Web technology

Week 1: Inleiding en HTML

Kristof Michiels

In deze les komen aan bod…

- Inleiding
  - Aanpak
  - Afspraken
  - Een beetje geschiedenis
  - Onze werkinstrumenten
- HTML

Inleiding

Aanpak

- Leerstof via theorieles op maandag
- Deze leerstof instuderen en telkens inoefenen in het labo
- Labo's vinden plaats in kleinere klasgroepen. Kom voorbereid, werk actief mee, stel vragen
- Docenten Kristof Michiels, David Verhulst, Jeroen De Vos
- Belangrijke links:
  - ECTS-fiche
  - Digitap
  - Webuntis

Wat verwachten we van jou?

- We vertrekken van 0, maar gaan snel
- Doorzettingsvermogen om te oefenen tot je het kent
- Je werkinstrumenten goed beheersen
- Je computer in goede staat houden, op tijd updaten
- Netjes en nauwkeurig werken (computers kunnen niet goed overweg met typfouten): concentratie!
- Aanwezig in de lessen en labo's / leerstof instuderen / oefeningen maken

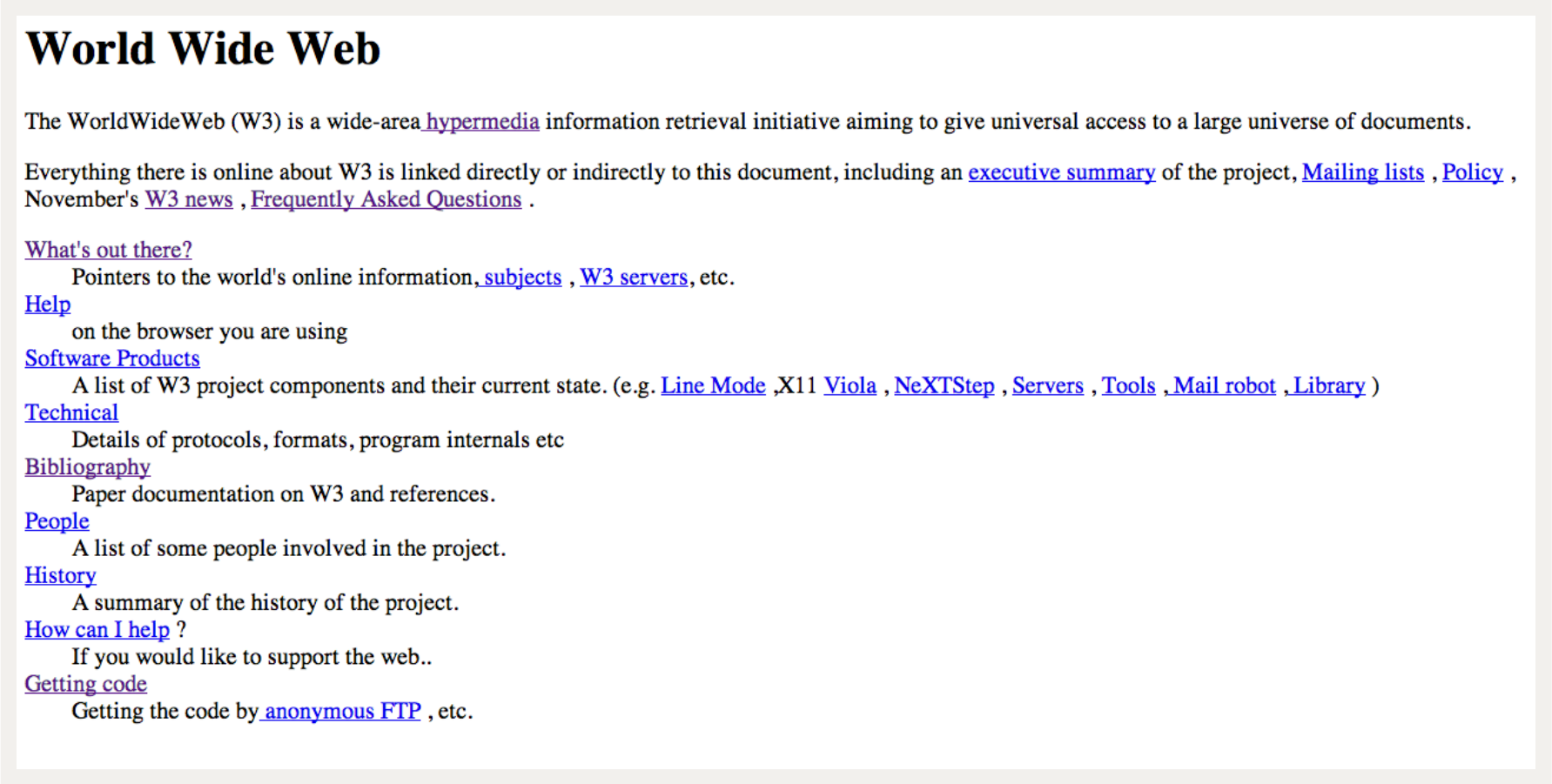
Web technology in perspectief

- "Ontwrichtende" technologie
- Ondertussen bijna 30 jaar oud met hele geschiedenis: boom, bust, wedergeboorte...
- We zijn de "klassieke website" ondertussen lang voorbij
- Core technologieën blijven dezelfde: HTML, CSS, JS
- Past zich aan aan de voortdurende technologische veranderingen (denk bvb AR, IOT, ...).
- Jullie zullen in de toekomst programmeren voor een "web" dat nog volop transformeert (en we ons vandaag nog niet kunnen voorstellen)

Tim Berners-Lee

De "vader" van het web

De allereerste website



World Wide Web

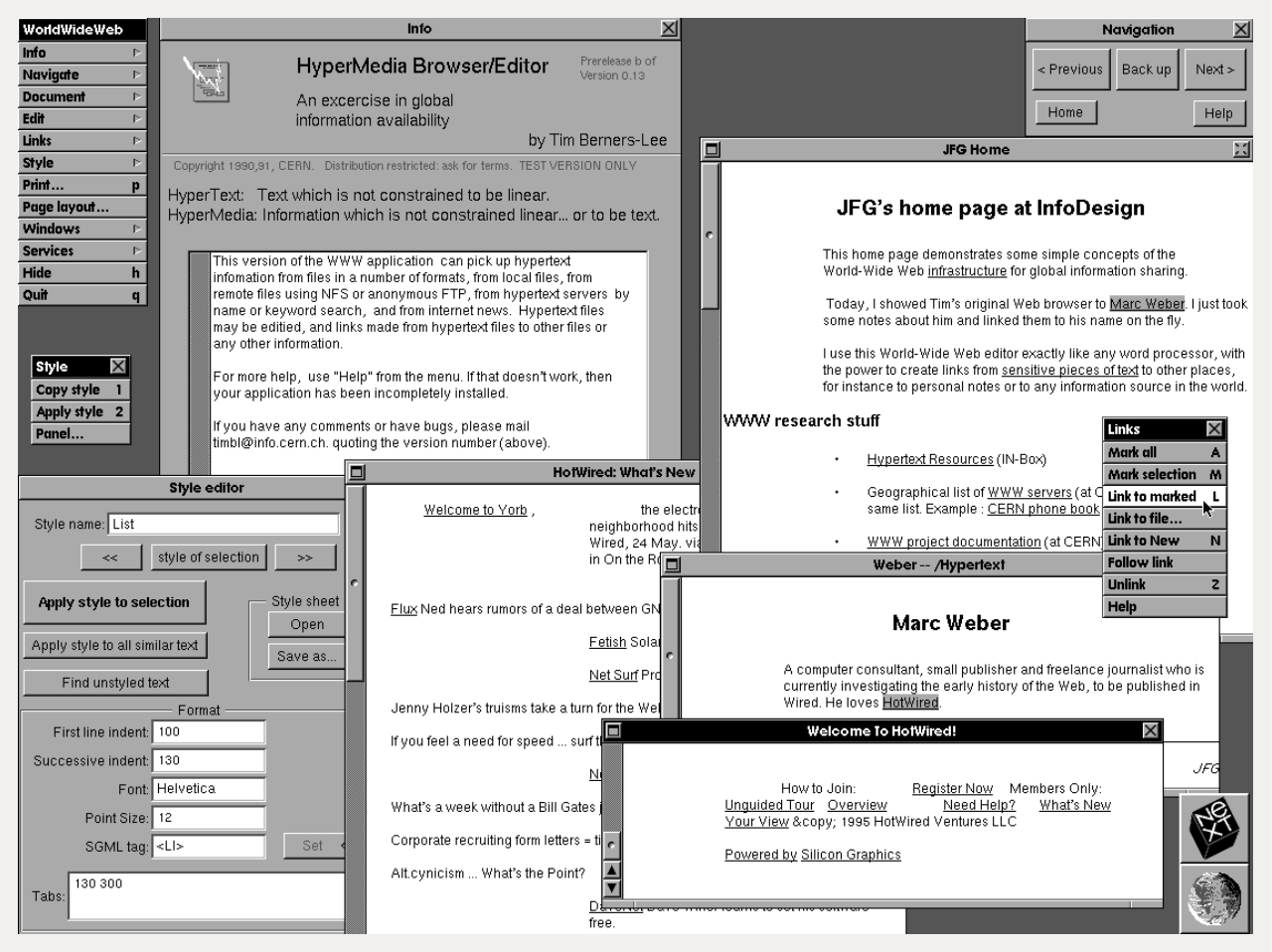
The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

http://bit.ly/over-de-eerste-website

De eerste browser



Tim Berners-Lee's desktop

3 bouwstenen van het web: HTML, CSS en JavaScript

Bouwsteen 1: HyperText Markup Language (HTML)

- De opmaaktaal voor het web
- Vandaag versie HTML versie 5
- Levende standaard: blijft voortdurend evolueren
- Beschrijft en structureert de informatie op een webpagina
- Eenvoudig om te leren en te schrijven
- Maar goed schrijven: is vaardigheid die komt met oefenen

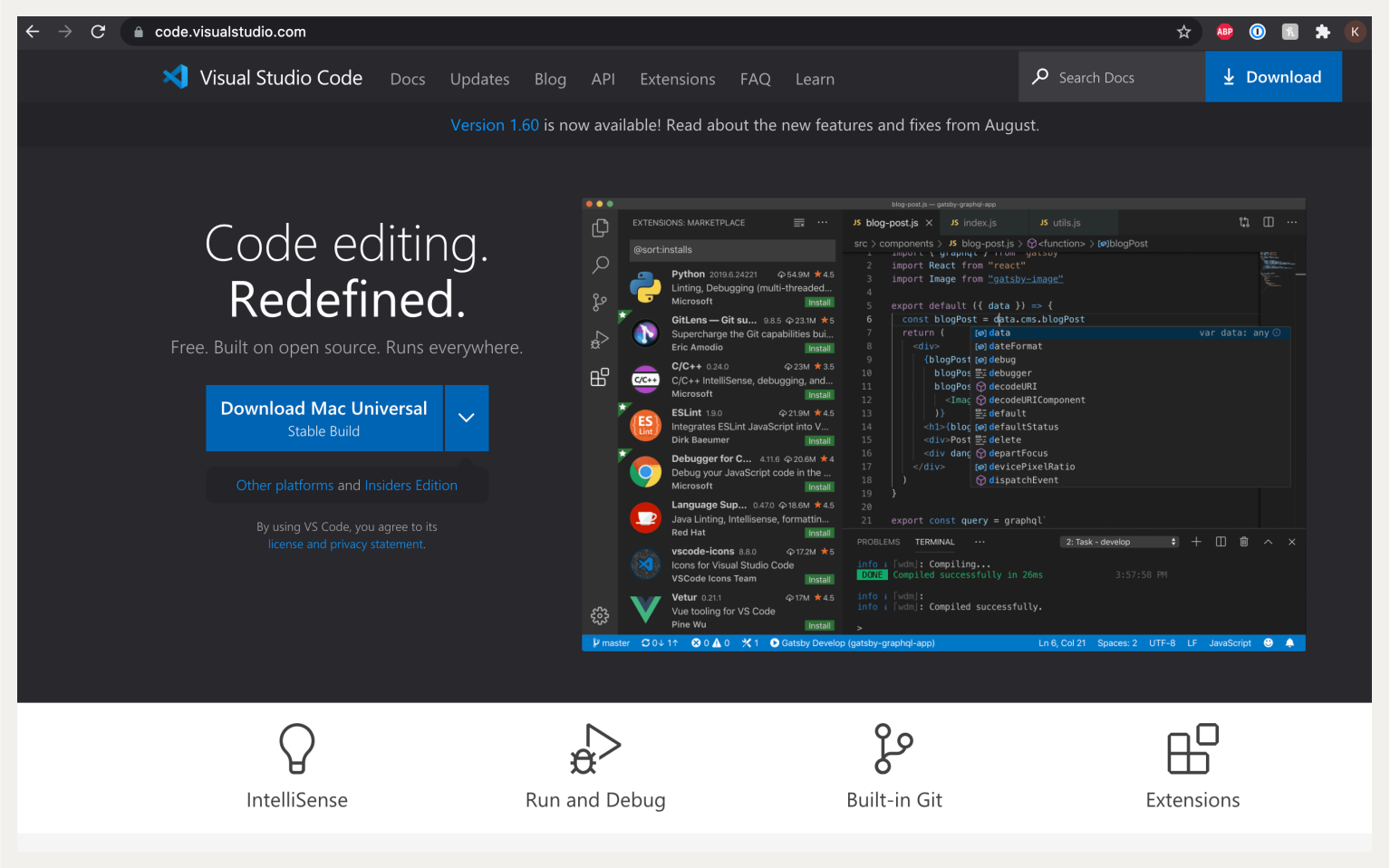
Bouwsteen 2: Cascading Style Sheets (CSS)

- De vormgeving van je HTML-elementen en pagina's
- Gaat over lettertypes, kleuren, achtergrondafbeeldingen, witruimte, layouts...
- Tot en met animaties!

Bouwsteen 3: JavaScript (JS)

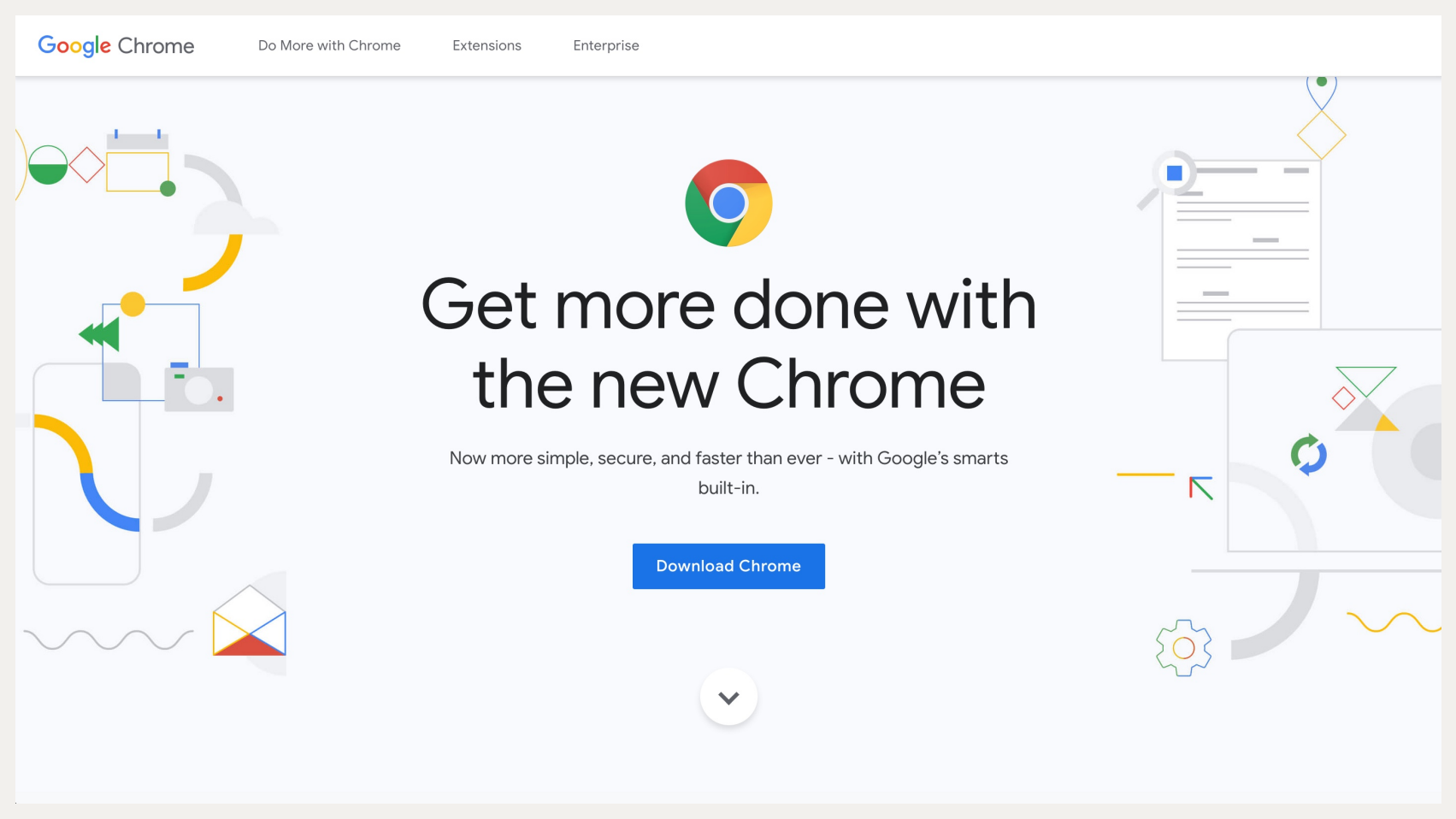
- Maakt webpagina's interactief
- JavaScript niet te verwarren met Java!
- Zeer veelzijdig: gaat vandaag veel breder dan het web

Onze werkinstrumenten: Visual Studio Code



https://code.visualstudio.com/

Onze werkinstrumenten: Google Chrome



https://www.google.com/chrome/

HTML

Dit zijn allemaal HTML-elementen

```
<h1>Ik ben een HTML-element</h1>

<a href="http://google.be">Ik ook</a>

<img src="mijn-element.jpg" alt="Ik ben een afbeelding">

<ul><li>En ik ook</li></ul>
```

- Een html-element wordt geschreven als een combinatie van tags en informatie
- De informatie krijgt betekenis door de tags die eraan verbonden worden
- De informatie zit in de meeste gevallen tussen een openingstag en een closing-tag
- Tags kunnen op hun beurt genest zitten in andere tags

HTML schrijf je een HTML-document?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Hallo wereld!</h1>
</body>
</html>
```

Dit is de minimale startversie (of skeletstructuur) van elk HTML-bestand. De browser weet dat het een html-bestand is omdat je in de naamgeving kiest voor een .html-bestand. Zoals bvb index.html

HTML schrijf je in een HTML-document

- DOCTYPE identificeert de pagina als bestaande uit HTML5
- We stoppen alles in een root-element (<html>)
- Binnen het root-element: een head en een body element
- Een meta-element: aangeven dat onze character encoding unicode versie UTF-8
- Een viewport meta-element: belangrijke instelling voor mobiele toestellen
- Verplicht ook: het title element
- In het body element komt alle html die wij gaan schrijven
- Je kan in dit voorbeeld al goed zien dat html-elementen bestaan uit tags

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 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

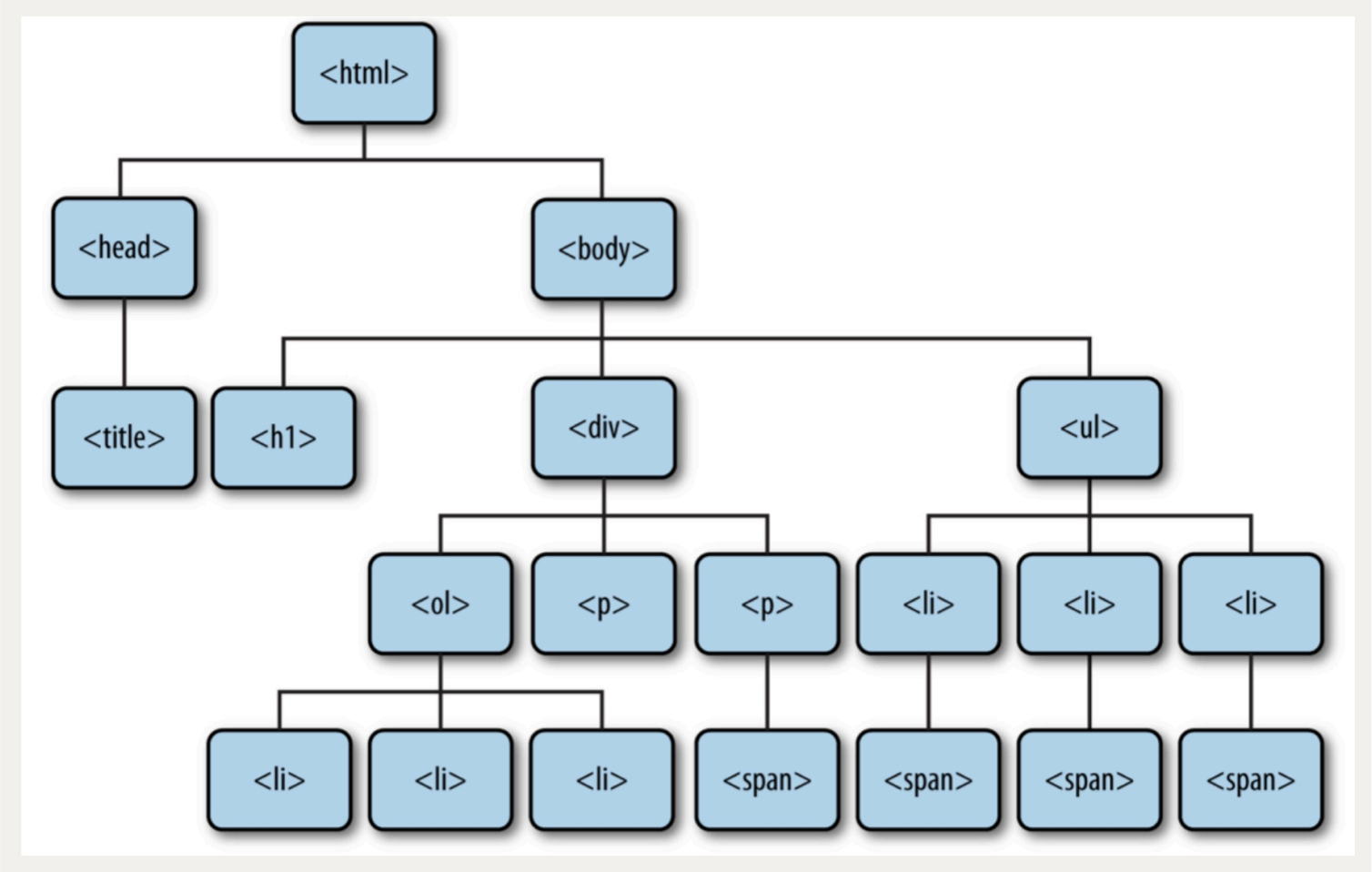
Het Document Object Model of kortweg DOM

- 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)

Het Document Object Model of kortweg DOM

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<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>
</html>
```

De DOM als boomstructuur



Dit is hoe de browser naar jouw html-pagina kijkt

Paragrafen: ...

- Het paragraaf-element is zowat de belangrijkste bouwsteen van tekstdocument
- Het begint met de -tag en eindigt met de -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: <ul> 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

Niet-geordende lijsten: <ul>...

- 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 (is aanpasbaar met CSS)
- Binnen de lijstelementen kan je vrij andere elementen gebruiken (zie volgend voorbeeld)

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

Niet-geordende lijsten: <ul>...

- 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 zullen we vaak koppelen aan html-element 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 vooruit: CSS zal je totale controle geven 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>
```

Verzamel-elementen die informatie organiseren

- Elementen als <h1>, <p>, <ul> 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 vaardigheid die je langzaam onder de knie zal krijgen

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

- 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>
        <img src="mijnlogo.jpg" alt="het Taito-logo">
        <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>
```

De aard van het element: block- of 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: 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: en

- We zagen eerder elementen als : main, header, footer, nav
- Dit zijn verzamelelementen, 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 staat voor *division*
- Het is een block-element
- Te gebruiken als een verzamel-element als er geen beter passend element bestaat

```
<div>
        <img src="mijnlogo.jpg" alt="het Taito-logo">
        <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 betekent '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>
```

Link elementen: <a>...

- "Anchor" of anker-elementen
- Hypertext links: de essentie van het web
- Hier zien we terug een (verplicht) attribuut verschijnen: het href-attribuut
- het href-attribuut verwijst naar de resource (de bron, meestal een bestand) 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>
```

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/toegepaste-informatica"
```

Relatieve URLs

- Beschrijven een padnaam relatief ten opzichte van het huidige document
- Om te linken naar documenten op je eigen site (op dezelfde webserver)
- Mag je doen door gewoon naar de bestandsnaam te wijzen
- Of door te verwijzen naar de folder waarin het bestand zit (indien het geval), gevolgd door een slash
- Indien meerdere niveau's diep, dit herhalen

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

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>
```

Afbeeldingen plaatsen met het img-element

- Is een inline element en heeft geen afsluittag
- Bevat 2 verplichte attributen: src en alt
  - src bevat de locatie van de afbeelding
  - alt voorziet in een tekstuele beschrijving van de afbeelding
- Telkens de browser een img element tegenkomt, zal het dit element proberen te vertalen naar een afbeelding

```
<p>Deze avond heb ik zin in een lekkere <img src="spaghetti.jpg" alt="spaghetti">.
Heerlijk!</p>
```

Padnamen voor afbeeldingen

- Het src-attribuut werkt op net dezelfde manier als het href-attribuut
- De resource die wordt gelinkt betreft hier wel een afbeelding (gif, jpg, png, webp)

```
<img src="afbeeldingen/spaghetti.gif" alt="Spaghetti-gerecht">

<img src="../afbeeldingen/spaghetti.gif" alt="Spaghetti-gerecht">
```

Het img element: width en height

- Je kan afmetingen meegeven met afbeeldingen met de attributen width en height
- 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)
- We zullen dit met CSS kunnen perfectioneren

```
<img src="spaghetti.jpg" alt="Spaghetti" width="250" height="250">
```

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
adipisicing 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>
```

Identificatie en classificatie met id en class

- We kwamen eerder al attributen 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
- Je mag een id, class of beiden toevoegen aan elk html-element

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="mijn_verzameling">
        <h2 id="mijn_hoofding">Mijn hoofding</h2>
        <p id="mijn_paragraaf">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>
```

Web technology - Startles

- Bedankt voor je aandacht - Tot in de labo-les!
- Email: kristof.michiels01@ap.be