

Interface Design Lab 1 - Web

Introductie tot HTML

Elementen voor opmaak en structuur

Kristof Michiels

In deze presentatie

- Structuur en betekenis van een HTML-document
- Welke elementen voorziet HTML voor ons?
- Hoe maak je een keuze uit deze elementen?
- De aanloop naar onze eerste oefeningen

Goed geschreven HTML zorgt voor structuur en betekenis in je documenten

Enkele termen verklaard

- Content: alle vormen van informatie in je documenten (teksten, afbeeldingen, links,...)
- Structuur: de manier waarop iets in elkaar zit, waarop elementen van een verzameling samenhangen
- Semantiek of betekenis: HTML-element zorgen voor semantiek/betekenis in een document
- Opmaken: structuur en betekenis geven met behulp van HTML

Correct HTML schrijven

- Elk stukje informatie moet in een html-element verpakt worden
- Anders spreken we over "naakte" of "anonieme" tekst. Je document is dan niet-correct opgemaakt
- 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

Drie content-elementen: paragrafen, hoofdingen en lijsten

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) kunnen evenwel niet

```
<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

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 2018</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 elementen (en wat zijn block-elementen?)

Block vs Inline-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: het zijn de elementen zijn met ellebogen. Ze gebruiken ze om de omliggende elementen naar een nieuwe lijn te duwen

Block vs 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
- 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>
```

Het img-element:

- Hiermee plaats je afbeeldingen
- Is een element zonder afsluittag
- 2 verplichte attributen: src en alt
- Met source geef je aan welke afbeelding je wil tonen (aanwezig in de folder waar ook je html-bestand staat)
- Bij alt geef je een woordelijke beschrijving van de foto

```

```

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>
```


id en class

Identificatie en classificatie met id en class

- We kwamen eerder een attribuut tegen (weet je nog dewelke?)
- 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>
```

HTML schrijven

- Geen exacte wetenschap
- HTML5 geeft je een set van elementen
- Er zijn vaak meerdere manieren om een document te beschrijven
- Een semantisch correct opgemaakt document kan op zich heel mooi zijn ;-)

In de oefeningensessies deze week:

- Onze tools leren gebruiken
- De algemene html leren gebruiken
- Leren foutenvrije html schrijven
- Teksten van de juiste elementen voorzien