

Web technology

Les 1: HTML

Kristof Michiels

Lesinhoud

- HTML-intro: 3 basis-elementen (p, h1...6, ul en ol)
- Een pagina structureren met header, main, footer en nav
- Inline en block-elementen (em en strong)
- Generieke elementen (div en span)
- Document Object Model (DOM)
- Link elementen
- Afbeeldingen
- Commentaar
- Id en class

Drie content-elementen: paragrafen, hoofdingen en lijsten

Correct HTML schrijven

- Elk stukje informatie op een pagina moet in een html-element verpakt worden
- Anders spreken we over "naakte" of "anonieme" tekst. Je document is dan niet-correct opgesteld
- Je dient je html-element juist te kiezen, volgens de betekenis van de ingesloten informatie
- Goed voor je lezers maar ook voor zoekmachines of [schermlezers](#)
- Hoe het er uit zal zien is hierbij niet van belang. De vormgeving is het domein van CSS

Paragrafen: <p>...</p>

- Het paragraaf-element is zowat de belangrijkste bouwsteen van tekstdocument
- Het begint met de <p>-tag en eindigt met de </p>-tag
- Het kan behalve tekst ook andere html-elementen bevatten (bvb. afbeeldingen)
- Hoofdingen, lijsten (we zien later dat dit block-elementen zijn) passen dan weer niet binnen een paragraaf

<p>

Lorem Ipsum is simply dummy text of the printing and typesetting industry.
 Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
 when an unknown printer took a galley of type and scrambled it to make a
 type specimen book.

</p>

Hoofdingen: <h1>, <h2>, ... , <h6>

- Met hoofdingen geef je titels en subtitels weer
- bvb. het h1-element begint met de <h1>-tag en eindigt met de </h1>-tag
- HTML geeft ons 6 niveau's voor hoofdingen
- <h1> voor de belangrijkste hoofding. Van daaruit werk je verder met <h2>, <h3>...

```
<h1>Je belangrijkste titel</h1>  
<h2>Een net iets minder belangrijke titel</h2>  
<h3>Een ondertitel van het derde niveau</h3>  
<h4>Een ondertitel van het vierde niveau</h4>  
<h5>Een ondertitel van het vijfde niveau</h5>  
<h6>Een ondertitel van het zesde niveau</h6>
```

Hoofdingen: <h1>, <h2>, ... , <h6>

- Regel: op elke pagina mag slechts één <h1>-element
- De overige h-elementen: zo vaak als nodig
- Je hoeft ze niet alle zes te gebruiken. Ook niet van <h1> tot <h6> voor elke volgende titel
- De structuur die je aanbrengt is zoals bij een boek: één titel (<h1>), hoofdstukken (allen <h2>), onderdelen van hoofdstukken (allen <h3>) enz...

```
<h1>Mijn blog</h1>  
<h2>Mijn eerste post</h2>  
<p>Blogtekst komt hier.</p>  
<h2>Mijn tweede post</h2>  
<p>Blogtekst komt hier.</p>
```

Lijsten: en

- Worden heel vaak gebruikt, veel méér dan voor "lijstjes"
- Links, afbeeldingen, tekstuele opsommingen: alles wat meer dan één keer voorkomt kan je in lijst vatten
- HTML geeft ons verschillende manieren om lijsten te maken: we zien er vandaag twee

Niet-geordende lijsten: ...

- Voor een opsomming zonder volgorde gebruiken we het ul-element
- Het element begint met de -tag en eindigt met de -tag
- Een lijst is binnenin opgebouwd uit lijstelementen. Deze worden geplaatst binnen li-tags (zie voorbeeld)
- De browser plaatst standaard een *bullet* voor elk list item (aanpasbaar met CSS)

```
<ul>  
  <li>Times New Roman</li>  
  <li>Courier</li>  
  <li>Comic Sans</li>  
</ul>
```

Niet-geordende lijsten: ...

- Binnen de lijstelementen kan je vrij andere elementen gebruiken (zie het voorbeeld)
- Merk op hoe we inspringen om alles leesbaar te houden. Insprongen maak je met tabs of spaties
- Html-elementen die in elkaar genest zitten: we spreken over *ouders* en *kinderen*

```
<ul>
  <li>
    <h3>Een titel</h3>
    <p>Quisque rhoncus euismod pulvinar. Nulla non arcu at lectus.</p>
  </li>
  <li>
    <h3>Een andere titel</h3>
    <p>Quisque rhoncus euismod pulvinar. Nulla non arcu at lectus.</p>
  </li>
</ul>
```

Geordende lijsten: ...

- Gebruik je wanneer de volgorde wél belangrijk is
- Het element begint met de -tag en eindigt met de -tag
- Voor de rest werken ze op gelijkaardige manier als niet-geordende lijsten
- Ook deze lijst is binnenin opgebouwd uit lijstelementen
- De browser kent standaard een nummering toe (aanpasbaar met CSS)

```
<ol>  
  <li>Helvetica</li>  
  <li>Garamond</li>  
  <li>Times New Roman</li>  
</ol>
```

Geordende lijsten: ...

- Met het start-attribuut kan je de nummering aanpassen
- Attributen komen vaak voor en hebben steeds dezelfde vorm: naam + "=" en daarna tussen aanhalingstekens de waarde
- Bij het schrijven van een attribuut gebruik je best geen spaties
- Hier mag je het attribuut weglaten, sommige elementen (bvb het img-element) hebben verplichte attributen

```
<ol start="5">  
  <li>Helvetica</li>  
  <li>Garamond</li>  
  <li>Times New Roman</li>  
</ol>
```

De list-style-type eigenschap

- Hiermee lopen we even vooruit: CSS geeft je controle over de manier waarop lijsten worden voorgesteld
- De *bullet* bij niet-geordende lijsten kan worden weggelaten, een vierkant of open cirkel worden, ...
- Bij geordende lijsten kan gekozen worden voor Romeinse cijfers, letters, ...

```
ul {  
    list-style-type: disc;  
}  
  
ol {  
    list-style-type: lower-roman;  
}
```

Lijst-elementen in elkaar vlechten

- We kunnen lijsten in elkaar vervlechten. De browser zal automatisch inspringen en de bullets aanpassen
- Je ziet: html schrijven is je hoofd erbij houden. Denk aan de insprongen

```
<ol>
  <li>Helvetica</li>
  <li>
    <ul>
      <li>Helvetica Neue</li>
      <li>Akzidenz Grotesk</li>
      <li>Neue Haas Grotesk</li>
    </ul>
  </li>
</ol>
```

Informatie organiseren: header, main, footer en nav

Informatie organiseren

- Elementen als `<h1>`, `<p>`, `` slaan op specifieke en kleinere stukjes informatie
- Er zijn ook elementen die betekenis geven aan een bepaald deel van een webpagina
- Het zijn stuk voor stuk verzamel-elementen die andere HTML-elementen gaan bevatten
- Hier de juiste keuzes maken is een skill die je langzaam onder de knie zal krijgen

Het main-element: <main>...</main>

- Hiermee identificeer je het belangrijkste deel van je webpagina
- Je trekt hiermee een cirkel rond de belangrijkste en unieke informatie op de pagina
- M.a.w. headers en footers, die we in de volgende slides zullen zien horen hier niet bij: zij bevatten informatie die terugkomt op andere pagina's

```
<main>
  <h1>Mat Mania or Exciting Hour</h1>
  <h2>Introduction</h2>
  <p>Is a 1985 Japanese <em>pro wrestling-themed</em> arcade game developed by
  Technōs Japan and published by Taito. It is a spiritual successor to the 1983
  arcade game Tag-Team Wrestling, also developed by Technōs Japan, but published
  by Data East. It was also ported to the PS4 console in 2015.</p>
</main>
```

Het header-element: <header>...</header>

- Voor een verzameling elementen bovenaan webpagina's, een sectie of artikel (zie verder)
- Geen vaste ingrediënten voor wat een header hoort te bevatten: jij kiest
- Bevat vaak een h1-element, een logo-afbeelding, een nav-element met links (komen verder nog aan bod)

```
<header>
  
  <h1>Taito Japan</h1>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Over ons</a></li>
    </ul>
  </nav>
</header>
```

Het footer-element: <footer>...</footer>

- Gebruikt voor verzameling elementen onderaan een pagina, sectie of artikel
- Geen vaste ingrediënten voor wat een footer hoort te bevatten: jij kiest
- Bevat vaak auteursinformatie, copyright-informatie, verwante documenten of pagina's, sitenavigatie ...

```
<footer>
  <p>Deze site werd gemaakt door Kristof Michiels te Antwerpen in 2021</p>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Over ons</a></li>
    </ul>
  </nav>
</footer>
```

Het nav-element: <nav>...</nav>

- Gebruikt om de belangrijkste navigatie-gedeeltes te benoemen
- Denk aan de navigatiebalk op de sites die je bezoekt: links naar andere pagina's op de site
- Je gaat deze links omwille van het lijst-aspect vaak in een ul-element plaatsen

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Over ons</a></li>
  </ul>
</nav>
```

Inline en block-elementen

Block-elementen

- Alle HTML-elementen zijn ofwel van het type block of inline
- Alle elementen die we tot hiertoe hebben gezien zijn van het type block
- Dit betekent dat ze beginnen na en eindigen voor een eigen witte regel
- Truukje: ze gebruiken "ellebogen" om de omliggende elementen naar een nieuwe lijn te duwen

Inline-elementen

- Inline elementen hebben die kracht niet!
- Ze zitten ook zo goed als steeds binnen een ander element
- Ze worden met name gebruikt om bepaalde accenten te leggen binnen een tekst
- Met CSS zullen we het "karakter" van een element kunnen aanpassen: van block naar inline of omgekeerd
- We gaan nu enkele inline elementen overlopen

Tekst benadrukken met em: `...`

- em staat voor emphasized
- We benadrukken hiermee één of meerdere woorden in de tekst
- We willen hiermee hun *belang in de zin* onderstrepen
- De browser drukt dit standaard schuin gedrukt af, maar we zullen leren aan te passen vanuit CSS

```
<em>Benadrukte tekst</em>
```

```
<p>Grotere context met <em>Benadrukte tekst</em></p>
```


Tekst benadrukken met strong: ...

- Ook met strong benadrukken we één of meerdere woorden in de tekst
- Verschil met em is subtiel. De woorden in strong springen als het ware uit de tekst, je kent aan hen een speciaal belang toe in de grotere context van de pagina
- De browser drukt dit standaard vet gedrukt af, maar we zullen leren aan te passen vanuit CSS

```
<strong>Benadrukte tekst</strong>
```

```
<p>Grotere context met <strong>Benadrukte tekst</strong></p>
```

Generieke elementen

Generieke HTML-elementen: <div> en

- Generiek = merkloos (bvb bij generische medicijnen)
- We zagen eerder elementen als : main, header, footer, nav
- Dit zijn verzamelementen, met een extra betekenis: bvb "hoofding" voor header
- Wat nu als geen enkel van deze genoemde elementen past bij een verzameling die je wil maken?
- Dan gebruik je div en span :-)

Het div-element: <div>...</div>

- Div staat voor *division*
- Het is een block-element
- Te gebruiken als een verzamel-element als er geen beter passend element bestaat

```
<div>
  
  <h1>Taito Japan</h1>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Over ons</a></li>
    </ul>
  </nav>
</div>
```

Het span-element: ...

- Span staat voor span (=omspannen)
- Het is een inline element
- Je gebruikt het bvb. binnen een tekst om een extra accent te leggen met CSS/JS

```
<p>Mijn broer heeft <span>blauwe</span> ogen.</p>
```

De DOM: Document Object Model

Nog even herhalen

- We schrijven al onze html binnen het body-element in onderstaande basisstructuur
- HTML-elementen zitten verweven in elkaar
- Wij geven dit voor onszelf weer door met tabinsprongen te werken

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Titel komt hier</title>
  </head>
  <body>
    Alle informatie op de pagina komt hier
  </body>
</html>
```

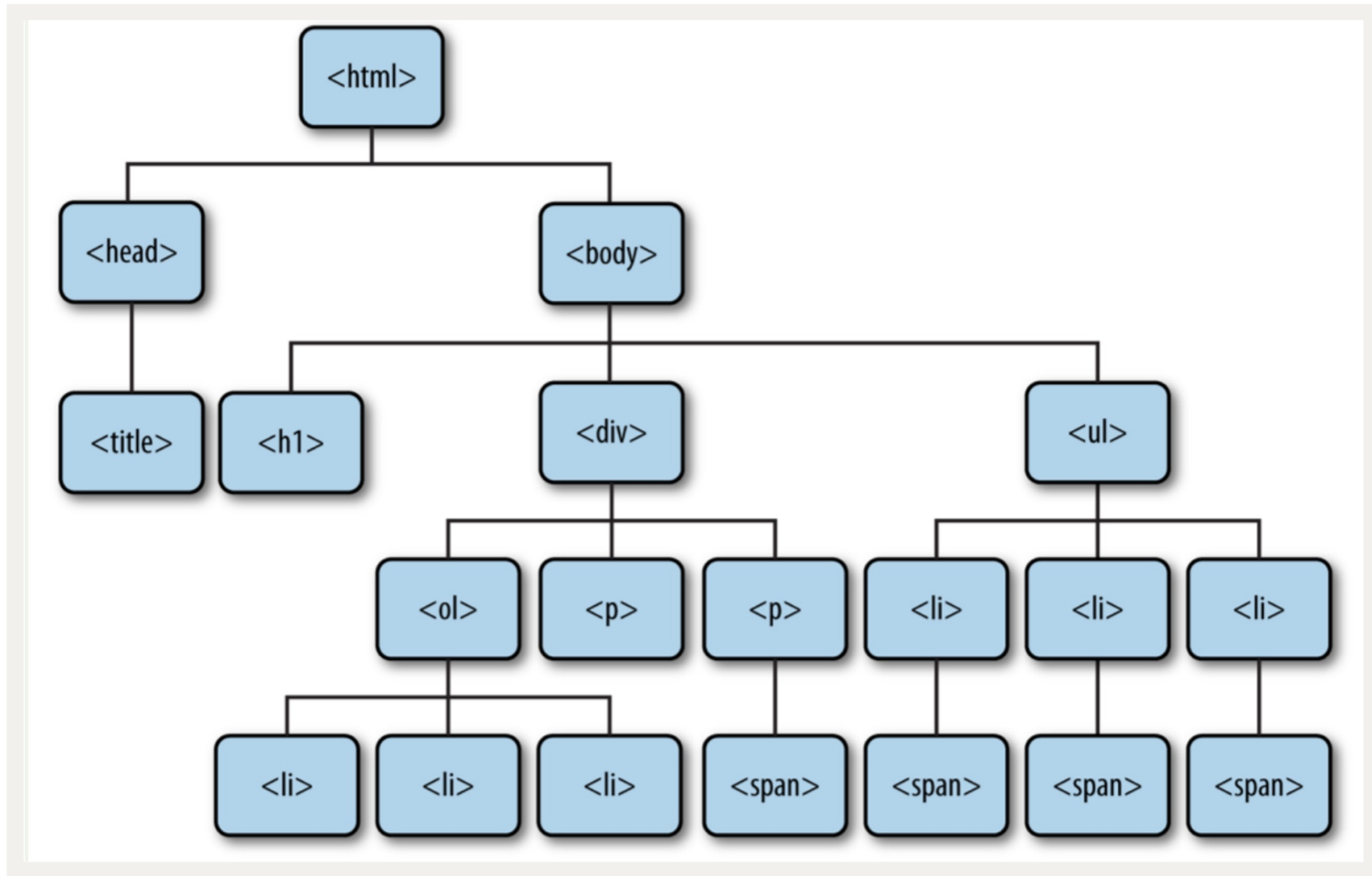
De DOM

- HTML tags definiëren een hiërarchische structuur die het Document Object Model wordt genoemd, kortweg de DOM
- Met de DOM kan geïnterageerd worden met een scripting taal als JavaScript of met CSS
- HTML elementen definiëren DOM-elementen. Dit zijn entiteiten die leven binnen de DOM
- vertaalt zich in een boomdiagram van de DOM... We spreken over:
 - afstammelingen (descendants)
 - kind-ouderrelaties (parents-children)

DOM

```
<body>
  <h1>Hallo, Wereld!</h1>
  <div>
    <ol>
      <li>List Item</li>
      <li>List Item</li>
      <li>List Item</li>
    </ol>
    <p>Dit is een paragraaf.</p>
    <p>En dit een <span>tweede</span>.</p>
  </div>
  <ul>
    <li>List Item <span>1</span></li>
    <li>List Item <span>2</span></li>
    <li>List Item <span>3</span></li>
  </ul>
</body>
```

DOM



The DOM als boomstructuur

Link elementen

Link elementen: <a>...

- "Anchor" of anker-elementen
- Hypertext links
- Zijn de essentie van het web
- 1 verplicht attribuut: href

Link elementen: <a>...

Een voorbeeld:

- het href-attribuut wijst naar de site die moet worden geopend als op de link wordt geklikt
- Tussen <a> en schrijf je de content die aanklikbaar moet worden

```
<a href="http://google.com">Ga naar de Google site</a>
```

Link elementen: <a>...

- Om van een afbeelding een link te maken zet je de afbeelding tussen <a> en
- Op deze manier kan je van elk HTML content element (of groep elementen) een link maken

```
<a href="http://google.com"></a>
```

Link elementen: <a>...

Een voorbeeld:

```
<ul>
  <li>
    <a href="http://google.com">
      <h2>De Google website</h2>
      <p>Op de Google website zoek je met behulp van zoektermen.</p>
    </a>
  </li>
  <li>...</li>
  <li>...</li>
</ul>
```

Link elementen: <a>...

- Browser geeft standaard weer in het blauw en onderlijnd
- Reeds bezochte links: paarse kleur
- Behoort tot vormgeving die wij gaan bepalen in CSS

Het href attribuut

```
<a href="url">Gelinkte content</a>
```

- URL staat voor *Uniform Resource Locator*
- Aangeklikte url wordt ingeladen in hetzelfde browservenster
- We kunnen linken naar absolute URLs of naar relatieve URLs

Absolute URLs

- Bevatten de volledige URL voor het document (inclusief http:// of https://), de domeinnaam en de padnaam (indien nodig)
- Je gebruikt een absolute URL indien je linkt naar een document buiten de eigen website
- Kan soms heel lang en intimiderend zijn
- Laat je hierdoor niet van de wijs brengen: het is slechts één attribuut met een ingevulde waarde

```
href="https://www.ap.be/opleiding/crossmedia-ontwerp"
```

Relatieve URLs

- Beschrijven een padnaam relatief ten opzichte van het huidige document
- Om te linken naar documenten op je eigen site (op dezelfde webserver)
- Geen http of https nodig

```
href="contact/index.html"
```

Relatieve URLs: linken binnen dezelfde folder

- Mag je doen door gewoon naar de bestandsnaam te wijzen

```
<a href="contact.html">Contacteer ons</a>
```

```
<a href="over.html">Over deze site</a>
```

Relatieve URLs: linken naar dieper gelegen folder

- Doe je door te verwijzen naar de folder gevolgd door een slash
- Indien meerdere niveau's diep, dit herhalen

```
<a href="recepten/spaghetti.html">Spaghetti maken</a>
```

```
<a href="recepten/pastas/spaghetti.html">Spaghetti maken</a>
```

Relatieve URLs: linken naar een hoger gelegen folder

- Hiervoor gebruik je de dot-dot-slash notatie ("../")
- Hiermee zeg je: ga één niveau hoger dan de huidige folder
- ../ staat dan voor de hoger gelegen folder en deze moet niet bij naam worden genoemd

```
<p><a href="../index.html">Terug naar de homepagina</a></p>
```

Relatieve URLs: linken naar een hoger gelegen folder

Verwijzen naar 2 niveau's hoger?

```
<p><a href="../../index.html">Terug naar de homepagina</a></p>
```

Linken met *Site Root*-relatieve padnamen

- Starten in de root folder en van daaruit alle subfolders opsommen
- Begint met een slash ("/") die de rootfolder symboliseert
- Je hoeft die rootfolder dus niet mee te benoemen, enkel de eventuele subfolders
- Werken enkel indien je bestanden op een webserver staan

```
<a href="/recepten/spaghetti.html">Spaghetti maken</a>
```

```
<a href="/recepten/pastas/spaghetti.html">Spaghetti maken</a>
```


Padnamen voor afbeeldingen

- Het src-attribuut werkt op net dezelfde manier als het href-attribuut
- Met het verschil dat deze laatste naar afbeeldingen verwijzen

```
  

```

Linken naar een specifieke plek op de pagina

- Je kan ook linken binnen een pagina
- Vaak gebruikt om shortcuts te voorzien naar specifieke plaatsen op lange informatierijke pagina's
- Twee stappen:
 - eerst ga je de bestemming identificeren (van een id voorzien)
 - daarna ga je een link aanmaken naar deze bestemming

```
<a href="#mijnbestemming">Link naar ondergelegen gedeelte</a>  
...  
<h2 id="mijnbestemming">  
<p>...</p>
```

Linken naar een specifieke plek op de pagina

- Je kan deze techniek ook gebruiken om te linken van beneden naar bovenaan de webpagina
- Je kan ook linken naar een specifieke plek op een andere pagina

```
<header id="hoofding">...</header>  
...  
<footer><a href="#hoofding">Terug naar boven</a></footer>
```

```
<a href="/recepten/spaghetti.html#bereidingstijd">Op andere pagina</a>
```

Openen in een nieuw browservenster

- Browser zal nieuw venster/tabblad openen
- Zeer spaarzaam gebruiken. Vaak geen goed idee op vlak gebruiksvriendelijkheid

```
<a href="https://www.ap.be" target="_blank">Artesis Plantijn Hogeschool</a>
```

Mail-links

- Browser zal mail-client openen en een nieuwe mail aanmaken naar mailto-adres
- Enkel indien browser geconfigureerd, werkt dus niet 100% van de tijd

```
<a href="mailto:kristof.michiels01@ap.be">Contacteer Kristof Michiels</a>
```

Afbeeldingen

Het img element

- Afbeeldingen plaats je met een img element
- Is zoals reeds gezegd een inline element
- Een img heeft geen afsluittag
- Heeft wel 2 verplichte attributen: src en alt

```
<p>Deze avond heb ik zin in een lekkere .
Heerlijk!</p>
```

Het img element

- Telkens de browser een img element tegenkomt, zal het dit element proberen te vertalen naar een afbeelding
- src en alt zijn verplicht
- src bevat de locatie van de afbeelding
- alt voorziet in een tekstuele beschrijving van de afbeelding

Het img element: src

- De onderkant van de afbeelding zal zich uitlijnen op de regel waarin de afbeelding is geplaatst
- Linken naar bestanden op een andere server (hotlinken): af te raden
- Downloaden (indien je rechten respecteert) en vanuit je eigen folders gebruiken krijgt onze voorkeur

Het img element: alt

- Het alt attribuut: is dus verplicht en belangrijk voor o.m. zoekmachines en schermlezers
- Zeker invullen wanneer:
 - een afbeelding deel uitmaakt van de content op de site
 - wanneer je een afbeelding gebruikt binnen een a-element
- Als een afbeelding niet bijdraagt tot de content op de pagina (meer eerder de vormgeving ondersteunt) dan mag je alt openlaten (alt="")

Het img element: breedte en hoogte

- Je kan afmetingen meegeven met afbeeldingen met de attributen width en height
- We zullen dit met CSS kunnen perfectioneren
- De getallen drukken de waarde uit in pixel (je mag ook px toevoegen aan getal)
- Hoogte én breedte meegeven kan een afbeelding vervormen
- Ga voor de exacte afmetingen, of volg de juiste verhouding, of geef slechts één waarde mee (ofwel width, ofwel height)

```
width="getal"  
height="getal"
```

```

```

Het img element

- Is dus inline
- Je kan het in een block-element stoppen
- Je zal het ook block kunnen maken met CSS (display: block;)

```
<p></p>
```

Commentaar

Commentaar toevoegen

- Handige manier om HTML te annoteren
- Steeds nuttig om de meer complexe stukken opmaak te voorzien van commentaar
- Goed voor anderen die de code moeten interpreteren, of voor jezelf later zodat men/je begrijpt waarom een bepaalde keuze werd gemaakt

```
<h1>Hallo, Wereld!</h1>  
<!-- Onderstaande tekst wordt nog aangepast -->  
<p>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua. Ut enim ad minim  
veniam, quis nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo consequat.</p>
```

id en class

Identificatie en classificatie met id en class

- We kwamen eerder al attributen tegen (weet je nog welke?)
- id en class zijn twee attributen waarmee je je elementen van een naam voorziet
- Die naam kan dan gebruikt worden vanuit CSS/JavaScript om iets met dat/die element(en) te doen
- Wordt vaak geassocieerd met div en span, maar je kan een id of class toevoegen aan alle html-elementen
- Het zijn wat we noemen globale attributen

Identificatie met id

- Gebruik id om te identificeren
- Een id is uniek: de waarde "hoofding" mag op deze pagina slechts één keer gebruikt worden
- Een id moet bestaan uit minstens 1 karakter. Gebruik als eerste karakter steed een letter
- Mag "a", "x", ... zijn. Mijn advies: probeer een betekenisvolle naam te geven

```
<div id="hoofding">  
  <h2>Mijn hoofding</h2>  
  <p>Mijn paragraaf</p>  
</div>
```

Classificatie met class

- Gebruik class om te classificeren
- De class-naam mag gedeeld zijn door meerdere elementen
- Door een klassenaam te gebruiken kunnen je CSS-stijlen toepassen op alle elementen die dezelfde klassenaam delen
- Elementen kunnen zowel een klassenaam en een id-naam hebben

```
<p class="beschrijving">Mijn paragraaf</p>  
<p class="beschrijving">Mijn paragraaf</p>
```