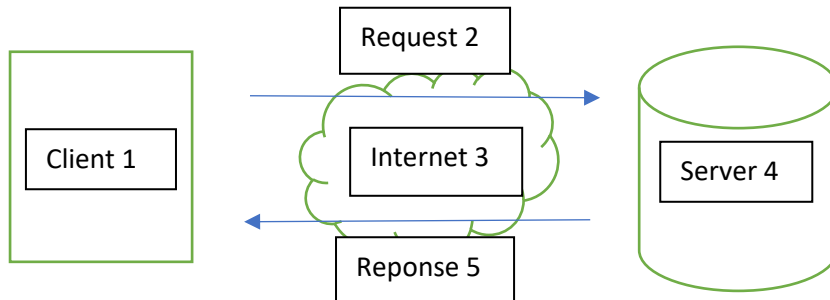


## Web development : Labo 1 - HTML

### Opdracht 1:



**Stap 1:** De client is een apparaat of software die informatie wil ophalen of versturen naar een server. Dus doet hij een request.

**Stap 2:** Request (*verzoek*) is waarbij de client de verzoek stuurt naar de server om specifieke informatie of een bepaalde actie uit te voeren.

**Stap 3:** Dit is de stap die het wereldwijde netwerk van computers en servers die met elkaar verbonden zijn via verschillende protocollen. Zonder Internet zal het niet lukken om van de Client naar de Server te bereiken.

**Stap 4:** De server is een computer of een specifieke softwaretoepassing die de verzoeken van de clients accepteert, verwerkt en antwoorden terugstuurt.

**Stap 5:** De response (*antwoord*) is nadat de server het verzoek heeft verwerkt, genereert het een antwoord dat informatie bevat over de uitkomst van het verzoek.

**Laatste stap** is dat de client het antwoord ontvangt van de server en verwerkt de informatie.

### Opdracht 2:

**https://:** het protocol

**www:** het sub domain

**bol.com:** de host

Poort werd niet vermeld, waarschijnlijk 8080 (https)

**/be/nl/p/hoe%E2%80%90werkt%E2%80%90dat%E2%80%90nou/9200000057347012:** het path

**?suggestionType=browse:** de parameters

**#product\_alternatives:** het fragment

### Opdracht 3:

Heading 1 (h1)

Heading 2 (h2)

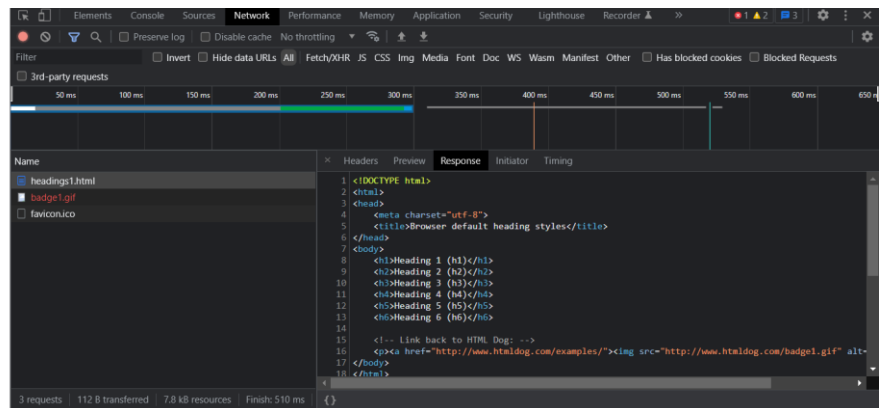
Heading 3 (h3)

Heading 4 (h4)

Heading 5 (h5)

Heading 6 (h6)





**Stap 1:** Open Google Chrome

**Stap 2:** Ga naar de website

**Stap 3:** F12 knop indrukken, daarna inspecteren (hierbij gaat de DevTools open).

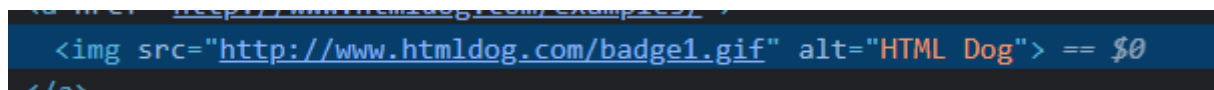
**Stap 4:** In de Chrome Developer Tools klik je op het “Network”-tabblad. (een lijst met alle netwerkverzoeken)

**Stap 5:** klik op verzoek naar “headings1.html” zodat je de details ervan kunt bekijken.

### Opdracht 4:

Naast headings1 werd ook badge1.gif en favicon.ico aangevraagd.

De gif wordt aangevraagd om dat hij gebruikt wordt in het HTML document.



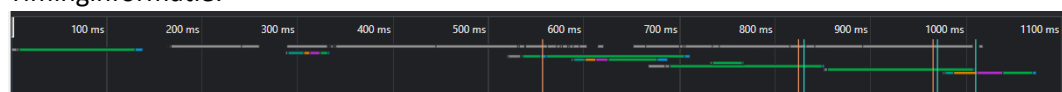
Favicon.ico wordt standaard aangevraagd voor het icoontje op je Chrome tabblad.

### Opdracht 5:

Types opgevraagde data: Jpegs, stylesheets, fonts, svg, scripts, png, xml, xhr, gif, plaintext, ping

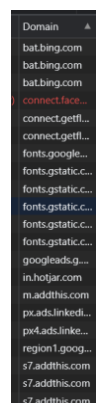
De requests worden van veel verschillende domeinen genomen →

Timinginformatie:



Je ziet dat meerdere requests op hetzelfde moment gebeuren, nog voor het vorige request klaar was. Ze zijn dus onafhankelijk van elkaar en gaat zo veel sneller.

Het is niet zo beperkt, er zijn redelijk veel requests die op hetzelfde moment gebeuren.



**Opdracht 6:** Dit is voor de constante refreshes, zodat men bijna meteen kan weten wanneer er een nieuwe mail toekomt.

**Opdracht 7:** Dat komt doordat de search nergens in de webpagina heeft gevonden op de server.

**Opdracht 8:** Er is geen webserver om je request op te vangen en te verwerken. Je kan dus ook geen error 404 krijgen omdat er geen webserver is die je die error kan geven. Bij Vives was dat er wel.

**Opdracht 9:**

Success	Redirect	Client Error	Server Error
<b>200</b> OK (geen problemen)	<b>301</b> De pagina is permanent verplaatst	<b>400</b> Bad Request, je request is mogelijks verkeerd opgesteld	<b>500</b> Internal Server error, iets is foutgelopen op de server zelf
<b>204</b> No content. De pagina is er wel, maar er is geen inhoud.	<b>302</b> Found	<b>401</b> Unauthorized, je hebt geen toegang tot de pagina	<b>503</b> Service unavailable, de gevraagde service kan niet bereikt worden
	<b>303</b> See other, zie andere pagina	<b>404</b> Pagina kon niet gevonden worden	

De HTTP-statuscodes en hun betekenis kan je opzoeken via online bronnen.

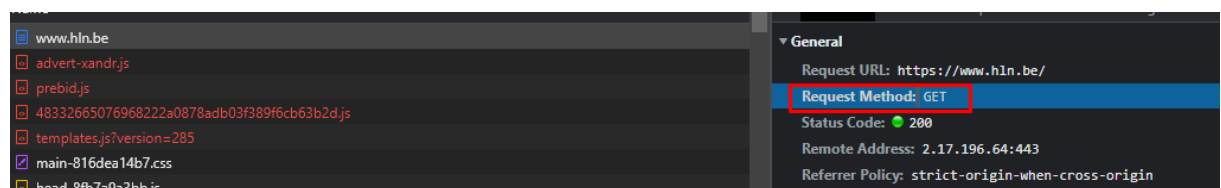
**Opdracht 10:**

- GET – Ontvang het document gespecificeerd door de URL.
- HEAD – Ontvang alleen de headers van het op te vragen document.
- POST – Zend gegevens naar de server.
- PUT – Vervang het document op de server door de verzonden data.
- DELETE – Verwijder het document.
- TRACE – Retourneert de aanvraag zodat een client kan zien welke wijzigingen of aanvullingen zijn gemaakt door tussenstations.
- OPTIONS – Vraag de mogelijkheden op dit niveau aan van de server.
- CONNECT – Vervangt de verbinding door een transparante TCP-/IP-tunnel, om bijvoorbeeld SSL-versleutelde communicatie (HTTPS) via een onversleutelde HTTP proxy te ondersteunen.
- PATCH – Gedeeltelijke modificatie van het docu

De bedoeling van GET is om een webpagina aan te vragen. Als je dus op een link klikt, zal een GET request gestuurd worden.

POST wordt gebruikt om gegevens te sturen naar de webserver, bijvoorbeeld bij een Submit knop. De data wordt verstuurd in de request body van de request.

GET kan ook gegevens doorsturen, maar het komt terecht in de URL als parameter. Als je dus bv. een wachtwoord invult moet je zeker POST gebruiken.



Als Google alle links zou indexeren door ze via een bot uit te testen, zouden producten inderdaad kunnen verdwijnen. Mijn eerste gedachte hierbij is dat de bot altijd GET gebruikt, dus is het misschien een goed idee om bv. de DELETE request te gebruiken om dergelijke “Wis” knoppen te maken.

### Opdracht 11:

63 requests

63 requests	2.1 MB transferred	4.1 MB resources	Finish: 8.70 s	DOMContentLoaded: 2.72 s	Load: 4.28 s
-------------	--------------------	------------------	----------------	--------------------------	--------------

**Stap 1 :** Google Chrome

**Stap 2:** Naar de website

**Stap 3:** Rechtermuisknop op de webpagina “Inspecteren”

**Stap 4:** Naar “Network”-tabblad.

**Stap 5:** Laad de pagina opnieuw op.

**Stap 6:** Je ziet lijst van de netwerkverzoeken gegenereerd terwijl de pagina wordt geladen

### Opdracht 12:

**2,1 MB aan resources verstuurd over netwerk en het duurt 6.07 seconden om de resources te laden.**

63 requests	2.1 MB transferred	4.1 MB resources	Finish: 6.72 s	DOMContentLoaded: 1.76 s	Load: 3.39 s
-------------	--------------------	------------------	----------------	--------------------------	--------------

**Nu 21.2 kB aan data en 1.91 seconden om in te laden. Dit komt omdat er resources gecached zijn. De pagina laadde hierdoor sneller.**

61 requests	21.2 kB transferred	4.1 MB resources	Finish: 1.91 s	DOMContentLoaded: 288 ms	Load: 709 ms
-------------	---------------------	------------------	----------------	--------------------------	--------------

**Bij de response headers zie je “cache-control” staan. Deze voeg je toe aan de kolommen. Soms staat er een “max age” bij. Dit is hoelang het “vers” blijft. Bij andere staat er “no cache”. Dit wil zeggen dat het steeds opnieuw moet inladen.**

cache-control: max-age=1209600
--------------------------------

**Stappen:** gebruik maken van Chrome Developer Tools

Zorg dat de Chrome developer tools actief zijn. Open dan de vives.be pagina, rechtsklik op de refresh knop en kies 'Empty cache and hard reload'. Hiermee dwingen we de browser om bij het herladen van de pagina z'n cache te negeren en alle resources bij de server op te vragen. Hoeveel kilobytes of megabytes aan data werd er verstuurd om alle nodige resources in te laden? Hoe lang duurde het vooraleer alle resources van de pagina waren ingeladen? Klik nu gewoon op de refresh knop. Kijk nogmaals hoeveel data er werd verstuurd. Waarom is dit zoveel minder? Laadde de pagina sneller?

### **Opdracht 13:**

**Nieuwsblad: 11 trackings**

**CNN: 25 trackings**

**Vives: 9 trackings**

**VRT: 12 trackings**

**website analytics**

**Stap 1:** Open Chrome en ga naar de Chrome Web Store.

**Stap 2:** In de Chrome Web Store, zoek naar "Ghostery" in de zoekbalk.

**Stap 3:** Klik op de extensie met de naam "Ghostery - Privacy Ad Blocker" om naar de extensiepagina te gaan.

**Stap 4:** Klik op de knop "Toevoegen aan Chrome" om de extensie te installeren.

**Stap 5:** Bevestig de installatie door op "Extensie toevoegen" te klikken wanneer daarom wordt gevraagd.

**Stap 6:** Na de installatie wordt het Ghostery-pictogram weergegeven in de rechterbovenhoek van de Chrome-browser. Klik erop om de extensie te openen.

**Stap 7:** Volg de setup-instructies van Ghostery om je voorkeuren in te stellen. Je kunt bijvoorbeeld kiezen welke trackers je wilt blokkeren en welke je wilt toestaan.

**Stap 8:** Bezoek de webpagina's die je wilt analyseren voor tracking scripts. Terwijl je een webpagina bezoekt, zal Ghostery tracking scripts detecteren en een lijst van gedetecteerde trackers weergeven in een pop-upvenster.

**Stap 9:** In het pop-upvenster van Ghostery kun je zien hoeveel tracking scripts er zijn gedetecteerd en welke soorten tracking scripts aanwezig zijn (zoals beacons, advertenties, analytics, enzovoort). Je kunt ook kiezen om bepaalde trackers te blokkeren of toe te staan door de selectievakjes aan te vinken of uit te schakelen.

**Stap 10:** Je kunt de informatie van Ghostery gebruiken om inzicht te krijgen in de trackingactiviteiten van de website en beslissen welke trackers je wilt blokkeren of toestaan.

## Opdracht 14:

Er zijn 2 verschillende layouts van de websites.

