormen de lemma’s voor definities (`<dd> </dd>`) in een verklarende lijst, zoals:

### Epibreren

Niet nader aan te geven werkzaamheden verrichten (waarvan men de indruk wil geven dat ze belangrijk zijn, ook al stellen ze helemaal niets voor).

### Interpunctie

Het plaatsen en de wijze van plaatsing van de leestekens.

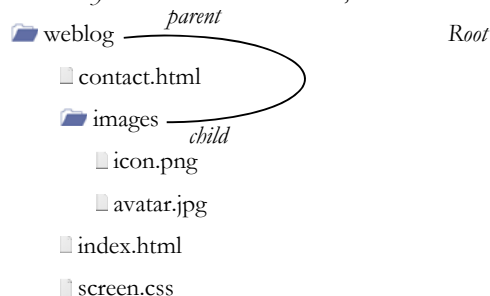
- *Nested lists*: Een lijst (`<ol> </ol>` of `<ul> </ul>`) kan opgenomen worden binnen een lijstitem (`<li> </li>`). Dit verschijnsel heet ‘*nesting*’ en kan ook worden toegepast op andere tags.
  - Item 1
    - Item 1.1
    - Item 1.2
  - Item 2

## 9. Links

- URL: *Uniform Resource Locator*. Het unieke adres van een webpagina.
- Link: Een verwijzing binnen een webpagina, naar een andere webpagina of naar een ander bestand.
- `<a> </a>`: Alles tussen deze tags wordt weergegeven als een link en is 'klikbaar'.  
`<a href="http://www.imdb.com">Internet Movie DataBase</a>`
- href: Het *href*-attribuut bevat de doellocatie van de link (`http://www.imdb.com`).
- Absolute link: Een absolute link specificeert de gehele *URL*, zoals in het bovenstaande voorbeeld. In de praktijk verwijst een absolute link vaak naar een andere website (extern).
- Relatieve link: Een relatieve link specificeert de *URL* vanaf het bestand waarin de link voorkomt. In de praktijk verwijst een relatieve link vaak naar een webpagina binnen een website (intern).  
`<a href="contact.html">contactformulier</a>`
- Link binnen pagina: Binnen een webpagina kan worden verwezen naar een *anchor* (#) op dezelfde webpagina.  
`<h2 id="inventarisatie">Een inventarisatie van soorten</h2>`  
Elders op de webpagina:  
`<a href="#inventarisatie">Naar de inventarisatie</a>`  
NB Er kan ook worden verwezen naar een *anchor* op een andere (externe) webpagina:  
`<a href="onderzoek.html#inventarisatie">Naar de inventarisatie</a>`
- target: Het *target*-attribuut specificeert of de doellocatie wordt geopend in het huidige venster of tabblad (`_self`) of in een nieuw venster of tabblad (`_blank`).  
`<a href="http://www.imdb.com" target="_blank">Internet Movie DataBase</a>`
- mailto: Een *mailto*-link opent het standaard-e-mailprogramma met een nieuw bericht aan het gespecificeerde adres, zoals:  
`<a href="mailto:peter@domein.nl">E-mail Peter</a>`

## 10. Directory-structuur

- Root: Het hoofdniveau in een boomstructuur.
- Structuur: Een *directory*-structuur is hiërarchisch, zoals in onderstaand overzicht.



- Relaties: *Parent, child, grandparent, grandchild*. (Zie p. 82 in het boek.)
- Pauzeopdracht: Trek de ontbrekende lijntjes die de relaties tussen bovenstaande items representeren en schrijf het type relatie erbij.

## 11. Afbeeldingen

- `<img />`: Met de *img-tag* wordt een afbeelding in een webpagina geplaatst.  
``
- **src**: Het *src*-attribuut verwijst naar de locatie van de afbeelding.
- **alt**: Het *alt*-attribuut bevat een korte omschrijving van de afbeelding.
- **title**: Het *title*-attribuut bevat toegevoegde informatie over de afbeelding.
- **height en width**: De *height*- en *width*-attributen specificeren respectievelijk de hoogte en de breedte van de afbeelding. Aangeraden wordt hiervoor CSS te gebruiken.
- **Afbeeldingstypen**: *jpeg* (voor foto's e.d.), *gif* (voor logo's e.d.), *png* (idem, wordt vaak gebruikt voor afbeeldingen met transparantie).

## 12. Tabellen

- **Tabel**: Een tabel toont gestructureerde gegevens, zoals hieronder.

Studentnummer	Naam	Propedeuse (ja/nee)
12001234	Peter de Jong	Nee
12005678	Willemijn den Ouden	Ja
12019012	Bertolt Brecht	Ja
- **Structuur**: Een tabel bestaat altijd uit rijen (*table rows*), die op hun beurt weer uit cellen (*table data*) bestaan. Elke kolom heeft als eerste cel een hoofd (*table heading*).
- `<table> </table>`: Alles tussen deze tags is onderdeel van de tabel.
- `<tr> </tr>`: *Table row*. Alles tussen deze tags representeert een rij in de tabel. In het bovenstaande voorbeeld is dat bijvoorbeeld de rij met de kolomhoofden of een rij met een specifieke student (zoals de rik {12001234, Peter de Jong, Nee}).
- `<td> </td>`: *Table data*. Alles tussen deze tags is tekst een cel binnen een tabelrij. In het bovenstaande voorbeeld is de inhoud van een cel bijvoorbeeld '12005678', maar ook 'Ja' en 'Bertolt Brecht'.
- `<th> </th>`: *Table heading*. Alles tussen deze tags vormt het hoofd van een kolom binnen een tabelrij. In het bovenstaande voorbeeld zijn er dus drie tabelhoofden, te weten **Studentnummer**, **Naam** en **Propedeuse (j/n)**.  
Let op! Ook een *th*-tag is onderdeel van een *table row* en valt daar dus nooit buiten.
- *Column spanning*: Met *column spanning* (het *colspan*-attribuut van een cel) wordt een cel over twee of meer kolommen verdeeld.

	8.00 uur	9.00 uur	10.00 uur
Maandag	Studeren	Ontbijten	Examen maken
Dinsdag	Studeren	Ontbijten	

- *Row spanning*: Met *row spanning* (het *rowspan*-attribuut van een cel) wordt een cel over twee of meer rijen verdeeld.

	8.00 uur	9.00 uur	10.00 uur
Maandag	Studeren	Ontbijten	Examen maken
Dinsdag		Hardlopen	Ontbijten

- Uitgebreide tabellen: De structuur (en semantiek) van een tabel kan worden verduidelijkt door middel van de *table head*- (<thead> </thead>), *table body*- (<tbody> </tbody>) en *table foot-tags* (<tfoot> </tfoot>). (Zie p. 135, 136 in het boek.)

### 13. Formulieren (*forms*)

- **Formulier:** Een HTML-formulier kan worden gebruikt voor het vergaren van informatie van bezoekers/gebruikers, middels formulervelden (zie hieronder). Denk daarbij aan NAW-gegevens, maar bijvoorbeeld ook aan een zoekterm op Google.
- **Gegevens-overdracht:** De gegevens die middels een formulier worden ingevuld, kunnen worden verstuurd naar een server. (Zie ook het *method*-attribuut.) (Het verwerkingsproces op de server valt buiten het bereik van dit vak.)
- **<form> </form>:** Alles tussen de *form*-tags is onderdeel van het formulier. De gegevensoverdracht is afhankelijk van de attributen *method* en *action*.
- **method:** Het *method*-attribuut van de *form*-tag specificeert hoe de gegevensoverdracht plaatsvindt. De attribuutwaarde dient **get** of **post** te zijn. Met de *get*-methode worden de gegevens uit het formulier via de URL verstuurd. Deze gegevens zijn zichtbaar voor gebruikers. Met de *post*-methode worden de gegevens uit het formulier via de zogenaamde *http-headers* verstuurd, waardoor ze onzichtbaar zijn voor gebruikers. Zie pagina 151 in het boek voor vuistregels.
- **action:** Het *action*-attribuut specificeert de doellocatie van het formulier, d.w.z. welk bestand de afhandeling van de verstuurde gegevens verzorgt. Denk hierbij aan het opslaan van gegevens in een database of het e-mailen van de gegevens. (Dit proces valt buiten het bereik van dit vak.)
- **id:** Het *id*-attribuut specificeert een unieke identificatie voor het formulier. Hier wordt later, in het kader van *CSS*, op teruggekomen.

### 14 Formulervelden (*form controls*)

- **<input />:** De *input*-tag wordt gebruikt om verschillende typen formulervelden te plaatsen. De verschillende typen worden gespecificeerd in het *type*-attribuut.
- **name:** Het *name*-attribuut specificeert welke gegevens in een formulerveld worden ingevuld. Denk daarbij aan *voornaam*, *achternaam*, *rekeningnummer* etc.
- **maxlength:** Het *maxlength*-attribuut specificeert hoeveel tekens in het betreffende formulerveld kunnen worden ingevuld. Zie het voorbeeld hieronder.
- **type:** Het *type*-attribuut specificeert het type invoerveld, zoals hieronder.  

```
<input type="text" name="postcode" maxlength="6" />
```

Mogelijke typen en daarbij horende attributen zijn:
  - **text:** Een tekstveld van één regel – **name**, **maxlength**, **value**.
  - **password:** Een tekstveld van één regel, waarin de ingevoerde tekens worden verborgen en worden weergegeven als \*- of ●-tekens – **name**, **maxlength**, **value**.

- **radio:** Een selectie van keuzemogelijkheden, waarbij maar één mogelijkheid kan worden geselecteerd – **name, value, checked**. Het *name*-attribuut moet voor een groep *radiobuttons* gelijk zijn (een groep representeert één keuze). Een voorbeeld van gebruik is de vraag naar geslacht (*man* of *vrouw*, maar niet allebei).
  - **checkbox:** Een keuzemogelijkheid, waarbij meer mogelijkheden kunnen worden geselecteerd – **name, value, checked**. Het *name*-attribuut moet voor een groep *checkboxes* gelijk zijn (een groep representeert één keuze). Een voorbeeld van gebruik is de vraag naar koffiegebruik (*melk* of *suiker*, of allebei).
  - **submit:** De knop die de formulierverzending start – **name, value**. Het *value*-attribuut representeert de waarde die op de knop staat vermeld.
  - **image:** De knop die de formulierverzending start. De standaardknop (**submit**) wordt vervangen door een gespecificeerde afbeelding – **name, value, src, width, height, alt**.
  - **file:** Een invoerveld waarmee, door middel van de getoonde *browse*-knop, een bestand kan worden geselecteerd – **name**.
  - **hidden:** Een verborgen tekstveld – **name, maxlength, value**.
  - **date:** Een tekstveld waarin een datum, volgens geldige notatie, kan worden ingevoerd – **name, value**.
  - **email:** Een tekstveld waarin een e-mailadres, volgens geldige notatie, kan worden ingevoerd – **name, value**.
  - **url:** Een tekstveld waarin een URL, volgens geldige notatie, kan worden ingevoerd – **name, value**.
  - **search:** Een tekstveld waarin een zoekterm kan worden ingevoerd – **name, value**.
- 
- **<textarea>**  
**</textarea>:** De *textarea*-tag wordt gebruikt om een tekstveld te tonen, waarin meerdere regels kunnen worden ingevoerd. De naam van het formulierveld wordt gespecificeerd in het *name*-attribuut.
  - **<select>**  
**</select>:** De *select*-tag wordt gebruikt om een zogenaamde *dropdownlistbox* te tonen. Attributen bij deze tag zijn **size** (hoeveel opties worden getoond) en **multiple** (hoeveel opties kunnen worden geselecteerd). Binnen de *select*-tags worden *option*-tags opgenomen.
  - **<option>**  
**</option>:** De *option*-tag wordt gebruikt om een keuzemogelijkheid binnen een *dropdownlistbox* op te nemen. Het *value*-attribuut specificeert de keuzewaarde. Tussen de tags wordt de te tonen waarde gespecificeerd.
  - **<label>**  
**</label>:** De *label*-tag specificeert een formulierveld, zodat duidelijk is om welke informatie wordt gevraagd. Het *for*-attribuut specificeert bij welk formulierveld het label hoort.
  - **<fieldset>**  
**</fieldset>:** De *fieldset*-tag specificeert informatiegroepen binnen een formulier. (Denk aan contactgegevens {naam, adres, woonplaats}, studiegegevens {instelling, studie, startjaar,



- `<legend>`  
`</legend>` eindjaar}, opmerkingen {onderwerp, opmerking} etc.)  
De *legend*-tag specificeert de naam (titel) van een *fieldset*. Zie ook pagina 164 in het boek.

## 15 Media (*iframes*, audio, video)

- `<iframe>`  
`</iframe>`: Binnen een *iframe* wordt een (functie van een) andere website, zoals een kaart op *Google Maps*, getoond. Het *src*-attribuut specificeert wat er in een *iframe* wordt getoond. De *height*- en *width*-attributen specificeren respectievelijk de hoogte en breedte van een *iframe*. Andere attributen zijn **scrolling**, **frameborder** en **seamless**. Zie pagina 190 in het boek.
- *Flash*: *Flash* kan worden gebruikt voor het opnemen van (al dan niet interactieve) multimedia op een webpagina. (Zie pagina's 202 – 212 in het boek voor achtergrondinformatie.)
- `<video>`  
`</video>`: De *video*-tag wordt gebruikt om een video te tonen. Attributen bij de *video*-tag zijn **src**, **poster**, **controls**, **width**, **height**, **controls**, **autoplay**, **loop** en **preload**. Zie pagina's 213 – 216 in het boek.
- `<audio>`  
`</audio>`: De *audio*-tag wordt gebruikt om audio te 'tonen'. Attributen bij de *audio*-tag zijn **src**, **controls**, **autoplay**, **preload**, **loop**. Zie pagina's 219 – 220 in het boek.
- `<source>`  
`</source>`: De *source*-tag wordt, binnen de *audio*-tags, gebruikt om meerdere geluidsbestanden te specificeren. Attributen bij de *source*-tag zijn **src** en **type**.

## 16. Id's en klassen (*id*- en *class*-attributen)

- *id*-attribuut: Elk HTML-element kan gespecificeerd worden door middel van een *id*-attribuut. Met dit attribuut krijgt een HTML-element een unieke aanduiding binnen een HTML-pagina. Belangrijk hierbij is dat de attribuutwaarde (de daadwerkelijke identificatiewaarde) maar eenmaal op dezelfde pagina mag voorkomen. Het onderstaande HTML-element mag dus slechts één keer op een webpagina voorkomen. Met *CSS* (en later met *JavaScript*) maken we dankbaar gebruik van het *id*-attribuut.  
`<p id="postmodernisme">Willem Brakmans postmoderne literatuur...</p>`
- *class*-attribuut: Waar met het *id*-attribuut een unieke waarde aan een HTML-element wordt toegekend, specificeert het *class*-attribuut welke klasse aan een HTML-element wordt toegekend. Meerdere HTML-elementen kunnen dezelfde klasse toegewezen krijgen. Zo kan vervolgens met CSS gemakkelijk een opmaak worden gespecificeerd voor alle elementen van dezelfde klasse binnen een webpagina.  
`<p class="literatuurstroming">Willem Brakmans postmoderne literatuur...</p>`  
NB klassen en id's kunnen worden gecombineerd.

### **Block- en inline-niveau**

- Blokelementen: Blokelementen worden in de browser standaard op een nieuwe regel weergegeven (onder elkaar). Voorbeelden van blokelementen zijn `<h1>`, `<p>`, `<ul>` en `<li>`.
- *Inline*-elementen: *Inline*-elementen worden in de browser standaard binnen een regel weergegeven (naast elkaar). Voorbeelden van *inline*-elementen zijn `<a>`, `<b>`, `<em>` en `<img>`.

## CSS

- *Cascading StyleSheets (CSS)*: *Cascading stylesheets* ('overervende stijlbladen') beschrijven de visuele opmaak van HTML-elementen. Denk dan aan typografie, kleur en *lay-out*.
- Opmaaktaal: CSS is een opmaaktaal en verschilt in syntaxis van HTML.
- Elementen: Ieder HTML-element, zowel een blok- als *inline*-element, kan worden opgemaakt door middel van CSS. Hieronder wordt een *h1-tag* opgemaakt.  
HTML: `<h1>De reis van de douanier naar Bentheim</h1>`  
CSS: `h1 {  
 font-family: Arial;  
 color: red;}  
h1 {  
 font-family: Arial;  
 color: red;}`
- *Selectors*: In de bovenstaande CSS-code is `h1` de *selector*. De *selector* is van toepassing op alle *h1*-elementen (alle *h1*-tags) binnen een HTML-pagina. De stijlregels die erbij horen worden declaraties genoemd.
- *Declaraties*: Declaraties specificeren de opmaak van het HTML-element waarnaar de *selector* verwijst.

Een declaratie bestaat uit een eigenschap of *property* (**font-family, color**) en een waarde of *value* (**Arial, red**).

- Interne CSS: Interne CSS wordt opgenomen in de *head*-sectie van een HTML-pagina (de opmaak ‘zegt’ immers iets over de inhoud van de pagina en is daarmee een metagegeven), middels de *style*-tag. Middels het *type*-attribuut dient te worden gespecificeerd dat het type “text/css” is.

...

```
<head>
    <title>Brakman, Willem - De reis van de douanier...</title>
    <style type="text/css">
        h1 {
            font-family: Arial;
            color: red;}
    </style>
</head>
```

...

NB Gebruik interne CSS slechts dan als een website uit niet meer dan een pagina bestaat.

- Externe CSS: Bij externe CSS wordt in de *head*-sectie van een HTML-pagina, middels de *link*-tag een verwijzing naar een CSS-bestand (.css) opgenomen. De attributen **href, type** en **rel** specificeren respectievelijk het pad naar het CSS-bestand, het type (ook hier “text/css”) en het type relatie (in dit geval “stylesheet”).

...

```
<head>
    <title>Brakman, Willem - De reis van de douanier...</title>
    <link href="css/styles.css" type="text/css" rel="stylesheet" />
</head>
```

...

NB Gebruik externe CSS als een website uit meer dan een pagina bestaat.

### ***CSS-selectors***

- *Universal selector*: Een *universal selector* is van toepassing op alle elementen binnen een HTML-pagina.  

```
* {
    font-family: Arial;}
```
- *Type selector*: Een *type selector* is van toepassing op het gelijknamige HTML-element.  

```
h1, h2 {
    color: green;}
```

- *Class selector:* Een *class selector* is van toepassing op HTML-elementen van de genoemde klasse en wordt gespecificeerd door middel van het `.`-teken.  
HTML: `<p class="author">Dit verhaal is geschreven door W. Brakman.</p>`  
CSS: `.author {`  
`font-style: italic;}`  
Ook specifieke element-klasse-combinaties kunnen worden aangesproken.  
`p.author {`  
`font-style: italic;}`  
De bovenstaande declaratie is alleen van toepassing op *p-tags* die de klassenwaarde “author” hebben.
- *Pseudo-class selector:* Een *pseudo-class selector* is van toepassing op HTML-elementen in een ‘bepaalde staat’. Denk hierbij aan een link waarop geklikt is. (Zie pagina’s 290 en 291 in het boek.)  
`a:visited {`  
`color: purple;}`
- *Id selector:* Een *id selector* is van toepassing op HTML-elementen met de genoemde (unieke) *id*-waarde en wordt gespecificeerd door middel van het `#`-teken.  
HTML: `<h1 id="top"> De reis van de douanier naar Bentheim...</h1>`  
CSS: `#top {`  
`color: purple;}`
- *Child selector:* Een *child selector* is van toepassing op HTML-elementen die direct binnen de *parent* staan (ouder-kind-relatie).  
`li>a {`  
`font-style: italic;}`  
Door middel van de bovenstaande *selector* en declaratie worden alle links die direct in een *list-item* staan gecursiveerd.
- *Descendant selector:* Een *descendant selector* is van toepassing op HTML-elementen die direct of indirect binnen de *parent* staan (grootouder-kind-kleinkind-relatie).  
`li a {`  
`font-style: italic;}`  
Door middel van de bovenstaande *selector* en declaratie worden alle links die direct of indirect in een *list-item* staan gecursiveerd (zoals links die binnen een *list-item* en binnen een *strong-tag* staan).
- *Overerving:* Binnen CSS is het concept van overerving zeer belangrijk. Sommige eigenschappen of *properties* gelden niet alleen voor de HTML-elementen van hun *selector*, maar ook voor hun *descendants*. Een voorbeeld hiervan is de *font-family*-eigenschap. Middels de onderstaande CSS-code worden (bijna) alle HTML-elementen, zoals paragrafen en links, in lettertype Arial weergegeven.  
`body {`  
`font-family: Arial;}`

Wanneer twee of meer *selectors* (direct of indirect) van toepassing zijn op een HTML-element, zijn de onderstaande regels van toepassing.

- a. Een regel overschrijft een ervoor gespecificeerde regel.
- b. Een specifieke(re) regel overschrijft een minder specifieke (generieke) regel.
- c. **!important** overschrijft regels **a** en **b**.

```
p b {  
    color: blue !important;}  
  
p b {  
    color: yellow;}
```

NB Zie bladzijden 239 en 240 in het boek voor voorbeelden.

## Kleur

- *Color*: De *color*-eigenschap specificeert de kleur van tekst binnen een HTML-element.  

```
p {  
    color: grey;}
```
- *Background-color*: De *background-color*-eigenschap specificeert de achtergrondkleur van een HTML-element.  

```
h1 {  
    background-color: yellow;}
```

De kleurwaarde wordt gespecificeerd door middel van een kleurnaam, een RGB-waarde of een HEX-waarde.
- *Kleurnamen*: Er zijn 147 kleurnamen waarmee kleurwaarden kunnen worden aangegeven.  

```
h2 {  
    color: red;}
```
- *RGB*: Een kleur kan worden gespecificeerd middels een *RGB*-waarde. (*RGB* staat voor rood, groen en blauw.) Elke deelwaarde binnen de *RGB*-waarde ligt tussen 0 (minimum) en 255 (maximum).  

```
h2 {  
    color: rgb(255,0,0);}
```
- *HEX*: Een kleur kan worden gespecificeerd middels een *HEX*-waarde. De waardenparen binnen de *HEX*-waarde representeren R, G en B en liggen tussen 0 (minimum) en c (maximum). Een *HEX*-waarde wordt aangeduid met het #-teken.  

```
p {  
    color: #ee0000;}
```
- *HSL en HSLA*: Een kleur kan worden gespecificeerd middels een *HSL*-waarde. *HSL* staat voor hue, saturation en lightness. Bij *HSLA* wordt de doorzichtigheid toegevoegd middels de alpha-waarde. (Zie pagina's 255 en 256 in het boek voor uitleg.)  

```
body {  
    background-color: hsl(15,100%,100%,0.5;}
```
- *Opacity*: De *opacity*-eigenschap specificeert de doorzichtigheid van een HTML-element. De *opacity*-waarde ligt tussen 0.0 (minimum) en 1.0 (maximum).

```
p {
```

```
background-color: red;
```

```
opacity: 0.5;}
```

## 21. Typografie

### – *Sans-serif*:

*Sans-serif* betekent schreefloos. Schreefloze lettertypen kennen geen zogenaamde schreef (de dwarsstrepen aan de horizontale en verticale uiteinden van de letters). Schreefloze letters zijn geschikt voor beeldschermtekst en minder voor gedrukte tekst. Een voorbeeld van een schreefloos lettertype is Helvetica (zie hieronder).

Abcdefg

### – *Serif*:

*Serif*-lettertypen kennen wel een schreef. Lettertypen met schreef zijn geschikt voor gedrukte tekst en minder voor beeldschermtekst. Een voorbeeld van een schreeflettertype is Times New Roman (zie hieronder).

Abcdefg

### – *Monospace*:

Bij *monospace*-lettertypen is elke letter van dezelfde breedte. Vooral voor code, zoals HTML- en CSS-code, is een *monospace*-lettertype geschikt, omdat e.e.a. zo goed uitlijnt. Een voorbeeld van een *monospace*-lettertype is Courier New.

Abcd

Mnwwj

### – Lettertypen:

Een lettertype of *font* dient op de *client* aanwezig te zijn (te zijn geïnstalleerd), om te kunnen worden weergegeven. Standaardlettertypes zijn bijvoorbeeld Arial, Times New Roman en Helvetica. Een niet-standaard-lettertype kan worden gebruikt middels de *font-face*-eigenschap. Zo kunnen lettertypen op de server of bijvoorbeeld *Google-fonts* worden gebruikt.

```
@font-face {
```

```
font-family: 'ChunkFiveRegular';
```

```
src: url('fonts/chunkfive.eot');}
```

```
h1 {
```

```
font-family: ChunkFiveRegular, Georgia, serif;}
```

NB Zie pagina 277 in het boek.

- *weight*: De *font-weight*-eigenschap (*light*, *medium*, *bold* of *black*) specificeert de dikte van een letter.

```
p {  
    font-weight: medium;}  
}
```

- *style*: De *font-style*-eigenschap (*normal*, *italic* of *oblique*) specificeert of een letter cursief wordt weergegeven.

```
.remark {  
    font-style: italic;}  
}
```

- *stretch*: De *font-stretch*-eigenschap (*condensed*, *regular*, of *extended*) specificeert hoe breed een letter en de tussenruimte zijn.

```
body {  
    font-stretch: extended;}  
}
```

- *size*: De *font-size*-eigenschap specificeert (in pixels, percentages of *ems*) hoe groot een letter wordt weergegeven.

```
body {  
    font-size: 12px;}  
  
h1 {  
    font-size: 200%;}  
  
h2 {  
    font-size: 1.3em;}  
}
```

NB *ems* specificeert de grootte van een letter relatief ten opzichte van de standaardgrootte van de letter m. Zie ook pagina's 274 – 276 in het boek voor een duidelijk overzicht.

- *Align*: De *text-align*-eigenschap (*left*, *right*, *center* en *justify*) specificeert de uitlijning van de tekst. (De tekst in de kaders is afkomstig uit Brakman 1983: 5.)

Kort geleden ben ik om redenen die hier niet ter zake doen uit een stadje gemieterd waarvan ik de naam maar niet zal noemen.	Kort geleden ben ik om redenen die hier niet ter zake doen uit een stadje gemieterd waarvan ik de naam maar niet zal noemen.	Kort geleden ben ik om redenen die hier niet ter zake doen uit een stadje gemieterd waarvan ik de naam maar niet zal noemen.	Kort geleden ben ik om redenen die hier niet ter zake doen uit een stadje gemieterd waarvan ik de naam maar niet zal noemen.
---	---	---	---

*Left*

*Right*

*Center*

*Justify*

- Andere tekst-eigenschappen: Andere tekst-eigenschappen zijn *transform* (p. 281), *decoration* (p. 282), *line-height* (p. 283), *letter-* en *word-spacing* (p.284) en *indent* (p. 287). Zie de respectievelijke pagina's in het boek voor uitleg.

## 22. Groeperen (*div*- en *span*-elementen)

- **<div> </div>:** Het HTML-element *div* is een *block-level* element (zie hand-out bij het vorige college) waarmee andere HTML-elementen kunnen worden gegroepeerd. Het *div*-element kent geen eigen semantiek. De opmaak en positionering van het element worden verzorgd door CSS, met behulp van het klasse- en/of *id*-attribuut in HTML. Het *div*-element wordt bijvoorbeeld gebruikt om een menu, bestaande uit lijstitems, te positioneren. Zie pagina 187 in het boek voor een voorbeeld.
- **<span> </span>:** Het HTML-element *span* is een *inline-level* element (in tegenstelling tot *div*, zie tevens hand-out bij het vorige college) waarmee andere HTML-elementen kunnen worden gegroepeerd of opgemaakt. Het *span*-element kent geen eigen semantiek. De opmaak en de positionering van het element worden verzorgd door CSS, met behulp van het klasse- en/of *id*-attribuut in HTML. Het *span*-element wordt bijvoorbeeld gebruikt om een tekst, waarvoor geen specifieke HTML-tag bestaat, op te maken. Zie pagina 188 in het boek voor een voorbeeld.

## 23. Boxes

- **Boxes:** Ieder HTML-element wordt, als het ware, als een omkaderd *block*- of *inline*-element beschouwd. Zo'n element lijkt op een doosje of een *box*, zoals in onderstaand voorbeeld. (De tekst in de paragraaf en in het kader is afkomstig uit Drs. P 1973.)

### HTML-code

<p>

Wij zijn hier aan de oever van een machtige rivier <br />  
De andere oever is daarginds, en deze hier is hier <br />  
De oever waar we niet zijn noemen wij de overkant <br />  
Die wordt dan deze kant zodra we daar zijn aangeland

</p>

### Weergave met *denkbeeldig kader*

Wij zijn hier aan de oever van een machtige rivier  
De andere oever is daarginds, en deze hier is hier  
De oever waar we niet zijn noemen wij de overkant  
Die wordt dan deze kant zodra we daar zijn aangeland

- **Eigenschappen:** Ook het (denkbeeldige) kader of de *box* kent eigenschappen in CSS. Zo kunnen de (minimale) breedte (**min-width**, **width**) en (minimale) hoogte (**min-height**, **height**) etc. worden gespecificeerd.
- **Breedte:** De breedte van *boxes* kan worden gespecificeerd middels de *width*-eigenschap. De breedte



kan worden opgegeven in *pixels* (**px**) of een percentage (%). Een minimale breedte kan op dezelfde manier worden gespecificeerd, middels de *min-width*-eigenschap.

- *Hoogte*: De hoogte van *boxes* kan worden gespecificeerd middels de *height*-eigenschap. De hoogte kan worden opgegeven in *pixels* (**px**) of een percentage (%). Een minimale hoogte kan op dezelfde manier worden gespecificeerd, middels de *min-height*-eigenschap.

- *overflow*: Als de inhoud van een HTML-element (zoals een paragraaf) niet binnen een *box* past, kan middels de *overflow*-eigenschap worden gespecificeerd hoe met het teveel aan inhoud wordt omgegaan. De waarde *hidden* geeft aan dat het teveel verborgen (niet weergegeven) wordt, de waarde *scroll* voegt een *scrollbar* toe.

- *border*: Middels de verzameling *border*-eigenschappen kan een rand (*border*) om een *box* worden weergegeven.

- *border-width*: dikte van de rand in termen (*thin*, *medium*, *thick*) of in pixels.
- *border-style*: stijl van de rand (*solid*, *dotted*, *dashed*, *double*, *groove*, *none* etc.).
- *border-color*: de kleur van de rand in termen, RGB-, HEX- of HSL-waarden.
- *Specificatie*: ook specifieke randen (boven, onder, links, rechts) kunnen worden gespecificeerd (*border-top*, *border-bottom*, *border-left*, *border-right*).

Zie voor een verkorte notatie pagina 312 in het boek.

- *shadow*: Met de *box-shadow*-eigenschap kan een schaduw om een *box* worden gezet.

p.one {

box-shadow: 8px 3px 6px 8px #777777;}

- *Horizontal offset*: de horizontale afstand van de schaduw (8px).
- *Vertical offset*: de verticale afstand van de schaduw (3px).
- *Blur distance*: de mate van vervaging (*blur*) van de schaduw (6px). Geen specificatie van de *blur-distance* zorgt voor een strakke, ondoorzichtige ‘schaduw’.
- *Spread of shadow*: de spreiding van de schaduw (8px).
- *Kleur*: de kleur van de schaduw in termen, RGB-, HEX- of HSL-waarden (#777777).

- *Radius*: De *border-radius* specificeert de mate waarin een rand wordt afgerond aan de hoeken. Ook specifieke afrondingen (boven, onder, links, rechts) kunnen worden gespecificeerd (*border-top-left*, *border-top-right*, *border-bottom-left*, *border-bottom-right*). Zie voor een verkorte notatie pagina 321 in het boek.

- *Padding*: *Padding* specificeert de ruimte tussen de inhoud van een *box* en de (al dan niet zichtbare) rand van die *box* in *pixels* (**px**), percentage (%) of *ems* (**em**).

- *Specificatie*: ook specifieke ‘ruimten’ (boven, onder, links, rechts) kunnen worden gespecificeerd (*padding-top*, *padding-bottom*, *padding-left*, *padding-right*).

Zie voor een verkorte notatie pagina 313 in het boek.

- *Margin*: *Margin* specificeert de ruimte tussen (de randen van) verschillende elementen of *boxes* in *pixels* (**px**), percentage (%) of *ems* (**em**).

- *Specificatie*: ook specifieke ‘ruimten’ (boven, onder, links, rechts) kunnen worden

gespecificeerd (*margin-top, margin-bottom, margin-left, margin-right*).

Zie voor een verkorte notatie pagina 313 in het boek.

- *Browserimplementatie*: In *Internet Explorer 6* worden, in tegenstelling tot andere *browser*, de *margin* en de *padding* tot de grootte van de box gerekend. Zie pagina 316 in het boek.

## 24. Lijsten

- *list-style-type*: Met de *list-style-type*-eigenschap kan het type *bullets* worden gespecificeerd.
  - *Unordered list*: *none, disc, circle* of *square*
  - *Ordered list*: *decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman* of *upper-roman*
- *list-style-image*: Met de *list-style-image*-eigenschap kan een afbeelding als *bullet* worden gebruikt.  

```
ul {  
    list-style-image: url("images/star.png");  
}
```
- *list-style-position*: Met de *list-style-position*-eigenschap kan worden gespecificeerd of de *bullets* binnen (*inside*) of buiten (*outside*) de *box* worden weergegeven.  
Zie voor een verkorte notatie pagina 336 in het boek.

## 25. Tabellen


- Tabellen: Net als elk ander HTML-element, kunnen ook tabellen middels CSS worden opgemaakt.
- *width*: De *width*-eigenschap kan worden gespecificeerd voor een tabel en bepaalt daarmee de breedte van de tabel.
- *padding*: De *padding*-eigenschap kan worden gespecificeerd voor een cel (*td, th*) en bepaalt daarmee de *padding* van een cel.
- *margin*: De *margin*-eigenschap kan worden gespecificeerd voor bijvoorbeeld een cel (*td, th*) en bepaalt daarmee de *margin* van een cel.
- *text-transform*: Middels de *text-transform*-eigenschap wordt tekst omgezet in hoofdletters (*uppercase*) of kleine letters (*lowercase*). Deze eigenschap wordt vaak gebruikt voor het benadrukken van kolomkoppen (*th*).
- Andere eigenschappen: Ook andere, meer generieke eigenschappen kunnen voor tabellen worden gespecificeerd. Denk daarbij aan *font-size, border, text-align* etc. Ook pseudoklassen kunnen worden gespecificeerd, bijvoorbeeld op tabelrijen van een *highlight* te voorzien. Zie pagina's 337 tot en met 340 in het boek.

## 26. Formulieren

- Formulieren: Net als elk ander HTML-element, kunnen ook formulieren en formulievelden middels CSS worden opgemaakt.
- *input*: Middels de *input-selector* kunnen tekstvelden worden opgemaakt. (Zie pagina 342 in het boek.) Denk ook aan *textarea*'s, die niet middels de *input-selector* worden gespecificeerd.
- *focus, hover*: Middels de pseudoklassen *focus* (*cursor* in het betreffende formulierveld) en *hover* (*cursor* over het betreffende formulierveld) (`input:focus, input:hover`) kan een

formulieveld (of ander HTML-element) dat *in een bepaalde staat verkeert* worden gespecificeerd. (Zie pagina 342 in het boek.)

- *fieldset*: Middels de *fieldset-selector* kan een *fieldset* worden opgemaakt. (Zie pagina 344 in het boek.)
- *legend*: Middels de *legend-selector* kan een *legend* bij een *fieldset* worden opgemaakt. (Zie pagina 344 in het boek.)
- *label*: Middels het opmaken van het *label*-element kunnen formulievelden onder andere worden uitgelijnd. (Zie pagina's 345 en 346 in het boek.)
- *cursor*: Middels de *cursor-selector* kan de *cursor* worden gespecificeerd. Voorbeelden van mogelijke cursors zijn *crosshair*, *pointer*, *move*, *text*, *wait*, *help* of een afbeelding uit een bestand. (Zie pagina 347 in het boek.)

+			I		
crosshair	pointer	move	text	wait	help

NB Let op bij het aanpassen van de *cursor*; gebruikers verwachten bij verschillende cursors verschillende systeemreacties.

## Afbeeldingen

- Afbeeldingen: Om de opmaak (CSS) en de inhoud (HTML) gescheiden te houden, wordt CSS gebruikt voor het specificeren van de opmaak van afbeeldingen in een HTML-pagina.
- *width*, *height*: Middels de *width*- en *height*-eigenschappen kan de afmeting van een afbeelding worden gespecificeerd.

### HTML-code

```

```

### CSS-code

```
img.small {  
    width: 330px;  
    height: 210px;}
```

- *float*: Middels de *float*-eigenschap kan de positie van een afbeelding (of ander element) binnen het omvattende element worden gespecificeerd.

### HTML-code

```

```

### CSS-code

```
img.align-right {  
    float: right;  
    margin-left: 10px;}  
  
img.small {  
    width: 330px;  
    height: 210px;}
```

NB '*margin-left: 10px;*' zorgt voor ruimte tussen de afbeelding en eventuele tekst links van de afbeelding. Merk ook op dat aan de *img-tag* in bovenstaand voorbeeld twee klassen (*align-right* en *small*) worden toegekend.

- *background-image*. Middels de *background-image*-eigenschap kan een HTML-element van een achtergrondaafbeelding worden voorzien. Met *background-repeat* (*repeat-x*, *repeat-y* en *no-repeat*) en *background-attachment* (*fixed*, *scroll*) kunnen respectievelijk de herhaling (zowel horizontaal als verticaal) en de adhesie van de achtergrondaafbeelding worden gespecificeerd. Met *background-position* kan de positie van de achtergrond worden bepaald. Zie ook pagina's 413 tot en met 416 in het boek.

```
body {  
    background-image: url("oever.jpg");  
    background-repeat: repeat-x;  
    background-attachment: fixed;  
    background-position: center top;}
```

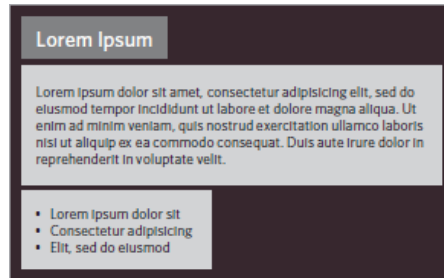
- *gradient*. Middels de *gradient*-eigenschap kan een HTML-element een kleurverloop als achtergrond krijgen. Deze eigenschappen wordt nog niet door alle *browsers* op dezelfde manier geïnterpreteerd. Zie pagina 419 in het boek.

## 2.1 Positie en *flow*

- *Boxes, block-level en inline elements:*

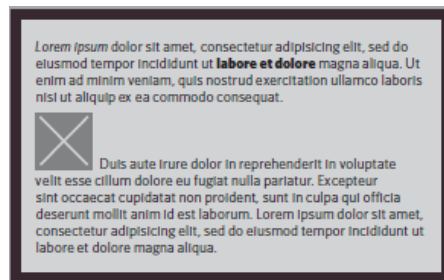
Ieder HTML-element wordt, als het ware, als een omkaderd *block*- of *inline* element beschouwd. Zo'n element lijkt op een doosje of een *box*, zoals in onderstaand voorbeeld. Zie ook de *hand-out* bij de vorige twee werkcolleges.

*Block-level elements:*



*Afbeelding afkomstig uit Ducket (2011: 361).*

*Inline element:*



*Afbeelding afkomstig uit Ducket (2011: 361).*

- *Parent elements:*

Wanneer een *block-level element* binnen een ander *block-level element* is opgenomen, wordt het buitenste element *parent* (ouder) genoemd. Deze relatie moet direct zijn. *Div*-elementen worden vaak als *parents* gebruikt, om webpaginaonderdelen te positioneren.

Verschillende  
*parent elements:*



*Afbeelding afkomstig uit Ducket (2011: 362).*

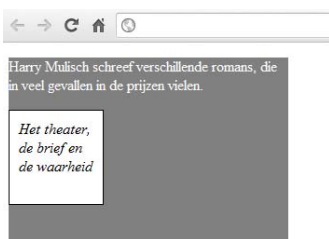
- *Flow:*

De uitvulling (*flow*) van een webpagina plaatst alle blokelementen onder elkaar, van boven naar beneden. *Inline* elementen worden naast elkaar geplaatst, van links naar rechts.

## Position:

De standaard-*flow* van een webpagina kan worden aangepast middels de *position*-eigenschap.

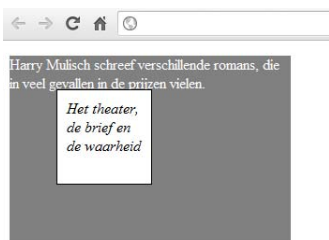
<b>position: static</b>	specificeert dat de standaard- <i>flow</i> geldt. (Zie pagina 365 in het boek.)
<b>position: relative</b>	bepaalt de positie vanaf de plaats binnen de (standaard-) <i>flow</i> . (Zie pagina 366 in het boek.)
<b>position: absolute</b>	bepaalt de positie ten opzichte van het <i>parent</i> -element en negeert hierbij standaard- <i>flow</i> en die van andere elementen binnen de <i>parent</i> . (Zie pagina 367 in het boek.)
<b>position: fixed</b>	bepaalt de positie ten opzichte van het <i>browser</i> -venster. (Zie pagina 368 in het boek.)



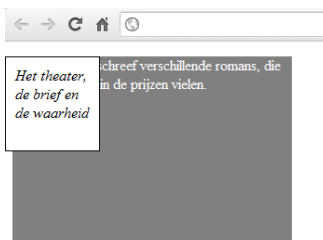
```
#parentElement {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: grey;  
  color: white;}  
p.example {  
  position: static;
```



```
#parentElement {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: grey;  
  color: white;}  
p.example {  
  position: relative;  
  top: 20px;  
  left: 50px;
```



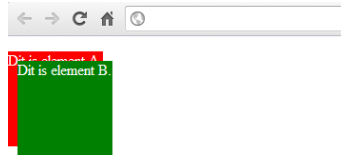
```
#parentElement {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: grey;  
  color: white;}  
p.example {  
  position: absolute;  
  top: 20px;  
  left: 50px;
```



```
#parentElement {  
  position: relative;  
  width: 300px;  
  height: 200px;  
  background-color: grey;  
  color: white;}  
p.example {  
  position: fixed;  
  top: 10px;  
  left: 0px;
```

– *z-index*:

Bij gebruik van *relative*, *fixed* en *absolute positioning*, kan het, zoals blijkt uit bovenstaande afbeeldingen, voorkomen dat elementen ‘over elkaar’ worden geplaatst. Om te bepalen welk element op de voorgrond komt, kan een zogenaamde *z-index* worden gebruikt. Hoe hoger de *z-index*-waarde, hoe meer een element ‘op de voorgrond treedt’. Zie het verschil in onderstaande situaties.



**HTML-code**

```
<p id="elementA">Dit is element A.</p>
<p id="elementB">Dit is element B.</p>
```

**CSS-code**

```
#elementA {
  position: absolute;
  top: 0px;
  left: 0px;
  width: 100px;
  height: 100px;
  background-color: red;
  color: white;}
#elementB {
  position: absolute;
  top: 10px;
  left: 10px;
  width: 100px;
  height: 100px;
  background-color: green;
  color: white;}
```



**HTML-code**

```
<p id="elementA">Dit is element A.</p>
<p id="elementB">Dit is element B.</p>
```

**CSS-code**

```
#elementA {
  position: absolute;
  top: 0px;
  left: 0px;
  width: 100px;
  height: 100px;
  background-color: red;
  color: white;
  z-index: 10;}
#elementB {
  position: absolute;
  top: 10px;
  left: 10px;
  width: 100px;
  height: 100px;
  background-color: green;
  color: white;
  z-index: 5;}
```

## 2. Floating

- float:

Met de *float*-eigenschap kunnen elementen binnen de *flow* horizontaal worden gepositioneerd. Andere elementen binnen het *parent*-element passen hun positie aan het element met de *float*-eigenschap aan. (De tekst in de afbeelding is afkomstig uit Dostoyevsky 1866/2000.)



**CSS-code**  
blockquote {  
width: 275px;  
font-family: georgia;  
font-size: 14pt;  
padding: 10px;  
background-color: #eee;  
border-top: 1px solid black;  
border-bottom: 1px solid black;}



**CSS-code**  
blockquote {  
width: 275px;  
font-family: georgia;  
font-size: 14pt;  
padding: 10px;  
background-color: #eee;  
border-top: 1px solid black;  
border-bottom: 1px solid black;  
float: right;}

Zie ook pagina's 370 tot en met 376 in het boek.

## 3. Afmetingen, *fixed* en *liquid lay-outs*

- Afmetingen: Verschillende apparaten kennen verschillende schermafmetingen. Zie pagina's 377 en 378 in het boek voor enkele actuele standaardafmetingen.
- Resolutie: De term *resolutie* wordt gebruikt voor het aantal pixels of beeldpunten dat een scherm breed en hoog is. Zo heeft een *iPhone-4*-scherm een resolutie van 640 bij 960 pixels, terwijl een *27 inch-iMac*-scherm een resolutie kent van 2560 bij 1440 pixels.
- Omgaan met afmetingen, *responsive webdesign*: Veel websites worden zowel op kleine schermen als met grote schermen bekeken. *Responsive webdesign* gaat ervanuit dat elk type apparaat (telefoon, *tablet*, PC etc.) een andere ervaring ondersteunt of overbrengt en dat het ontwerp van een website daarop moet reageren.
- *Fixed lay-out*: De grootte van de webpagina is niet afhankelijk van de resolutie van het scherm waarop de website wordt bekeken. Zo zal een *fixed lay-out* van 960 pixels breed horizontale *scrollbars* krijgen op een scherm van 800 bij 600 pixels groot.

*Voordelen:* Er kan precies, 'tot op de pixel', worden ontworpen.  
Er is grote controle over de positie van elementen.  
De regellengte kan worden bepaald.  
De afmetingen van afbeeldingen blijven altijd gelijk.



*Nadelen:* Een scherm wordt vaak niet optimaal benut, er is veel lege ruimte.  
 Als de tekstgrootte in de *browser* wordt aangepast, schaalt de rest van het ontwerp niet mee.  
 Het ontwerp werkt vaak het best op laptop- of PC-schermen.  
 Het ontwerp neemt vaak veel verticale ruimte in.

- *Liquid lay-out:* De grootte van de webpagina is afhankelijk van de resolutie van het scherm waarop de website wordt bekeken. Zo zal een *liquid lay-out* op een scherm van 800 bij 600 pixels smaller zijn dan op een scherm van 2560 bij 1440 pixels.

*Voordelen:* Het gehele scherm wordt benut.  
 Als het scherm van de gebruiker klein is, past het ontwerp zich hierop aan.

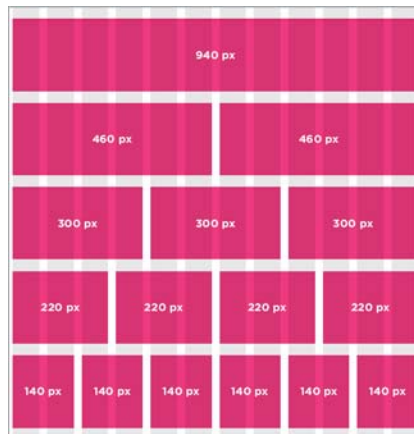
Als de tekstgrootte in de *browser* wordt aangepast, schaalt de rest van het ontwerp mee.

*Nadelen:* Het ontwerp kan er anders uitzien dan in eerste instantie werd bedoeld.  
 Bij hoge resoluties worden tekstregels vaak te lang om een prettige leeservaring te garanderen.

Zie ook pagina's 381 tot en met 386 in het boek.

## 31. Grids

- *Grid:* Een *grid* is een standaardorganisatie van beeldelementen. *Grids* worden in visueel ontwerp gebruikt als een richtlijn voor vlakverdeling. Ook bij ontwerpen voor het web kan een *grid* handig zijn. Het gebruik van een *grid* maakt het eenvoudiger om een evenwichtig ontwerp te maken.
- *960 pixel grid:* Een veelgebruikte *grid* is de *960 pixel grid*, waarin de te ontwerpen webpagina in twaalf (denkbeeldige) gelijke kolommen wordt verdeeld.



Afbeelding afkomstig uit Duckett (2011: 389).



Afbeelding afkomstig uit Duckett (2011: 388).

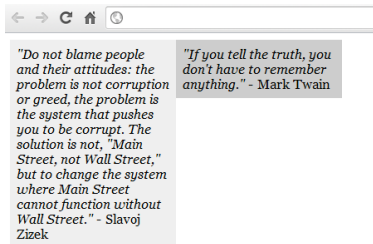
Zie ook pagina's 387 tot en met 394 en <http://www.960.gs>

### 32. Meerdere *stylesheets*

- Meerdere *stylesheets*: Meerdere *stylesheets* kunnen worden gebruikt voor de opmaak van een webpagina. Als twee *stylesheets* dezelfde eigenschap van een element specificeren, dan overschrijft, zoals op basis van de overervingsregels (zie *hand-out* bij werkcollege 4) valt te verwachten, een later opgeroepen *stylesheet* een daarvoor opgeroepen *stylesheet*.
- Link: Meerdere *stylesheets* kunnen worden toegepast door meerdere verwijzingen naar CSS-bestanden in de *head*-sectie van de HTML-pagina op te nemen.  
`<link href="lay-out.css" type="text/css" rel="stylesheet" />`  
`<link href="typography.css" type="text/css" rel="stylesheet" />`
- @import: Meerdere *stylesheets* kunnen worden toegepast door in een CSS-bestand een ander CSS-bestand te importeren.  
`@import url("forms.css");`  
`@import url("typography.css");`

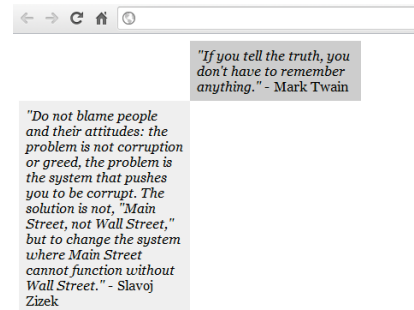
### 33. *Float* en *clear*

- *Float*: Met de *float*-eigenschap kunnen elementen binnen de *flow* horizontaal worden gepositioneerd. Andere elementen binnen het *parent*-element passen hun positie aan het element met de *float*-eigenschap aan.
- *Clear*: Met de *clear*-eigenschap kan worden gespecificeerd of een element andere elementen met de *float*-eigenschap naast zich duldt. De *clear*-waarden specificeren welke zijden *geen elementen* naast zich toelaten. Mogelijke waarden zijn *left*, *right* en *both*.



#### CSS-code (ingekort)

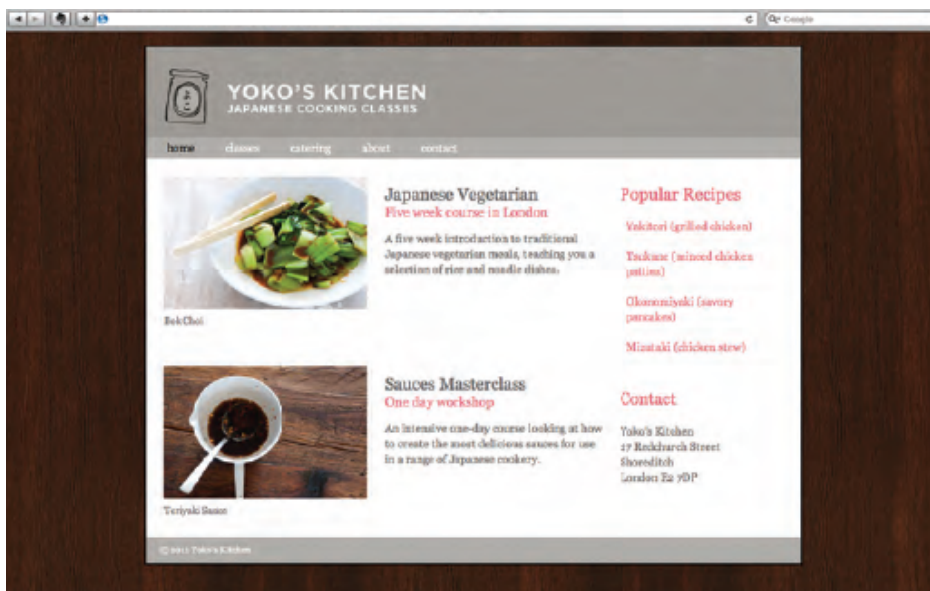
```
div.float {  
    width: 46%;  
    float: right;  
    background-color: #ccc;}  
  
div.clear {  
    width: 46%;  
    background-color: #eee;}
```



#### CSS-code (ingekort)

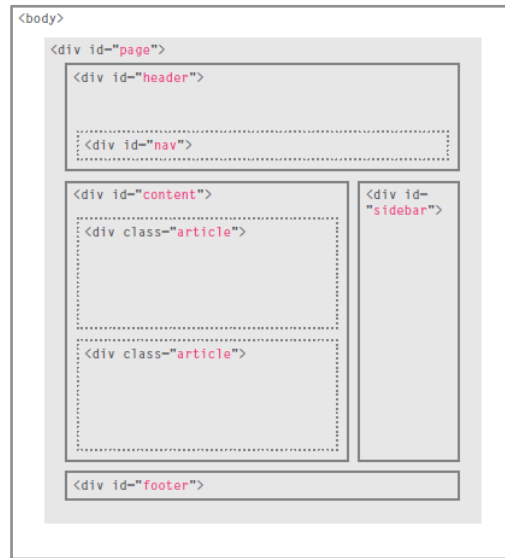
```
div.float {  
    width: 46%;  
    float: right;  
    background-color: #ccc;}  
  
div.clear {  
    width: 46%;  
    clear: right;  
    background-color: #eee;}
```

### 34. *Lay-out* in HTML-5



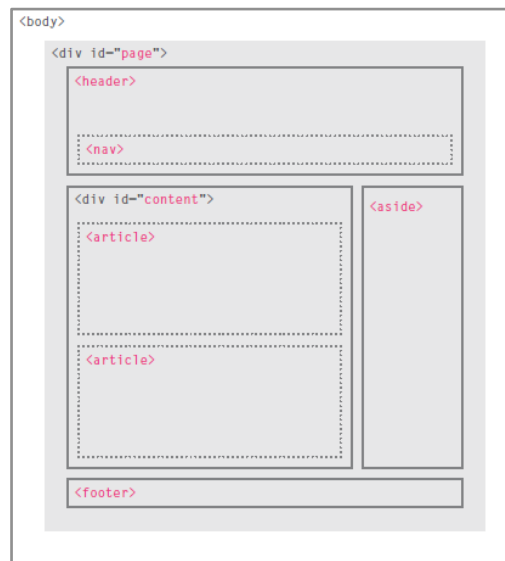
Afbeelding afkomstig uit Duckett (2011: 443).

- *Lay-out in div's*: De *lay-out* van een webpagina kan worden opgebouwd door middel van *div's*. Het nadeel van deze aanpak is dat de *div's* zelf niets zeggen over de webpagina – ze kennen geen semantische waarde.



Afbeelding afkomstig uit Duckett (2011: 431).

- *Lay-out elements*: De *lay-out* van een webpagina kan worden opgebouwd door middel van *lay-out elements*. Het voordeel ten opzichte van een *lay-out* in *div's* is dat de *lay-out*-elementen betekenis bijdragen aan de pagina.



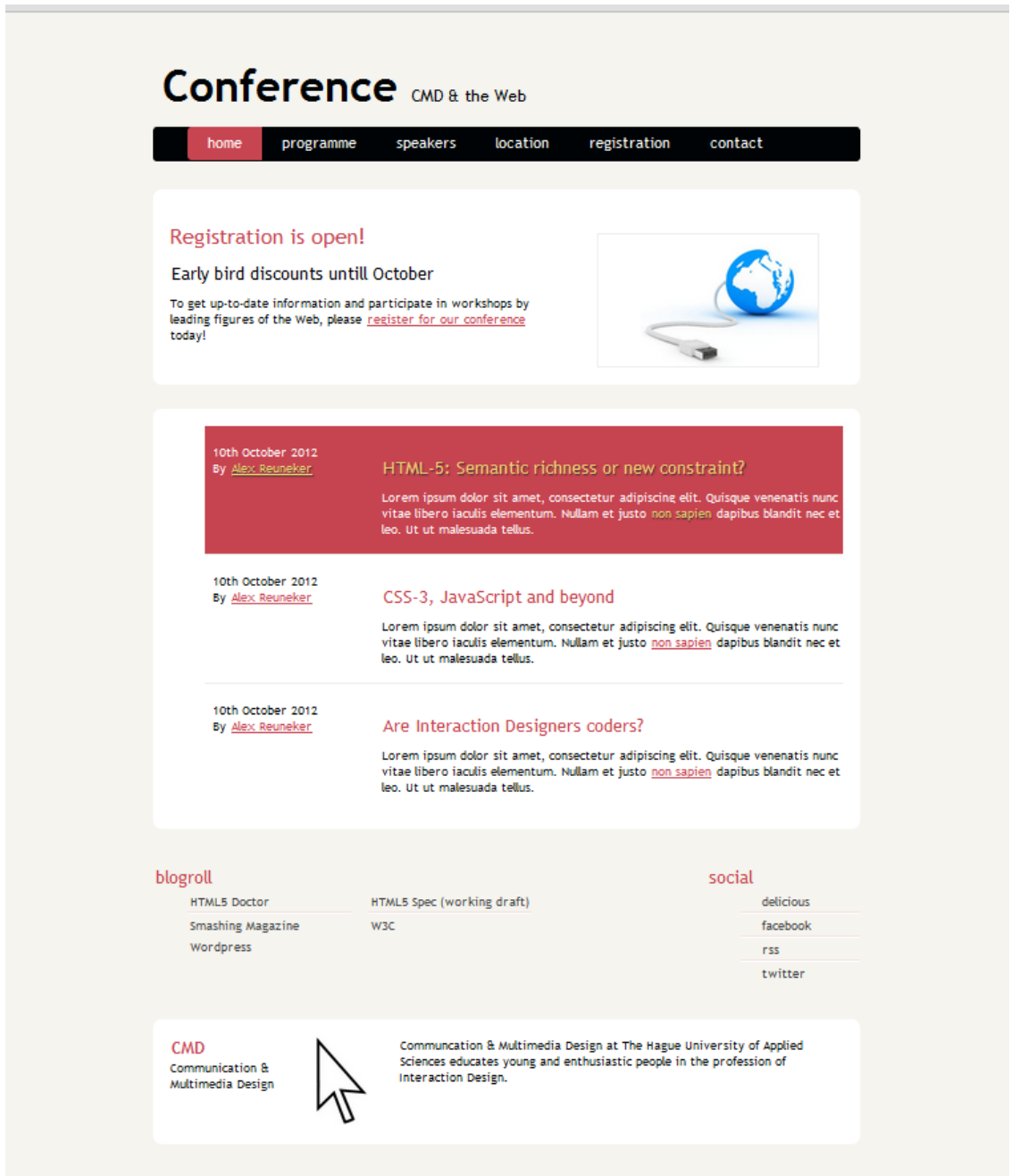
Afbeelding afkomstig uit Duckett (2011: 432).

### 35. **Lay-out**-elementen

- `<header> </header>`: Het *header*-element kan worden gebruikt voor *header* van een webpagina, maar ook voor de *header* van een *article* of *section*.
- `<footer> </footer>`: Het *footer*-element kan worden gebruikt voor *footer* van een webpagina, maar ook voor de *footer* van een *article* of *section*.
- `<nav> </nav>`: Het *nav*-element kan worden gebruikt voor hoofdnavigatie van een webpagina. Vaak bestaat de navigatie uit een *unordered list*-sectie.
- `<article> </article>`: Het *article*-element kan worden gebruikt voor een (tot op zekere hoogte) semantisch zelfstandig onderdeel van een webpagina. Denk hierbij aan een *blog-post*, een krantenartikel of omschrijving van een cursus. Een *article*-sectie kan ook andere *article*-secties bevatten.
- `<aside> </aside>`: Het *aside*-element kan worden gebruikt voor gerelateerde, maar niet essentiële informatie bij een *article* of een gehele webpagina. Denk hierbij aan contactinformatie of een lijst met gerelateerde links of boeken.
- `<section> </section>`: Het *section*-element kan worden gebruikt voor het groeperen van secties binnen een webpagina. Denk hierbij aan een *blog*-sectie, een nieuwssectie etc. Een *section* kan meerdere *articles* bevatten, maar een *article* kan ook *section*-elementen bevatten.
- `<hgroup> </hgroup>`: Het *hgroup*-element kan worden gebruikt voor het groeperen van koppen. Zo kunnen *h1*- en *h2*-elementen aan elkaar gekoppeld worden.
- `<figure> </figure>`: Binnen deze tags wordt een afbeelding opgenomen. De afbeelding kan zo worden voorzien van een ondertekening met de *figcaption*-tag.
- `<figcaption> </figcaption>`: Binnen deze tags wordt het ondertekening opgenomen. Deze tags worden *genest* in het *figure*-element. (Zie ook de *hand-out* bij werkcollege 2.)
- *Lay-out*-elementen *vs* *div*'s: *Lay-out*-elementen dragen betekenis bij aan de pagina en verdienen daarom de voorkeur boven *div*'s. Echter, niet voor alle doeleinden zijn *lay-out*-elementen voorhanden. In die gevallen worden *div*- of *span*-elementen gebruikt.

### 3. Benoem de *lay-out*-elementen

Omcirkel en benoem de juiste *lay-out*-elementen in onderstaande afbeelding.



Afbeelding gebaseerd op Coding An HTML 5 Layout From Scratch op Smashing Magazine

**Bronnen**

Brakman, Willem (1983). *De reis van de douanier naar Bentheim*. Amsterdam: Querido.

Coding An HTML 5 Layout From Scratch. (2009). *Smashing Magazine*. Opgevraagd via

<http://coding.smashingmagazine.com/2009/08/04/designing-a-html-5-layout-from-scratch/>

Dostoyevsky, Fyodor (1866/2000). *Crime and Punishment*. Hertfordshire: Wordsworth Editions Ltd.

Drs. P (1973). *De Veerpont. Zingt allen mee met Drs. P*. Polydor.

Duckett J. (2011) *HTML and CSS: Design and Build Websites*. Indianapolis: John Wiley and Sons Inc.