



# Web Development I

**Les 1: HTML basis deel 1 - CSS  
intro**

**HO  
GENT**

# Inhoud

---

- ▶ Surfen op het web.
- ▶ Basisbegrippen HTML
- ▶ Een eerste website maken
- ▶ Basisbegrippen CSS
- ▶ CSS toevoegen aan onze website
- ▶ Omgaan met files en folders
- ▶ Introductie CSS selectors

# **Surfen op het web**

# Surfen op het web

---

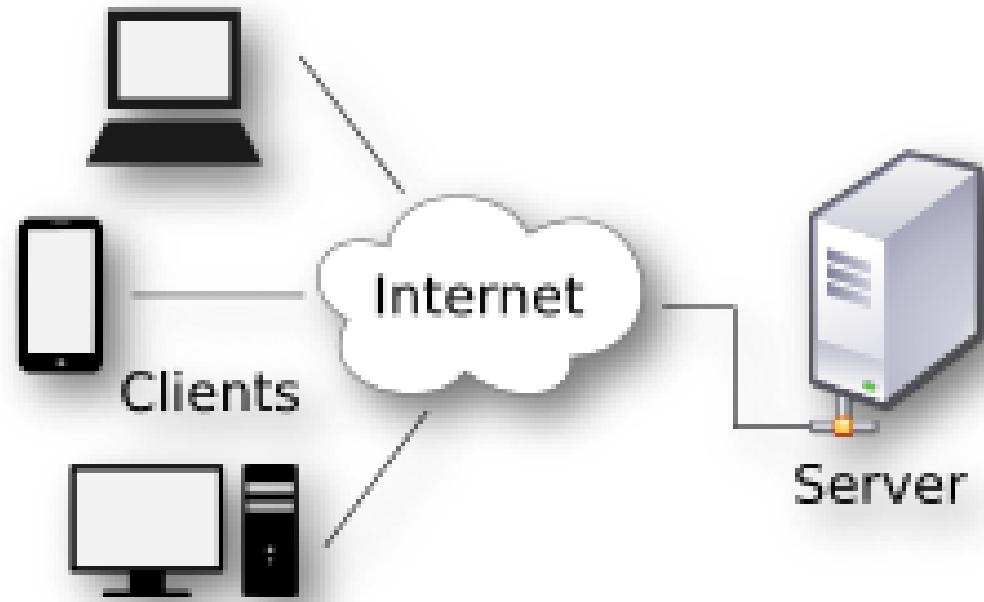
- ▶ Dagelijks zijn we actief op het Web:
  - facebook of andere (a)sociale media raadplegen
  - filmpje kijken
  - shoppen
  - nieuws lezen
  - fotootjes kijken
  - muziek (il)legaal downloaden
  - gamen

# Surfen op het web

---

- ▶ Als je een webpagina bezoekt, ga je eigenlijk een verzoek doen aan een server (URL = adres van webpagina – <https://www.vrt.be/nl/>) om een webpagina naar jouw toestel (laptop – tablet – smartphone) door te sturen.
- ▶ Als deze pagina op de server staat, zal deze naar jouw toestel doorgestuurd worden. Je browser (de client) op je toestel zal deze pagina opbouwen en weergeven.
- ▶ Surfen op het web maakt dus gebruik van een client-server architectuur. Meer details hierover in het OLOD Computer Networks.

# Client – server architecture



Voor meer info zie: [What is "The Web" and how does it work? | Web Demystified, Episode 0](#)

# Kennismaking met HTML en CSS

---

- ▶ De client (browser) begrijpt drie talen:
  - **HTML** – ‘HyperText Markup Language’ is de taal die je gebruikt om je webpagina’s te beschrijven. Met behulp van HTML-codes breng je **structuur** en **betekenis (semantiek)** aan in je webpagina’s, door middel van headings, tabellen, afbeeldingen, lijsten, ...  
(De HTML-taal voorziet ook in enkele zeer eenvoudige gebruikersinteracties: het aanklikken van verwijzingen, het invullen van tekstvelden, het doorsturen van formulierdata naar de server,...)
  - **CSS** – ‘Cascading Style Sheets’ is de taal waarmee je de **opmaak** en **layout** van je webpagina’s beschrijft.  
(Met behulp van CSS zijn eenvoudige gebruikersinteracties mogelijk: bijv. een grotere afbeelding weergeven als je over een miniatuur afbeelding hovert, ...)
  - **JavaScript** is een programmeertaal waarmee je **complexe interacties** aan je webpagina’s kan toevoegen.

# OLODS Web Development

---

- ▶ Web Development I:
  - behandelt HTML en CSS: maken van responsive statische websites.
- ▶ Web Development II
  - behandelt JavaScript (client side)
- ▶ Front-end Web Development
  - gebruik van een client side framework
- ▶ Web Services
  - behandelt de server side.
- ▶ Enterprise Web Development: Java
  - Combinatie van client & server in Java.



# **HTML**

HyperText Markup Language

# Een HTML-document aanmaken

---

- ▶ Een HTML-document, HTML-pagina, webpagina of HTML-bestand
  - is een tekstbestand met als bestandsextensie .html
  - is platform onafhankelijk
  - je kan webpagina's maken:
    - met behulp van een eenvoudig tekstbewerkingsprogramma (text editor) bijv. notepad, brackets, ...
    - met een IDE (Integrated Development Environment) bijv. Visual Studio, WebStorm, ...
    - met een WYSIWYG HTML-editor programma bijv. Adobe Dreamweaver, ...
  - voor dit olod gebruiken we **Visual Studio Code** als text editor.
- ▶ *Voor extra info zie What text editors are available?*

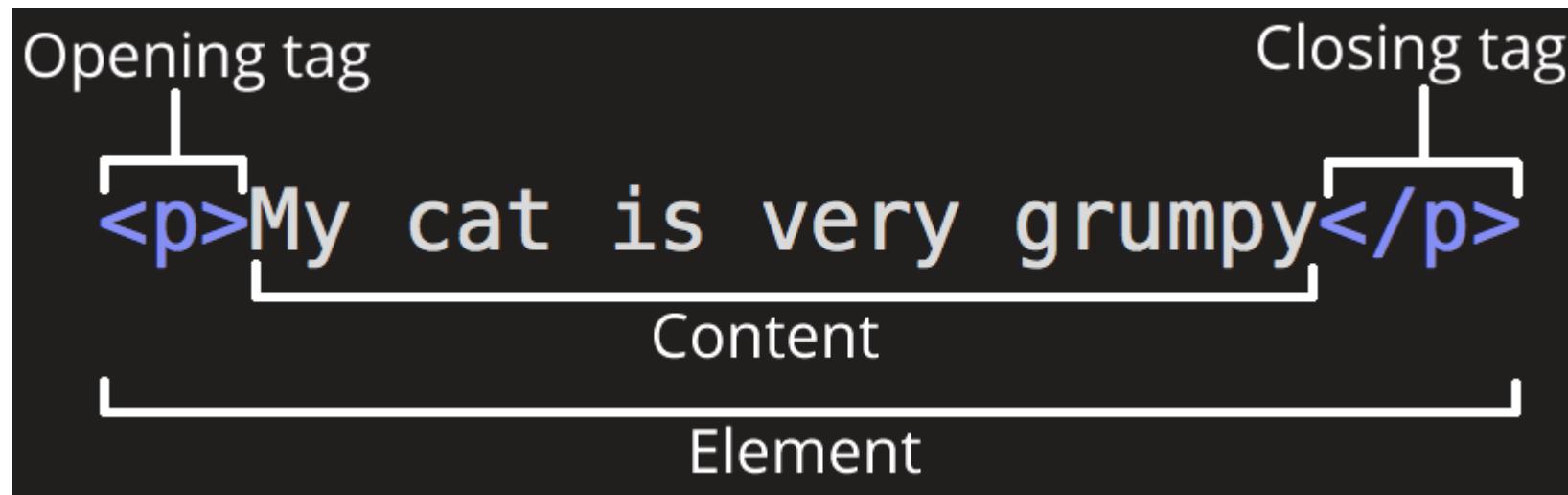
# HTML syntax

---

- ▶ HTML beschrijft de **structuur** van je HTML document en de **betekenis** (semantiek) van de verschillende onderdelen.
- ▶ De drie belangrijkste begrippen zijn:
  - element
  - tag
  - attribute

# Elements, Tags

- ▶ Een HTML-pagina (webpagina) bevat HTML-elementen. Een voorbeeld van een veel gebruikt HTML-element is een p-element (paragraph element).
- ▶ Merk op dat het begin en het einde van het p-element aangeduid worden door middel van een ‘Opening tag’ `<p>` en een ‘Closing tag’ `</p>`.



BRON: [https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/HTML basics#anatomy of an html element](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics#anatomy_of_an_html_element)

# void elements

- ▶ Sommige HTML-elementen hebben alleen een start-tag en geen end-tag we noemen ze void elements.
- ▶ Het Line Break element `<br>` is bijvoorbeeld een void element.

```
<p>De beste hogeschool in Vlaanderen en<br>  
kleine omstreken.</p>
```

- ▶ In plaats van `<br>` schrijft men ook soms `<br />` (omdat deze syntax compatibel is met XML)

# Nesting elements

---

- ▶ Je kan elementen in andere elementen stoppen, dit noemen we nesten van elementen. Om het woord **very** te benadrukken in onderstaande paragraaf, kunnen we het woord in een strong-element stoppen. Het strong-element is dan genest in het p-element.

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

# Attributes

- ▶ Attributen bevatten extra informatie over een element. Elementen kunnen geen, één of meerdere attributen hebben.
- ▶ Een attribuut bestaat uit een naam en een waarde en wordt in de openingstag geplaatst na de naam van de tag.
- ▶ De attribuutwaarde wordt toegekend aan de attribuutnaam door een gelijkheidsteken. De waarde staat tussen dubbele aanhalingstekens.  
(dit mogen ook enkele aanhalingstekens zijn, maar wij zullen steeds dubbele gebruiken)
- ▶ Syntax: **attribute name="attribute value"**

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

In het voorbeeld is **class** de attribuutnaam en **editor-note** de attribuutwaarde.

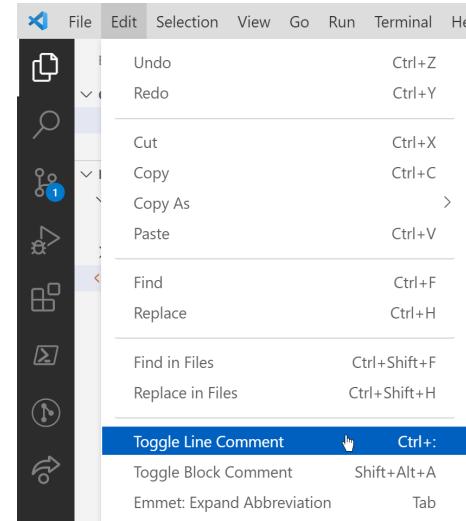
# Commentaar (comments)

- ▶ Naast HTML-tags kan je in een webpagina ook commentaar opnemen:

```
<!-- dit is commentaar -->
```

- ▶ De sneltoets in Visual Studio Code om commentaar aan/uit te zetten is (voor een azerty-toetsenbord)

**Ctrl+:**



# Voorbeelden van HTML-elementen

---

- ▶ Koppen: <h1> tot <h6>
- ▶ Paragraaf: <p>
- ▶ Hyperlink: <a>
- ▶ Afbeelding: <img>

# Koppen en paragrafen

---

- ▶ De <h1> tot <h6> HTML-elementen stellen zes niveaus van section headings voor.

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
```

- ▶ Het <p> HTML-element stelt een paragraaf voor.

```
<p>
  This is a paragraph of text.
  This is a paragraph of text.
</p>
```

# Hyperlinks – het -element (anchor element)

- Met het element <a> (of het anchor element) kunnen we een hyperlink creëren in een HTML document. Met het attribuut **href** kunnen we aangeven naar welke locatie de bezoeker moet gaan wanneer er op de hyperlink geklikt wordt. De inhoud (content) van het element is een indicatie van de bestemming van de link en wordt in browsers standaard onderlijnd weergegeven.

```
<p>
```

```
    Bekijk de info op <a href="https://www.hogent.be">HoGent</a> om meer te  
    weten te komen over Hogeschool Gent.
```

```
</p>
```

# Afbeeldingen – het -element

- ▶ Om een afbeelding toe te voegen aan een webpagina wordt het <img>-element gebruikt.
- ▶ Dit element bevat geen image maar wel een verwijzing naar de image. Hiervoor dient het **src**-attribuut. De waarde van dit attribuut bevat de locatie van de image. Dit kan binnen de website zijn, maar het attribuut kan ook verwijzen naar een image op een andere website.
- ▶ Een verplicht attribuut bij is **alt**. Dit is een korte beschrijving van de image. Dit is o.a. voor slechtziende gebruikers die door middel van ‘screen reading software’ pagina’s “bekijken”. Deze software zal de waarde van het alt-attribuut voorlezen.
- ▶ Merk op: het img-element heeft geen content (en dus geen closing tag). Het is een void-element.

```

```

# Globale attributen

---

- ▶ Welke attributen je allemaal kan gebruiken bij een bepaald HTML-element kan je o.a. opzoeken op de MDN website.
- ▶ Een aantal attributen kan je bij elk HTML-element gebruiken. We noemen deze globale attributen. Enkele veel gebruikte globale attributen zijn:
  - class: specificeert een of meerdere class names voor een element.
  - id: definieert een unieke identifier (ID) voor een element.
  - lang: stelt de taal in van een element.

Zie MDN voor een overzicht van de globale attributen.

# Het id-attribuut

---

- ▶ Het id-attribuut definieert een unieke identifier (ID) voor een element.
  - id's zijn case sensitive (hoofdlettergevoelig)
  - id's mogen geen spaties bevatten

```
<button id="feedback-button">Feedback</button>
```

- Als we een element voorzien van een id dan kunnen we er naar linken of het id gebruiken als identificatie in een Script of een CSS style sheet (zie verderop)

# Het class-attribuut

- ▶ Het class-attibuut specificeert één of meerdere class names voor een element (in deze introductie zullen we ons beperken tot één class name, zie verderop voor het gebruik van meerdere class names bij een element)
  - class names zijn case sensitive (hoofdlettergevoelig)
  - class names mogen geen spaties bevatten

```
<p>Narrator: This is the beginning of the play.</p>
<p class="note">Above point sounds a bit obvious. Remove/rewrite?</p>
<p>Narrator: I must warn you now folks that this beginning is very exciting.</p>
<p class="note">[Lights go up and wind blows; Caspian enters stage right]</p>
```

- classes laten ons toe om meerdere elementen te selecteren voor opmaak met CSS (of voor manipulatie met JavaScript).

# Het lang-attribuut

- ▶ Via het lang-attribuut stel je de taal in van een element. Dit is niet verplicht maar is handig voor zoekrobotten, screenreaders, ...
- ▶ Mogelijke waarden zijn "en" voor Engels, "fr" voor Frans en "nl" voor Nederlands.
- ▶ Best practice: stel steeds de standaardtaal in van je webpagina door aan het html-element een **lang**-attribuut toe te voegen.

```
<html lang="en">
```

Het is ook mogelijk om naast de taal een regio te vermelden. Bijvoorbeeld **en-GB** => Engels zoals gebruikt in Groot-Brittannië.

```
<p lang="en-GB">This paragraph is defined as British English.</p>
```

# HTML5 specificatie

---

- ▶ In dit OLOD wordt gewerkt met HTML5
- ▶ Een overzicht van de verschillende versies van HTML: zie appendix.
- ▶ De HTML5 specificatie: <http://www.w3.org/TR/html5/>.

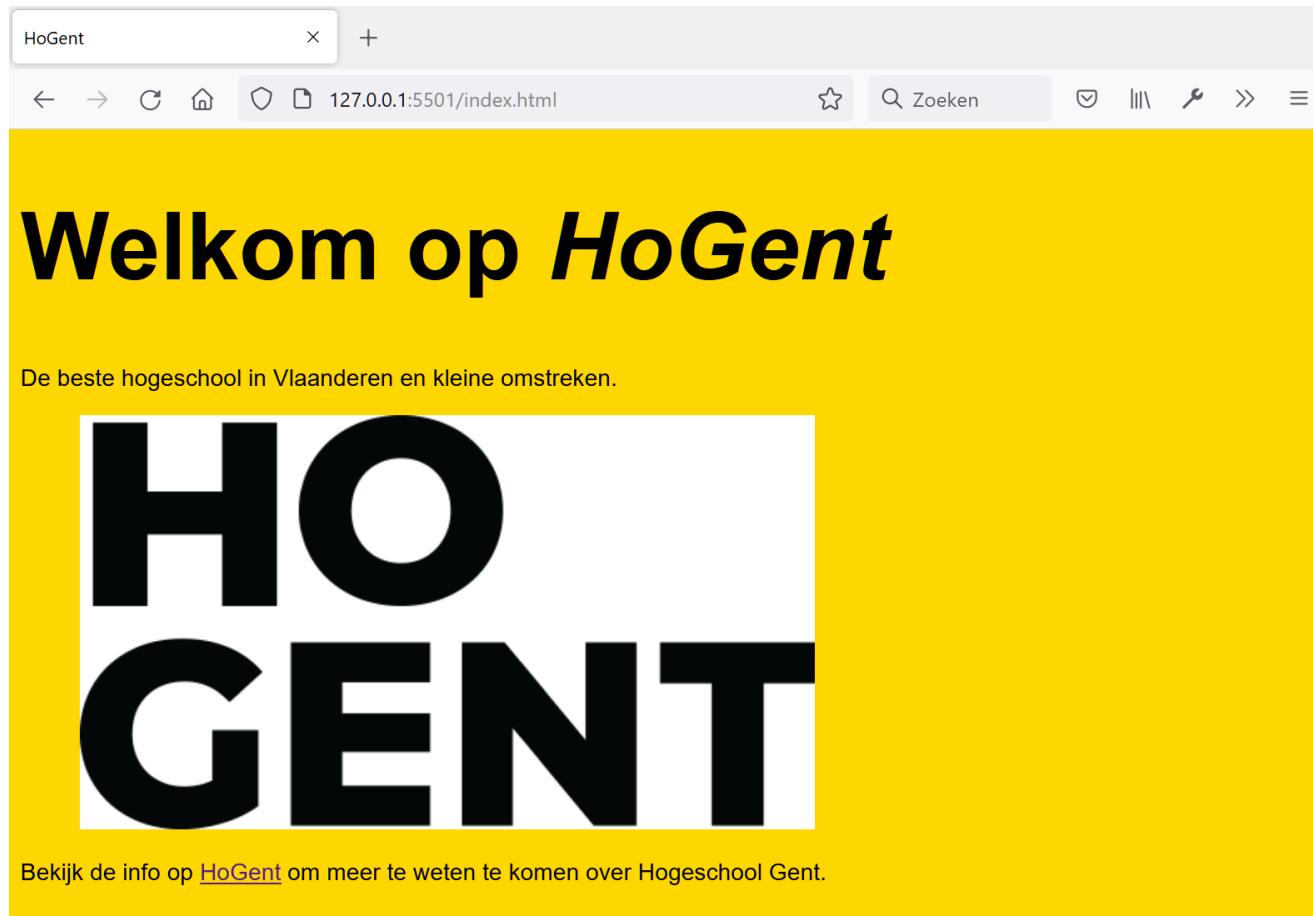


Het **World Wide Web Consortium (W3C)** en het **WHATWG** zijn de organisaties die momenteel verantwoordelijk zijn voor de ontwikkeling van HTML5

# Een eerste website maken

# Te bekomen eindresultaat

- Hieronder zie je het eindresultaat van de website die we gaan maken.



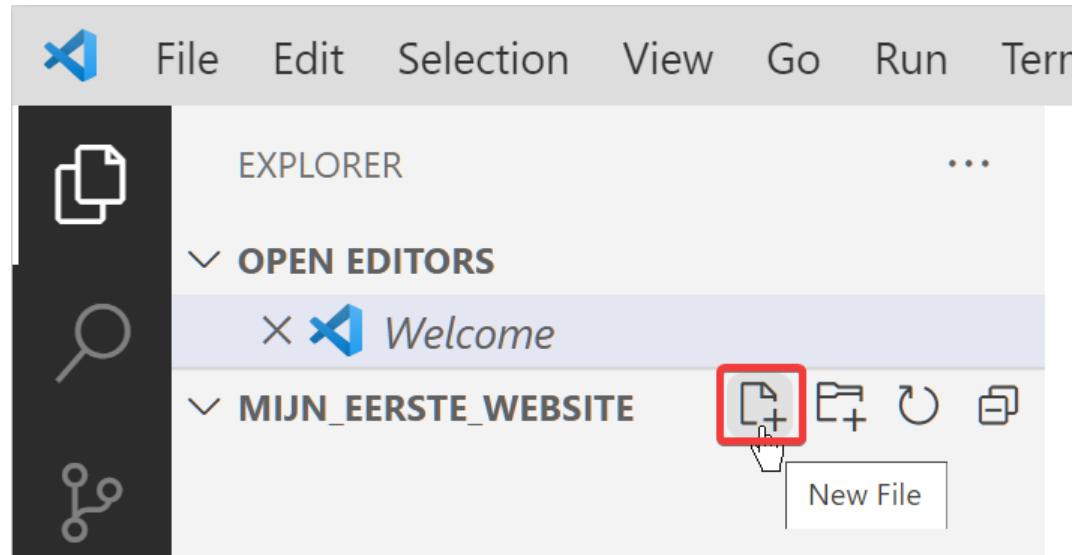
# Een eerste website maken



- ▶ Een website bestaat steeds uit meerdere bestanden (html-bestanden, CSS-bestanden, afbeeldingsbestanden, ...) bijgevolg starten we met het maken van een nieuwe map om al deze bestanden in te bewaren:
  - maak in de map **01TH-HTMLBasisDeel1-CSSIntro** een map **mijn\_eerste\_website**.  
(doe dit bijvoorbeeld via de Windows Verkenner)
- ▶ De afbeelding met het logo van HOGENT kan je downloaden vanaf het eindresultaat van de website:
  - maak in de map **mijn\_eerste\_website** een submap **images** en plaats daarin het bestand met het logo nl. **hogent\_logo.png**.

# Een eerste website maken (vervolg) 🚀

- ▶ Maak in de map **mijn\_eerste\_website** een leeg **index.html** bestand aan.  
(open de folder in Visual Studio Code en klik op **New File**)



De startpagina (Home Page) van je website geef je steeds als naam **index.html**.  
De namen voor de andere webpagina's mag je zelf kiezen.

# Opbouw van een HTML5 document

Alle webpagina's hebben een gelijkaardige opbouw:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World!!</h1>
    <p>This is a web page.</p>
  </body>
</html>
```

# Opbouw van een HTML5 document

---

<!DOCTYPE html>

- is steeds de eerste lijn in je document
- het is geen html-tag, het is een instructie voor de browser en geeft aan volgens welke specificatie de HTML moet worden geïnterpreteerd door de browser.
- <!DOCTYPE html> wijst op HTML versie 5.
- alhoewel DOCTYPE ook in kleine letters mag geschreven worden schrijven we het in hoofdletters om aan te geven dat het geen html-tag is.
- een DOCTYPE is ook nodig als je jouw HTML-code wilt laten controleren op fouten door een Validator (zie verderop).

# Opbouw van een HTML5 document

---

- ▶ Na de DOCTYPE-regel volgt het **html**-element.  
Het html-element bevat twee elementen nl. **head** en **body**.
  - **head**-element: bevat metadata, dit is extra informatie over de pagina zoals de gebruikte karakterset, de title van je document, linken naar externe bestanden zoals een css-bestand (zie verderop), ...  
De title wordt weergegeven in je browser tabblad, dit is het enige visuele element van het head-element.
  - **body**-element:  
bevat alle zichtbare content van onze webpagina.

# Opdracht



- ▶ Voer in **index.html** de HTML-code in die je terugvindt op de volgende slide.
  - Maak hierbij gebruik van emmet in Visual Studio Code:

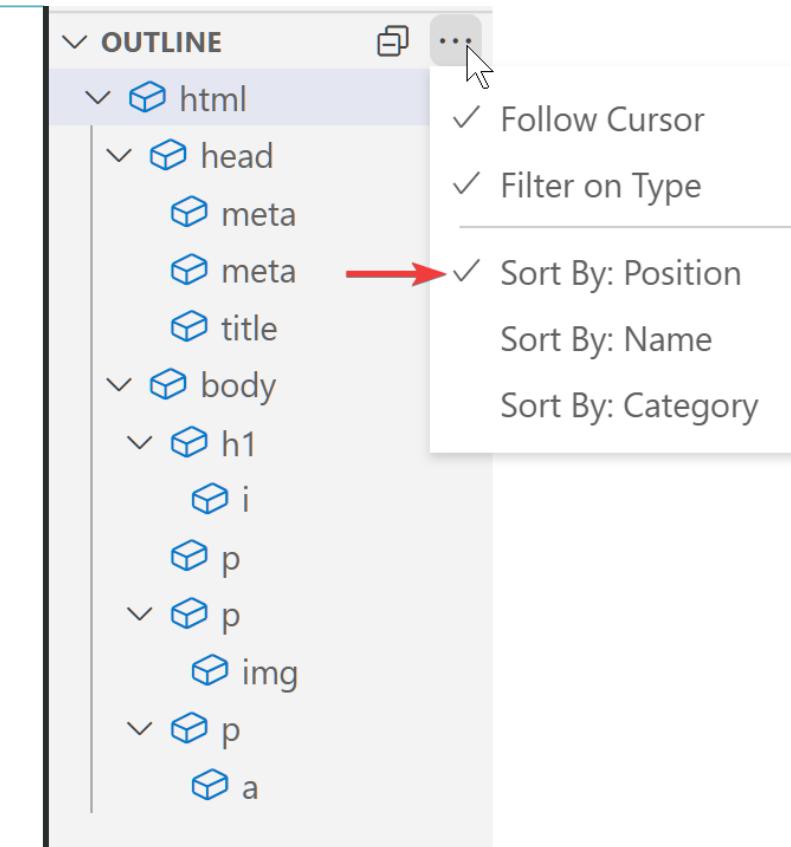
Emmet abbreviation	Emmet expansion
!<tab>	<!DOCTYPE html> <html lang="en"> ... </html>
h1<tab>	<h1></h1>
i<tab>	<i></i>
img<tab>	<img src="" alt="">
...	

- Tip Formatteer regelmatig je code: klik met de rechtermuisknop in de file en kies **Format document** of druk **Shift+Alt+F**

# De Document tree (DOM)

- ▶ Je kan de document tree visualiseren in VS Code:  
in Explorer > Outline > Sort By: Position

```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>HoGent</title>
  </head>
  <body>
    <h1>Welkom op <i>Hogeschool Gent</i></h1>
    <p>De beste hogeschool in Vlaanderen en kleine omstreken.</p>
    <p></p>
    <p>
      Bekijk de info op <a href="https://www.hogent.be">HoGent</a> om
      meer te weten te komen over Hogeschool Gent.
    </p>
  </body>
</html>
```



# Gebruik van de live server



- ▶ Om de live server te starten en index.html weer te geven in jouw standaard Browser klik je in VS Code met de rechtermuisknop op **index.html** en kies je **Open with Live Server**.
- ▶ Je kan jouw webpagina ook bekijken door deze gewoon te openen via het File System (dubbelklik op het html-bestand in de Windows Verkenner), maar er zijn een aantal voordelen aan het gebruik van de live server.
  - Telkens als je een bestand bewaart in VS Code zal jouw webpagina automatisch vernieuwd worden in de browser.  
Tip Zet **File > Auto save** aan.
  - In Web Development II, ... zullen we sowieso een webserver nodig hebben, dus kunnen we er maar alvast aan wennen.



**CSS**

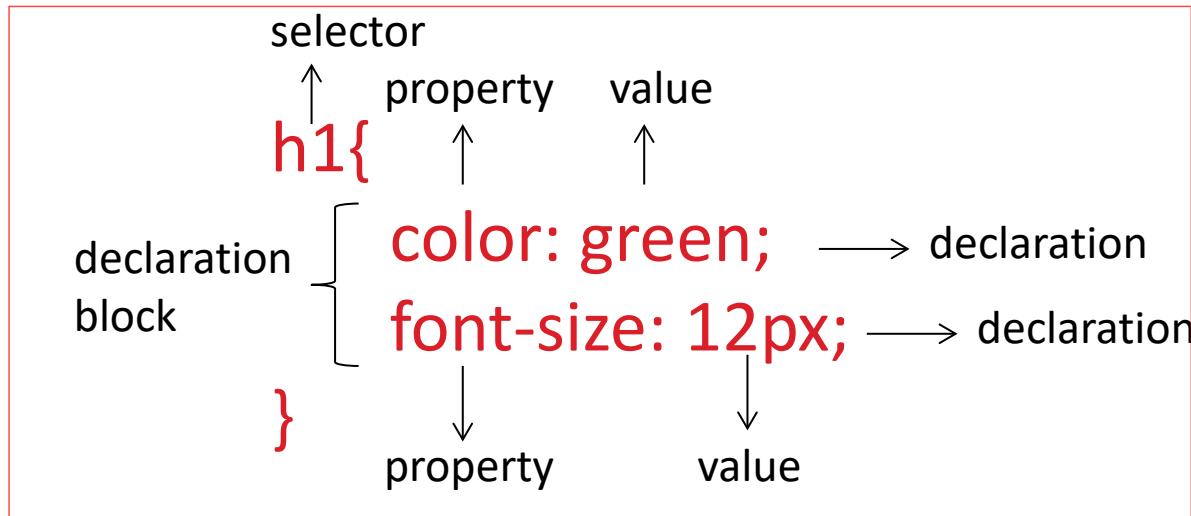
Cascading Style Sheets

- ▶ Cascading Style Sheets (afgekort CSS) is de technologie voor de opmaak van je webpagina.
- ▶ CSS-code bestaat uit **stijlregels** waarmee we de **stijl** (bijv. kleur, de lettergrootte, ...) en de **lay-out** (bijv. meerdere kolommen, ...) van webpagina's kunnen vastleggen.
- ▶ De 'look and feel' van een ganse website kan met CSS snel aangepast worden.

Leuk voorbeeld: <http://csszengarden.com/>

- ▶ De belangrijkste begrippen in CSS zijn:
  - selector
  - style rule (stijlregel), declaration, property, value

# Opbouw van een stijlregel



- ▶ Een aantal begrippen
  - **Selector:** duidt aan welk(e) element(en) beïnvloed worden door deze stijlregel. Is het ‘target’ in ons HTML document (hier alle h1-elementen).
  - **Declaration:** is van de vorm **property: value;**  
De declaratie eindigt altijd op een ;
  - **Property:** de CSS-eigenschap die je wil instellen
  - **Value:** waarde voor de eigenschap.
  - **Declaration block:** alles wat tussen {} staat

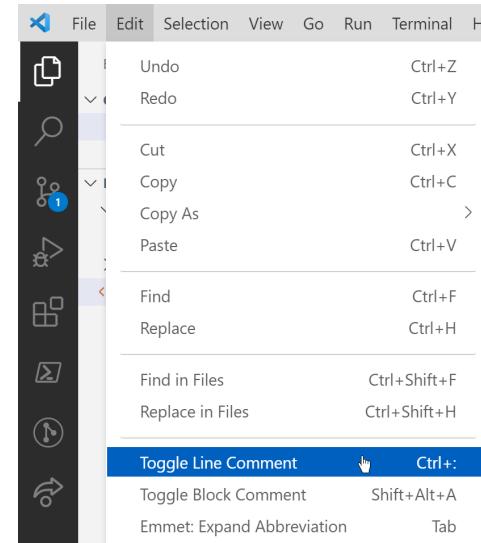
# Commentaar (comments)

- ▶ De manier om commentaarregels op te nemen in CSS-code is:

```
/* dit is commentaar */
```

- ▶ De sneltoets in Visual Studio Code om commentaar aan/uit te zetten is voor een azerty-toetsenbord

**Ctrl+:**



# **CSS linken aan HTML**

# CSS toevoegen aan een webpagina

---

- ▶ Er zijn 3 manieren waarop men CSS stijlregels kan toevoegen aan een webpagina.
- ▶ In deze les bekijken we enkel het werken met **External style sheets**: CSS stijlregels in een extern bestand. Voor de andere manieren zie verderop: CSS basis.

# External style sheets

---

- ▶ Stijlregels worden meestal opgenomen in een apart css-bestand.  
Vanuit de webpagina's wordt dan verwezen naar het externe css-bestand.  
Er kunnen meerdere css-bestanden gekoppeld worden aan 1 webpagina.
- ▶ En ook omgekeerd. Alle webpagina's van een website kunnen verwijzen  
naar eenzelfde css-bestand.  
Een stijlwijziging in dat css-bestand geldt dan voor alle webpagina's op de  
hele website. Alle opmaak ziet er dan uniform uit voor de volledige  
website.

# External style sheet linken aan een webpagina

```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>HoGent</title>
    <link rel="stylesheet" href="css/styles.css" />
  </head>
```



Wijst naar het externe css-bestand

# Opdracht: css-bestand toevoegen



- ▶ Maak een map **css** aan.
- ▶ Maak in deze map een leeg css-bestand aan met naam **styles.css**
- ▶ Voeg in **index.html** de link naar het css-bestand toe.

```
<link rel="stylesheet" href="css/styles.css" />
```

Je kan dit snel bekomen door `link<tab>` te typen. Visual Studio Code zal dan de emmet-afkorting uitbreiden naar `<link rel="stylesheet" href="">` waarna je enkel nog het pad naar het css-bestand moet invullen. Hiervoor kan je dan eventueel nog `Ctrl+spatiebalk` gebruiken om een lijst met suggesties te bekomen.

# Opdracht: css-code toevoegen



- ▶ Voeg onderstaande stijlregels toe aan het css-bestand. Ook bij het invoeren van de CSS-properties kan je emmet gebruiken:

ff<tab> => font-family

bgc<tab> => background-color

...

Zie ook <https://docs.emmet.io/cheat-sheet/>

```
body{  
    /*lettertype*/  
    font-family: sans-serif;  
    /*achtergrondkleur*/  
    background-color:gold;  
}  
h1{  
    /* onze heading een beetje groter maken */  
    font-size: 4rem;  
}
```

- ▶ Bekijk het resultaat met de ‘live server’.

# **Omgaan met files en folders**

# Richtlijnen naamgeving files en folders

---

- ▶ Gebruik steeds kleine letters voor namen van mappen (folders) en bestanden (files).
  - Op Windows zijn map- en bestandsnamen case-insensitive (hoofdletterongevoelig) maar dit is niet het geval op linux.
- ▶ Gebruik geen spaties.
  - Gebruik min-tekens of onderstrepingsstekens in map- en bestandsnamen ter vervanging van spaties.
  - Google raadt aan om min-tekens te gebruiken in plaats van onderstrepingsstekens. Wil je dus jouw webpagina optimaliseren voor de google zoekrobot gebruik dan min-tekens ipv van onderstrepingsstekens.
- ▶ Vermijd speciale karakters in map- en bestandsnamen. Sommige karakters zijn niet toegestaan en andere hebben een speciale betekenis.

# Folder structuur website

---

- ▶ Het is wenselijk om je website op te delen in folders, zodat alles overzichtelijk en geordend blijft.
- ▶ Folders hebben een eenvoudige naam, meestal 1 woord.
- ▶ Bij een kleine website met een **beperkt aantal pagina's** plaats je alle HTML bestanden in de root folder van je site.
- ▶ Bij een grotere website wordt er steeds gewerkt met een menu dat per menu-item een aantal pagina's bevat. Je maakt dan best een folder per menu-item aan, waarin alle pagina's geplaatst worden.

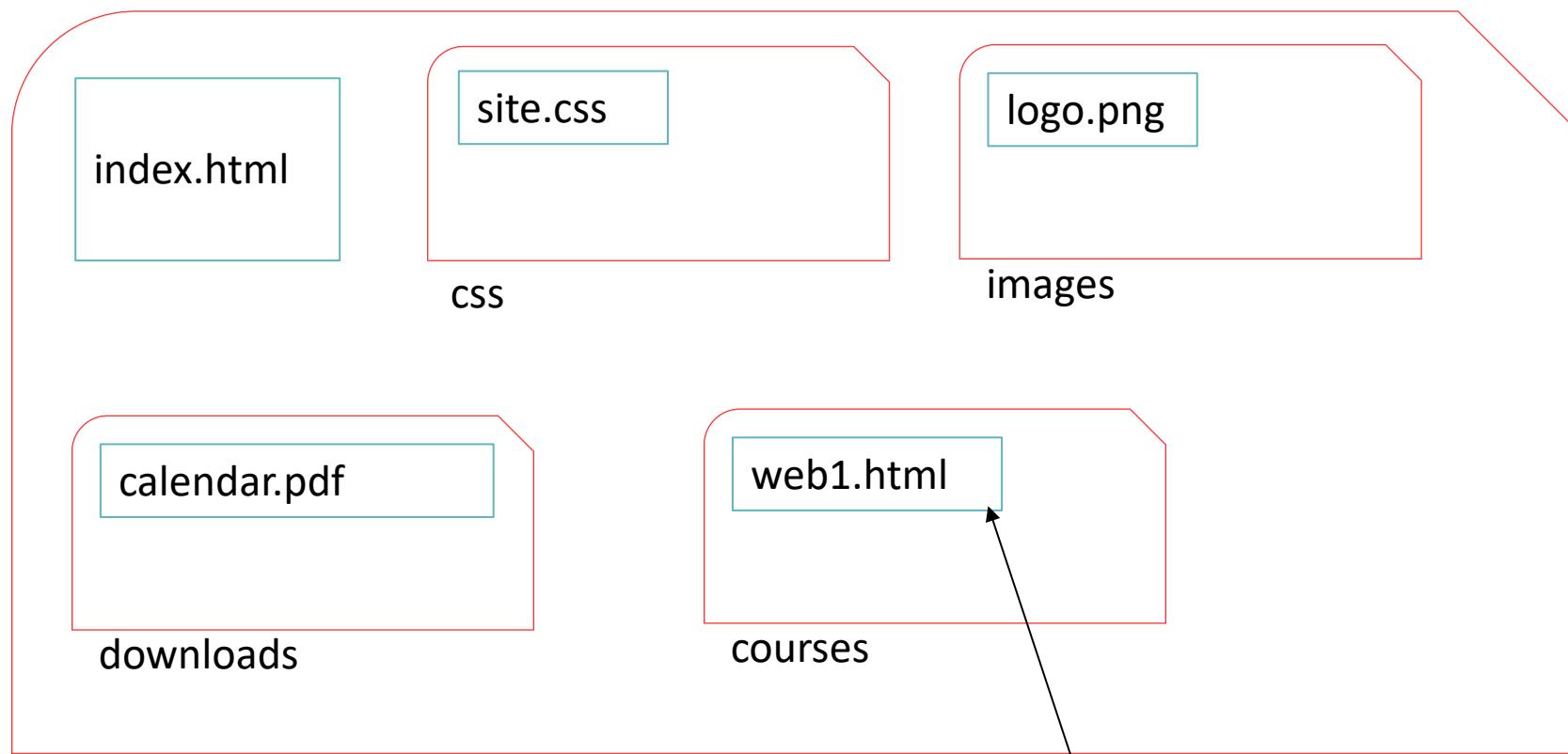
# Folder structuur website (vervolg)

---

- ▶ De root bevat steeds je index.html (startpagina)
  - Webservers gaan, indien er geen bestandsnaam vermeld is in de URL automatisch op zoek naar de index.html pagina
    - <http://website.com> -> <http://website.com/index.html>
- ▶ De overige bestanden zoals afbeeldingen, CSS, pdf-bestanden, javascript plaats je in aparte mappen (eventueel samen in de map assets).
  - folder “images” voor je afbeeldingen
  - folder “css” of “styles” voor de stijlbestanden.
- ▶ Merk op: indien je veel afbeeldingen gebruikt, kan je binnen de map images zelf meerdere mappen aanmaken per onderwerp.

# Folder structuur website

- ▶ Folder structuur bepaalt ook de url.  
Stel: een website gehost op [www.hogent.be](http://www.hogent.be)



<http://www.hogent.be/courses/web1.html>

# **Intro CSS selectors**

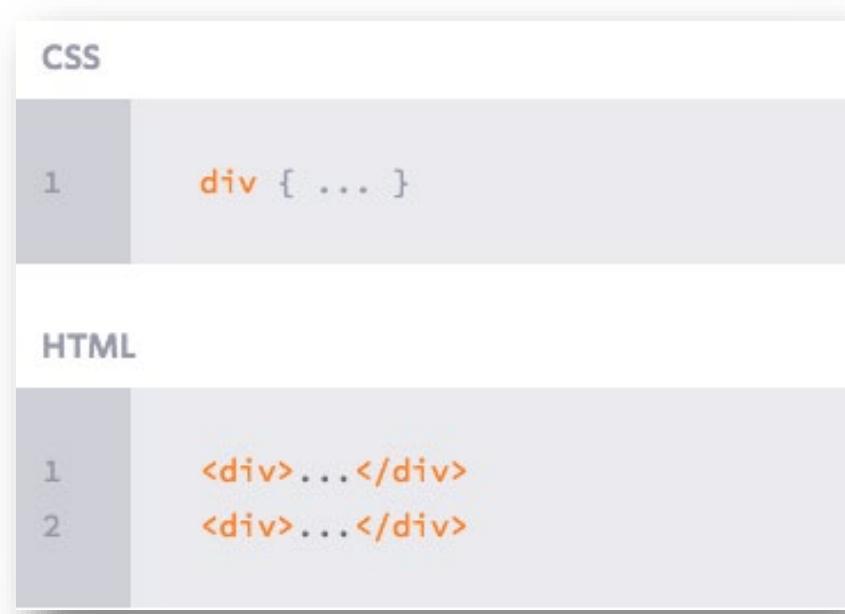
# Introductie CSS selectors

---

- ▶ Overzicht
  - Type selector
  - Class selector
  - Id selector
  - Group selector
  
- ▶ Meer over CSS selectors, zie verderop: CSS basis

# Type Selectors

- ▶ Type Selectors selecteren alle elementen van een bepaald type.
- ▶ Indien we alle `<div>` elementen eenzelfde stijl willen geven, moeten we alle div's selecteren. Dit gebeurt door de naam van het element als selector te gebruiken.



The image shows a screenshot of a code editor interface. On the left, there is a vertical toolbar with icons for file operations. The main area is divided into two tabs: 'CSS' at the top and 'HTML' below it. In the 'CSS' tab, line 1 contains the CSS rule `div { ... }`. In the 'HTML' tab, line 1 contains the opening tag `<div>...` and line 2 contains the closing tag `</div>`.

```
1 div { ... }
<div>...
</div>
```

# Class Selectors

- ▶ Class selectors laten toe om verschillende elementen dezelfde opmaak te geven. Deze elementen hebben allen dezelfde value voor het class-attribuut (voor elementen met meerdere classes zie verderop)
- ▶ De naam van een class selector begint met een . (punt) gevolgd door de waarde van het class-attribuut.



The image shows a screenshot of a code editor with two tabs: 'CSS' and 'HTML'. In the 'CSS' tab, line 1 contains the rule '.awesome { ... }'. In the 'HTML' tab, lines 1 and 2 contain the elements '<div class="awesome">...</div>' and '<p class="awesome">...</p>'. The code is color-coded: the class names ('awesome') are orange, and the HTML tags ('div', 'p') are blue.

```
CSS
1 .awesome { ... }

HTML
1 <div class="awesome">...</div>
2 <p class="awesome">...</p>
```

# ID Selectors

- ▶ ID selectors worden gebruikt om een uniek deel (element) van de pagina te identificeren en worden in de style sheet voorafgegaan door een #.
- ▶ Geef aan een element een id attribuut met een zelfgekozen waarde. Een id mag maar één keer voorkomen binnen dezelfde pagina.
- ▶ Id's worden in CSS minder gebruikt dan classes.



The image shows a screenshot of a code editor with two tabs: 'CSS' and 'HTML'. In the 'CSS' tab, there is one line of code: '1 #shayhowe { ... }'. In the 'HTML' tab, there is one line of code: '1 <div id="shayhowe">...</div>'. The code is color-coded: the '#shayhowe' in the CSS selector and the 'id="shayhowe"' in the HTML attribute are orange, while the class name and attribute value are green.

```
1 #shayhowe { ... }
1 <div id="shayhowe">...</div>
```

# Group Selectors

- ▶ Group selectors worden gebruikt als je dezelfde stijl wilt gebruiken bij verschillende CSS selectors.
- ▶ De CSS selectors worden in de Group selector gescheiden van elkaar met behulp van een komma.

```
h1,.note, a {  
    font-style: italic;  
    color: red;  
}
```

# Opdracht: CSS-bestand aanpassen



- ▶ Voeg de twee onderstaande stijlregels toe aan het css-bestand:

```
img{  
    /* een achtergrondkleur instellen voor het transparante logo */  
    background-color: white;  
    /* de afbeelding responsive maken */  
    max-width: 100%;  
}  
  
.inspringing {  
    /* sommige paragrafen geven we een linkse inspringing */  
    margin-inline: 40px;  
}
```

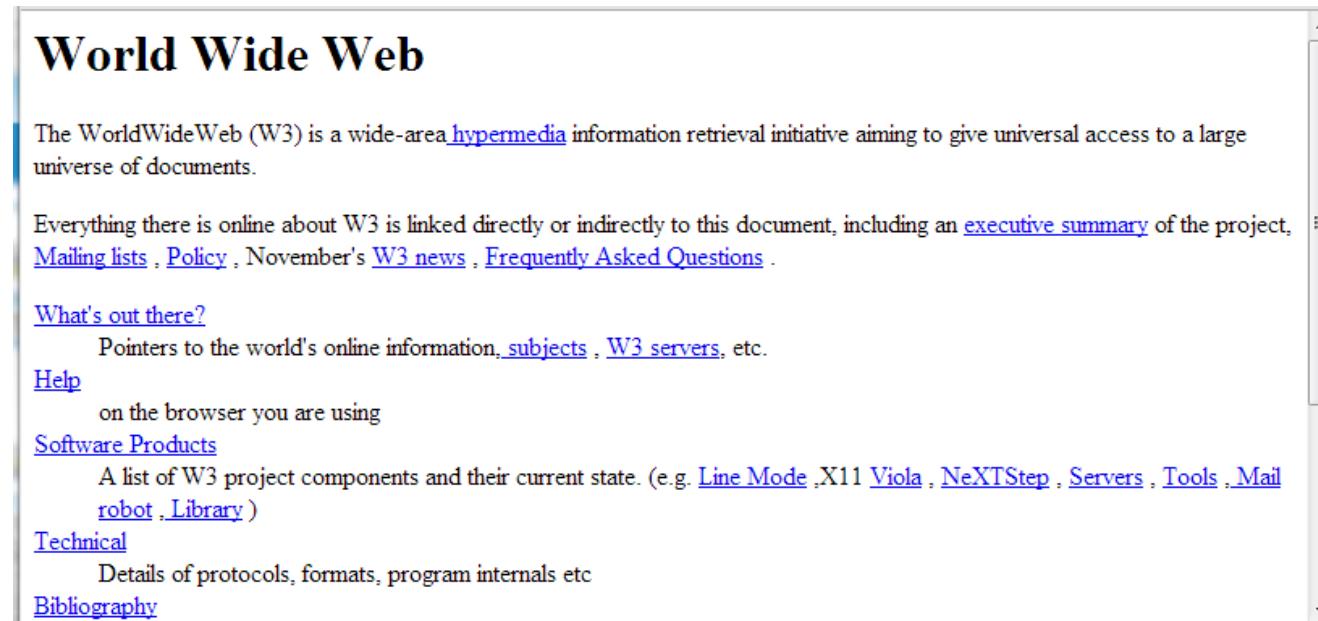
- ▶ Voeg aan het tweede p-element in index.html een **class**-attribuut toe met als waarde **inspringing**.

```
<p class="inspringing">...
```

# **Appendix**

# Eerste website ooit

- ▶ De eerste website die ooit gemaakt werd, is terug te vinden op  
<http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html>



The screenshot shows a single-page website with a dark background and white text. The title "World Wide Web" is at the top. Below it is a paragraph about the W3 project. A horizontal line follows, then a section with links like "What's out there?", "Help", "Software Products", and "Technical". At the bottom is a "Bibliography" section.

**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.

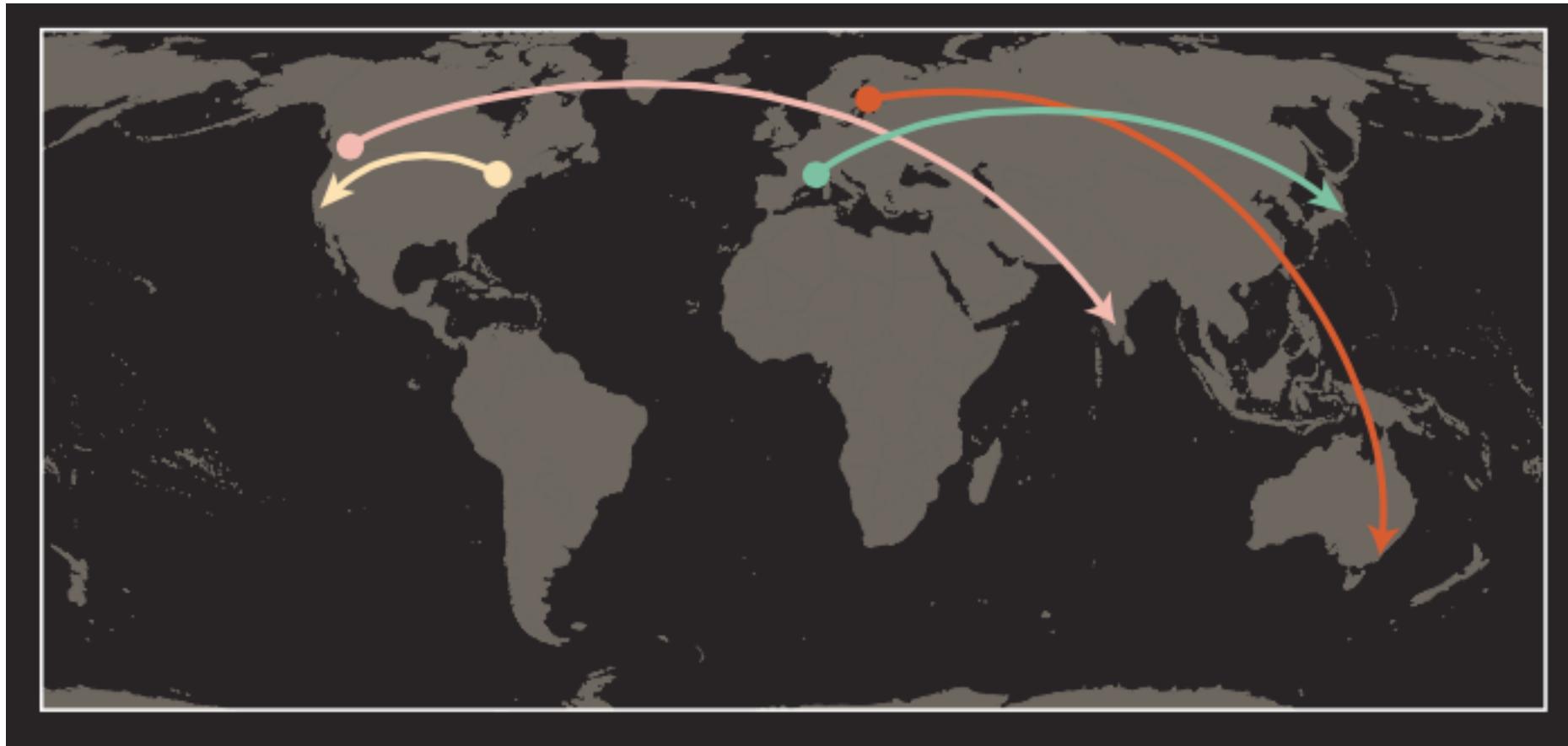
[Help](#)  
on the browser you are using

[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](#)

# Werking van het WWW



# Een beetje geschiedenis (Wikipedia)

---

- ▶ HTML werd in 1991 bedacht en ontwikkeld door Tim Berners-Lee om wetenschappelijke documenten van het CERN in Genève gemakkelijker toegankelijk te maken. Hij ontwikkelde ook de eerste webbrowser, WorldWideWeb genaamd. In 1993 werd de webbrowser Mosaic ontwikkeld door het NCSA, die als eerste grafische elementen in HTML ondersteunde. Op basis daarvan ontstond in 1994 Netscape Navigator. Met de komst van deze beide browsers werd het World Wide Web echt populair. Beide browsers voegden elementen toe aan de oorspronkelijke HTML. Kort daarop mengde Microsoft zich in de strijd met een eigen webbrowser.

# Een beetje geschiedenis (Wikipedia)

---

- ▶ De geschiedenis van HTML ging in de eerste jaren hand in hand met de ontwikkeling van webbrowsers. De belangrijkste spelers daarin waren de concurrerende browsers van Netscape en Microsoft. Met elke uitgebrachte serversversie werd de HTML uitgebreid of aangepast. De organisatie W3C (World Wide Web Consortium) werd opgericht door Tim Berners-Lee, met het doel om standaarden vast te stellen, maar moest regelmatig constateren dat de standaard waaraan gewerkt werd, inmiddels achterhaald was.

# Een beetje geschiedenis (Wikipedia)

---

- ▶ De eerste officiële specificatie van HTML door de W3C organisatie was HTML 2.0, geaccepteerd in september 1995. Eerder was er gewerkt aan uitbreiding van de oorspronkelijke standaard onder de naam HTML+ (gepubliceerd in november 1993). Al vóór de officiële acceptatie van HTML 2.0, werd er de volgende specificatie (HTML 3.0) voorbereid. Deze specificatie werd achterhaald door het uitkomen van Netscape versie 3 en het voorstel werd in september 1995 verlaten. Het werk werd voortgezet in HTML versie 3.2, dat de naam "Wilbur" meekreeg. Wilbur werd geaccepteerd in januari 1997.

# Een beetje geschiedenis (Wikipedia)

---

- ▶ In december 1997 werd de HTML 4.0 specificatie geaccepteerd, waarop nog enkele kleine correcties zijn aangebracht in HTML 4.01 (geaccepteerd in december 1999).
- ▶ De HTML-4.01-specificatie was lange tijd bedoeld als de laatste specificatie van HTML. Verdere ontwikkeling van de markup-taal zou plaatsvinden in de vorm van XHTML (een subset van XML) en in de ontwikkeling van stijlbladen en JavaScript. Deze versie was niet uitwisselbaar met de eerdere HTML

# Een beetje geschiedenis (Wikipedia)

---

- ▶ W3C-directeur Tim Berners-Lee zette echter in een blogposting in oktober 2006 een nieuwe koers uit. Hij constateerde dat een groot deel van de webontwikkelaars en -uitgevers niet bereid was gebleken over te stappen naar X(HT)ML, en dat het daarom nodig was de ontwikkeling van HTML toch voort te zetten. Die ontwikkeling diende volgens Berners-Lee parallel te lopen aan de ontwikkeling van XHTML.

# Een beetje geschiedenis (Wikipedia)

- ▶ Het W3C stelde daarop in maart 2007 een nieuwe HTML Working Group in, die al in juni 2007 een Working Draft diende te kunnen presenteren. Om dit te kunnen realiseren, zou de Working Group nauw samenwerken met de WHATWG-groep (Web HyperText Application Technology Working group), die uit ongenoegen over de koers van het W3C sinds 2004 werkte aan HTML 5 (een upgrade voor HTML4)
- ▶ 2019: WHATWG & W3C komen overeen om samen te werken aan 1 versie van HTML:  
[html.spec.whatwg.org](https://html.spec.whatwg.org)

