

Labo 2

opleidingsonderdeel **Web Development 1**

studiegebied **Handelswetenschappen, bedrijfskunde**

Bachelor in de **toegepaste informatica**

campus **Kortrijk**

Sander Vandeputte

docent: **Pim Debaere**

academiejaar **2025-2026**

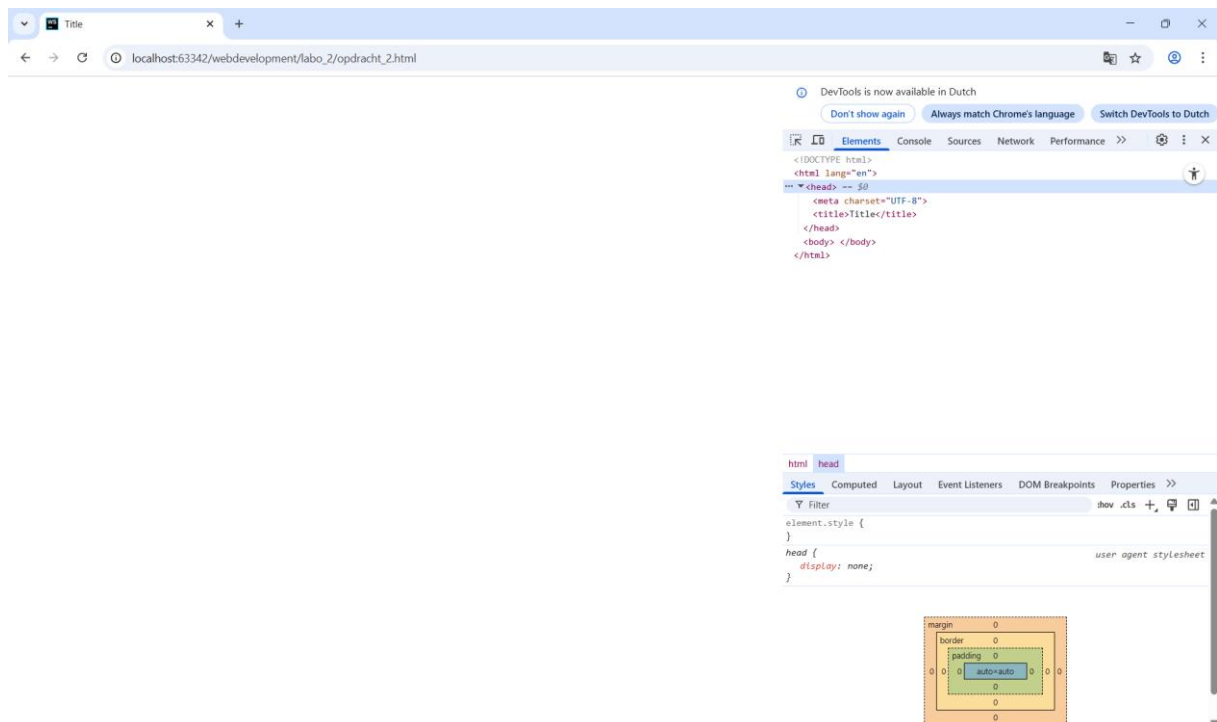
Labo 2 (indienen als PDF)

Opdracht 1 (vergelijk de broncode met de response body in network)

Deze zijn hetzelfde. Het enige verschil is dat de code in de response body niet met die sprongetjes werkt waardoor het er iets minder duidelijk uit ziet.

Opdracht 2 (Maak een nieuw bestand en bekijk in chrome. Wat zie je?)

Helemaal niks want er is nog geen tekst, paragraaf, titel, afbeelding, ...



Opdracht 3

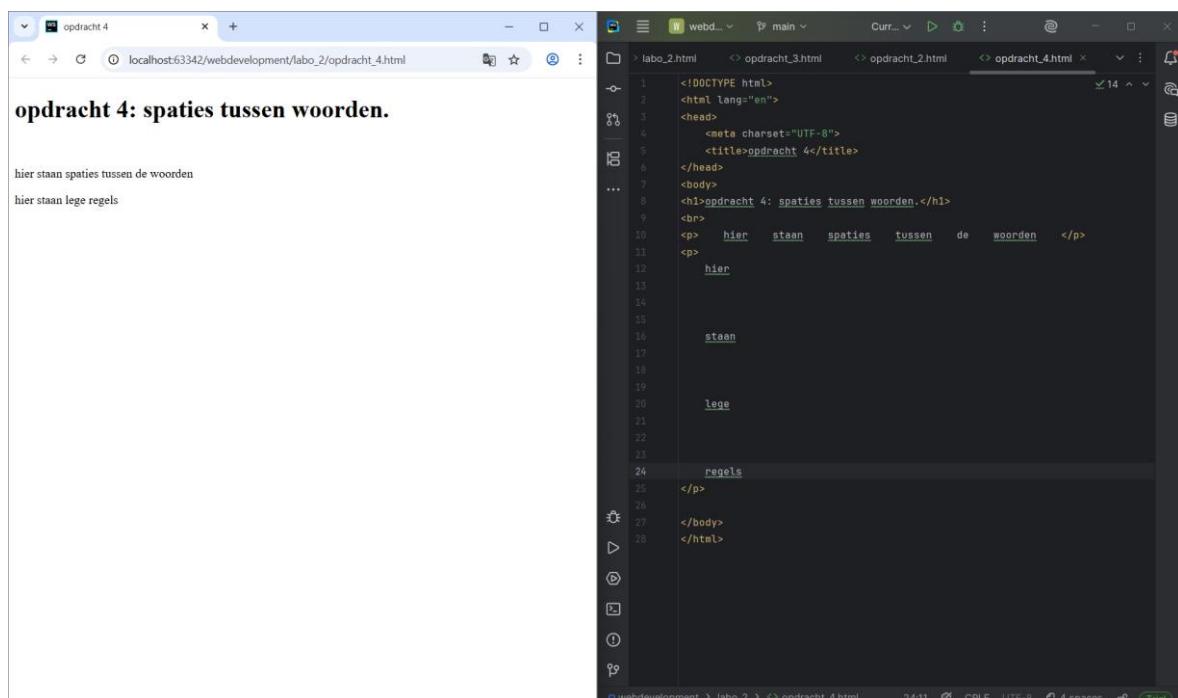
3.1:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   <p> Dit is mijn eerste paragraaf</p>
9   <br>
10  <a href="https://www.vives.be/nl"> link naar de vives website</a>
11  <p>Dit is een eenvoudige paragraaf <a href="https://www.vives.be/nl">met een hyperlink</a> en nog wat meer tekst</p>
12
13 </body>
14 </html>
```

3.2 (verschil tussen file:// protocol (erop klikken in verkenner) en http:// protocol (open in browser via webstorm) op de host 'localhost' op poort 63342 voor het ophalen van een lokale HTML code):

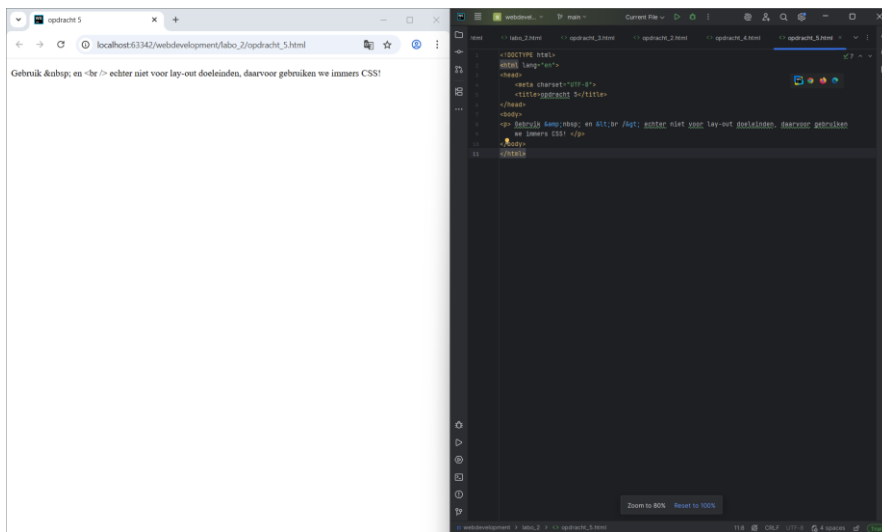
- File://: Het bestand wordt rechtstreeks van de computer geopend. Hier komt geen webserver van te pas. Er is een redelijk grote kans dat sommige functies geblokkeerd worden omwille van veiligheid.
- http://: gebruikt een lokale webserver die door webstorm zelf wordt aangeboden. Het html bestand wordt geladen alsof het een echte website is.

Opdracht 4



Zoals je kunt zien maakt het niet uit hoeveel spaties of lege regels je tussen woorden plaatst. De browser negeert extra witruimte en zet de woorden gewoon achter elkaar.

Opdracht 5 (geef de tekst correct weer door de juiste entity references te gebruiken.):



Bepaalde tekens kunnen niet zomaar in code als tekst getypt worden omdat deze een speciale betekenis hebben in HTML. Het kan zijn dat de browser anders sommige tekens ziet als code terwijl deze bedoeld zijn als tekst.

De gebruikte entity references zijn:

- & voor &-teken
- > voor >-teken
- < voor <-teken

Opdracht 6 (waarom zijn volgende uitdrukkingen niet correct genest?):

• (4x2 + 2(

Geen enkel haakje wordt gesloten

• (4x2(+2

Geen enkel haakje wordt gesloten

• (4x(2+2)

Er zijn 2 geopende en maar 1 gesloten haakje

• (4x(2+2(

Er zijn lege haakjes en er zijn 3 geopende en 1 gesloten haakjes

besluit: elke keer dat er een haakje geopend wordt moet het ook gesloten worden. Hier is dat niet het geval.

Opdracht 7 (zijn de haakjes correct genest?):

- `{() }` Ja. Deze zijn correct genest. Als er een haakje open gaat wordt deze ook weer op het correcte moment gesloten.
- `{{() }` Nee. Deze haakjes zijn niet correct genest. Er zijn 3 accolades en maar 1 vierkant haakje. Geen enkele van de accolades sluit en er is geen geopend vierkant haakje. Dit kan niet.
- `{()})` Ja. Deze zijn correct genest. Elk haakje dat geopend wordt word ook weer gesloten.
- `{()} { }` Nee. Deze haakjes zijn niet correct genest. De volgorde van het openen en sluiten van de haakjes klopt niet. Tussen de accolades staat een geopend vierkant haakje maar dat vierkant haakje wordt pas gesloten nadat de accolade al gesloten is. Dit kan niet.
- `{{(}` Nee. Deze haakjes zijn niet correct genest. Er wordt 2 keer geopend met het accolade maar deze worden nooit gesloten.

Opdracht 8

8.1 (is een spatie net voor het woord "te" eigenlijk nodig?): Ja want anders wordt het linkte in plaats van link te.

8.2 :

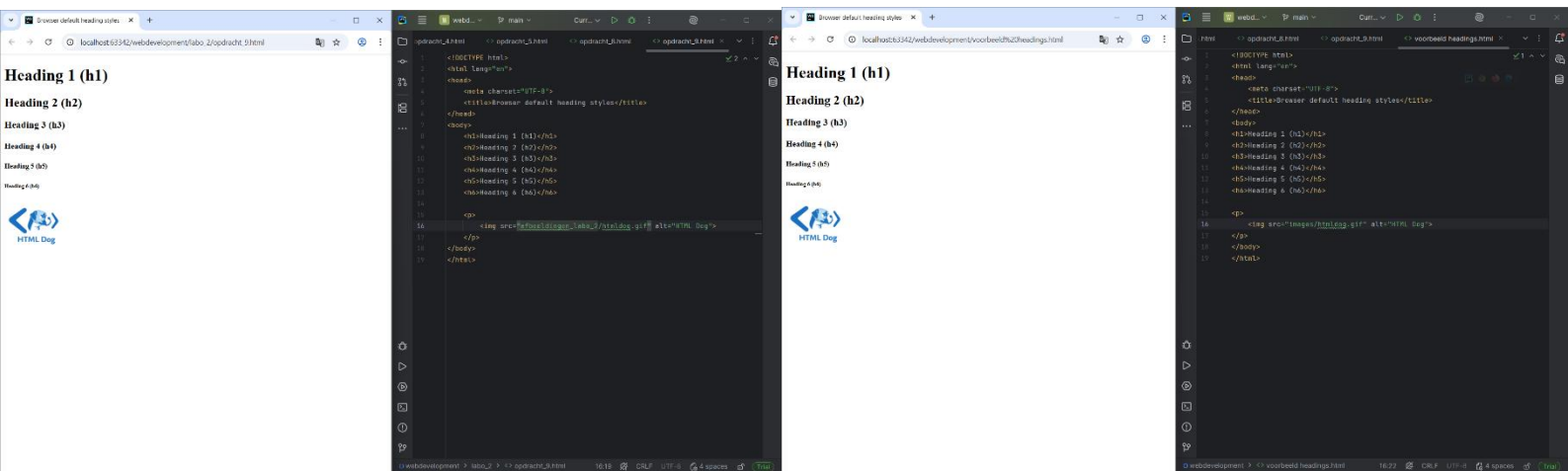
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Title</title>
6  </head>
7  <body>
8
9  <ul>
10   <li><span>hier komt een </span><a href="">link</a><span> te staan</span></li>
11   <br>
12   <li><span>hier komt <strong>een </span><a href="">link</a></strong><span> te staan</span></li>
13   <br>
14   <li><span>hier komt <strong>een </strong></span><a href=""><strong>link</strong></a></strong><span> te staan</span></li>
15
16 </ul>
17
18 </body>
19 </html>
```

De 1^{ste} lijn is hoe het gegeven was. Zonder vetgedrukte woorden.

De 2^{de} lijn is hoe de auteur het verandert had. Dit is niet correct want het is niet correct genest. De strong wordt voor het woord “een” getypt maar er niet na. Hierdoor ondervindt de closing tag van `` problemen.

De 3^{de} lijn is de correcte. Hierbij wordt `` apart voor het woord “een” en voor het woord “link” gedaan waardoor het niet in de problemen komt met de closing tag ``.

Opdracht 9:



9.1(Heb je de waarde van het src attribuut moeten aanpassen?): Ja. Omdat de afbeelding in de map images geplaatst is, moest de waarde van het src attribuut aangepast worden naar **images/html5dog.gif** zodat de browser wist op welke plaats deze moest zoeken.

Opdracht 10 (wat is verkeerd in onderstaande code?):

```
<ul>
  <li>een</li>
  <li>twee</li>
  <ol>
    <li>jan</li>
    <li>piet</li>
    <li>joris</li>
  </ol>
  <li>vier</li>
</ul>
```

De fout is dat de genummerde lijst **** zomaar tussen de opsomming van een ongenummerde lijst **** staat. Als je een lijst in een andere lijst wil zetten, moet je die binnen een **** zetten. Hierbij is ook de verbetering aanwezig aan de rechterkant.

```
<!-- juist -->
<ul>
  <li>een</li>
  <li>twee
    <ol>
      <li>jan</li>
      <li>piet</li>
      <li>joris</li>
    </ol>
  </li>
  <li>vier</li>
</ul>
```

Opdracht 11

(Open nu de Chrome Developer Tools en kijk op het Application tabblad bij 'local storage' en 'cookies' welke data deze website door de browser laat bewaren. Je zult zien dat er per domein waar de webpagina gebruik van maakt, aparte data wordt bijgehouden. Wijzig eens de regio/taal van de pagina (links onderaan de webpagina van Microsoft) en kijk opnieuw naar de cookies.

Zie je waar de gekozen taalinstelling wordt bijgehouden?):

Je kan in het **blauwe balkje** zien welke cookie bijhoudt op welke taal de website staat. In dit geval heet het cookie: “Cascade.AuthSSO”

The image displays two screenshots of the Microsoft website with the Chrome Developer Tools open, specifically the Application tab. The first screenshot shows the 'Cascade.AuthSSO' cookie highlighted in blue. The second screenshot shows the same cookie highlighted in blue, with a blue bar at the bottom of the cookie list indicating the selected language.

Cookie List (First Screenshot):

Name	Value	Domain	Path	Expires	Size	Http	Secure	SameSite	Partitioned	CrossSite	Priority
Cascade.AuthSSO	1	www...	/	2025...	16						Med...

Cookie List (Second Screenshot):

Name	Value	Domain	Path	Expires	Size	Http	Secure	SameSite	Partitioned	CrossSite	Priority
Cascade.AuthSSO	1	www...	/	2025...	16						Med...

Opdracht 12 (Waar precies bewaart de webapplicatie welke toestellen je selecteerde ter vergelijking? Ben je nog andere herkenbare informatie tegengekomen op die beide tabbladen? Indien ja, welke?):

De geselecteerde tv's worden bewaard in cookies onder de naam **LG5_CompareCart**. Naast die informatie ben ik ook tegengekomen waar de taal van de website opgeslagen wordt.

The screenshot shows the LG website's 'OLED-evolutie' section, comparing two models: 'LG OLED evo AI 4K G5 2025 97-inch smart-tv' and 'LG OLED evo AI 4K C5 2025 83-inch smart-tv'. Both have a 4.9 star rating. The browser's developer tools are open, displaying the 'Application' tab. Under 'Cookies', the 'LG5_CompareCart' cookie is selected, showing its value as 'MD07603841%7CMD0751250%7C...'. Other cookies like 'session-id' and 'ubid-main' are also listed.

Opdracht 13 (Waar wordt de winkelwagen van Amazon opgeslagen en hoe worden de gegevens bewaard zonder ingelogd te zijn?):

De inhoud van de winkelwagen wordt niet lokaal opgeslagen maar via locale storage. Amazon gebruikt cookies om de gebruiker te herkennen. Deze cookie bevat een unieke ID. Die cookies kunnen de cookies met de naam **session-id** of **ubid-main** zijn. Elke keer dat ik van de website ging en hem opnieuw bezocht veranderde de ID van deze cookies niet. Ook al ben je niet ingelogd, werkt amazon met die cookie-ID's.

The screenshot shows the Amazon website's 'Kit' section. The browser's developer tools are open, displaying the 'Application' tab. Under 'Cookies', the 'session-id' cookie is selected, showing its value as '132-2401089-5758117'. Other cookies like 'ubid-main' are also listed.