GEO1007 Geoweb Technology

Lab 1 & 2

Hongyu Ye (6286240)   Lars van Dijk (5571308)

Lab 1 Web app basics: structure and communication

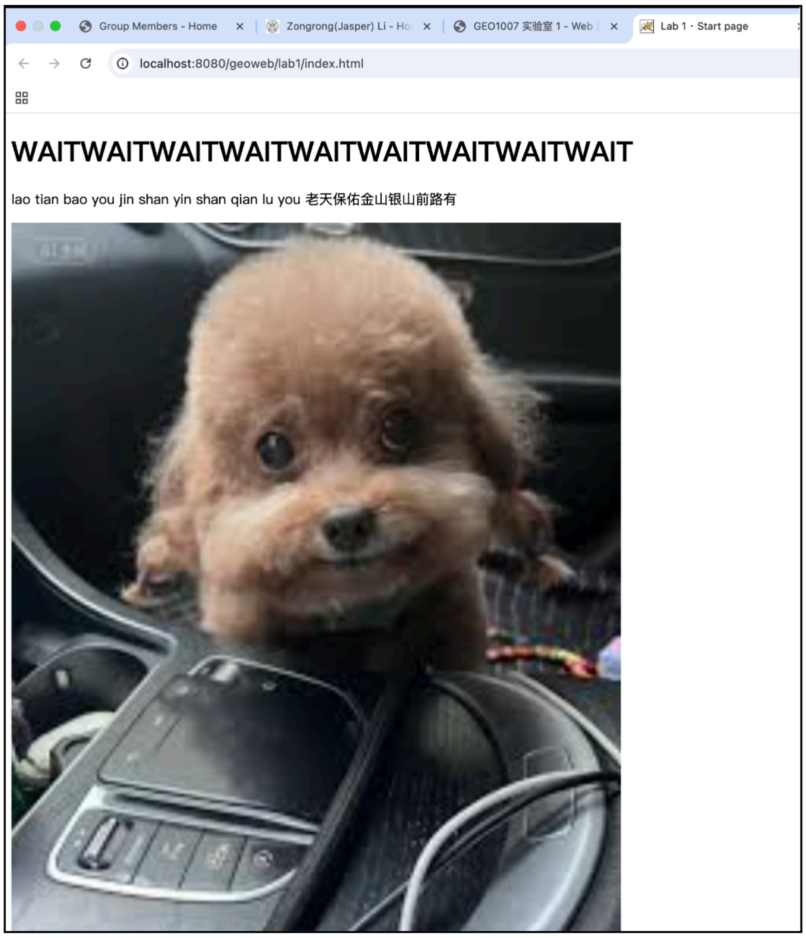
Question 1

a & b

Tag	Function
html	Wraps the entire document; tells the browser "everything inside here is HTML."
head	Contains information *about* the document (metadata) rather than content that renders on the page—things like the title, character set, CSS links, scripts, etc.
title	Sets the text shown in the browser tab, bookmark names, and search-engine results; only one per page.
meta	Provides machine-readable info. In this snippet the charset="utf-8" meta tag tells the browser how to decode the bytes into characters.
body	The visible part of the page. Anything you want a visitor to see or interact with goes inside the body.
h1	The highest-level heading—usually the main title. Search engines treat it as a key signal of what the page is about.
p	Blocks of text separated from others by vertical space. Browsers automatically add some margin before and after.
img	Inserts an image; src points to the file and alt provides substitute text for screen-readers or if the image fails to load.
h3	A third-level heading. It sits two levels below h1 (h2 would be between). Used for sub-sections to create a logical outline.
ol	Displays its children (li items) as a numbered list (1, 2, 3 ...). You can change numbering style with CSS or the type attribute.
li	Represents a single item inside a list (ol or ul). The browser automatically adds marker numbers or bullets.
a	Creates a hyperlink; the href attribute points to the destination URL. Users can click or keyboard-navigate to follow it.
br	Forces a line break inside text flow—similar to hitting "Enter" once in a word processor. Should be used sparingly in semantic HTML.

<!-- ... -->	Adds comments for developers; browsers ignore the content, so it never appears on the page. Handy for notes or temporary code.

c



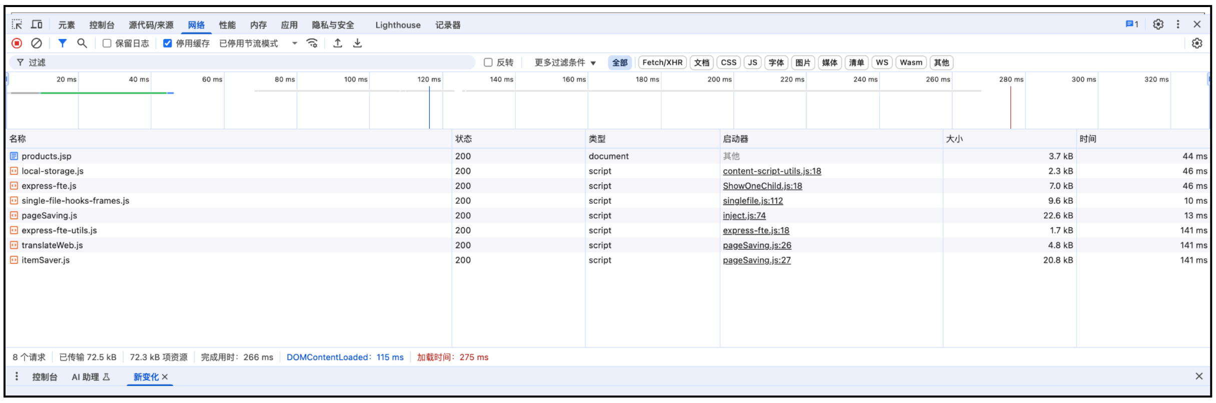
WAITWAITWAITWAITWAITWAITWAITWAITWAIT

lao tian bao you jin shan yin shan qian lu you 老天保佑金山银山前路有

Question 2

a & d

1. **http://localhost:8080/examples/jsp/jsp2/tagfiles/products.jsp**



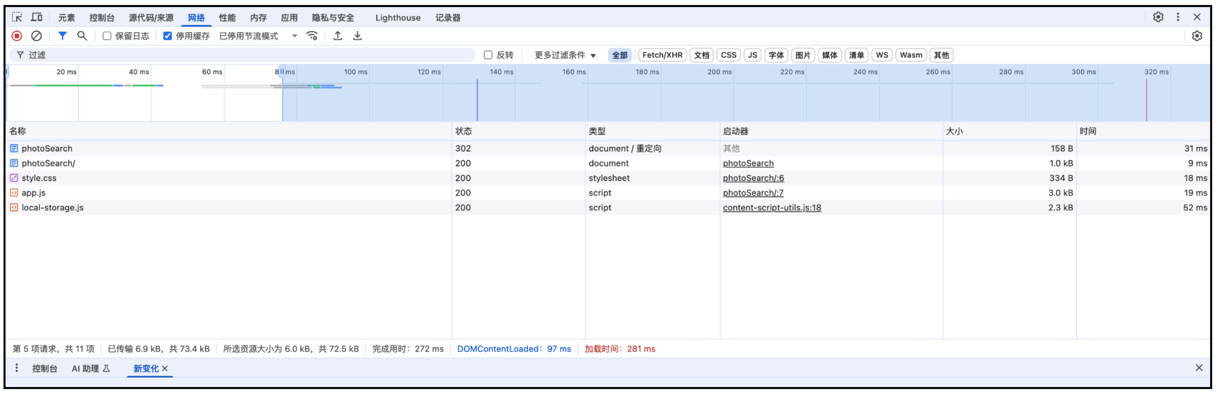
名称	状态	类型	启动器	大小	时间
products.jsp	200	document	其他	3.7 kB	44 ms
local-storage.js	200	script	content-script-utils.js:18	2.3 kB	46 ms
express-fte.js	200	script	ShowOneChild.js:18	7.0 kB	46 ms
single-file-hooks-frames.js	200	script	singlefile.js:112	9.6 kB	10 ms
pageSaving.js	200	script	inject.js:74	22.6 kB	13 ms
express-fte-utils.js	200	script	express-fte.js:18	1.7 kB	141 ms
translateWeb.js	200	script	pageSaving.js:26	4.8 kB	141 ms
itemSaver.js	200	script	pageSaving.js:27	20.8 kB	141 ms

8 个请求 | 已传输 72.5 kB | 72.3 kB 项资源 | 完成用时：266 ms | DOMContentLoaded：115 ms | 加载时间：275 ms

product.jsp

(URL: http://localhost:8080/examples/jsp/jsp2/tagfiles/products.jsp)

2. **http://localhost:8080/geoweb/lab1/photoSearch/**

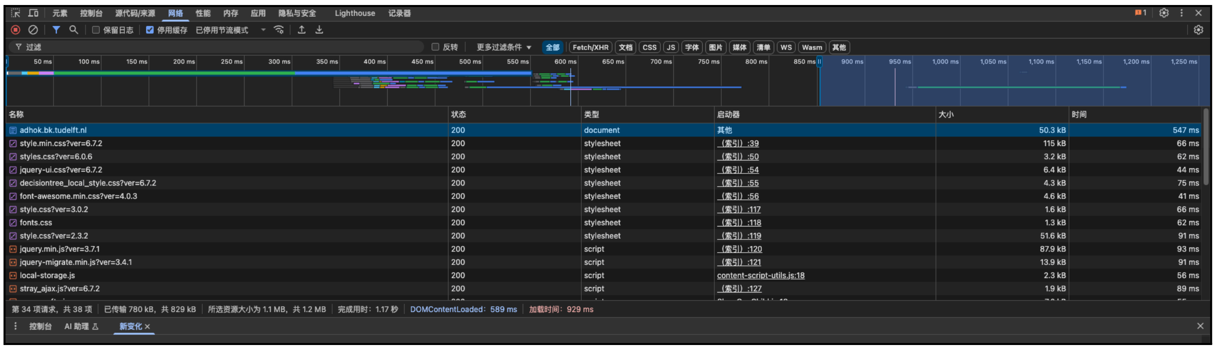


名称	状态	类型	启动器	大小	时间
photoSearch	302	document / 重定向	其他	158 B	31 ms
photoSearch/	200	document	photoSearch	1.0 kB	9 ms
style.css	200	stylesheet	photoSearch/:6	334 B	18 ms
app.js	200	script	photoSearch/:7	3.0 kB	19 ms
local-storage.js	200	script	content-script-utils.js:18	2.3 kB	52 ms

第 5 项请求，共 11 项 | 已传输 6.9 kB，共 73.4 kB | 所选资源大小为 6.0 kB，共 72.5 kB | 完成用时：272 ms | DOMContentLoaded：97 ms | 加载时间：281 ms

photoSearch
(URL: http://localhost:8080/geoweb/lab1/photoSearch)
photoSearch/
(URL: http://localhost:8080/geoweb/lab1/photoSearch/)
style.css
(URL: http://localhost:8080/geoweb/lab1/photoSearch/style.css)

3. **https://adhok.bk.tudelft.nl/**

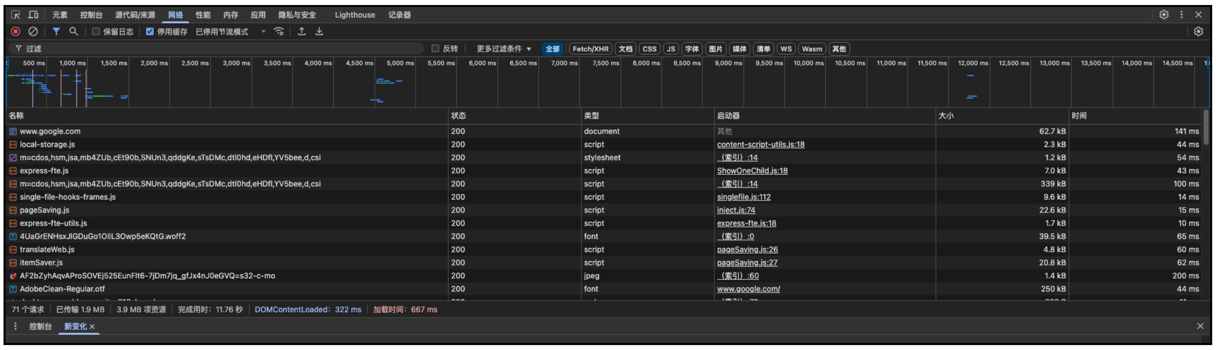


名称	状态	类型	启动器	大小	时间
adhok.bk.tudelft.nl	200	document	其他	50.3 kB	547 ms
style.min.css?ver=6.7.2	200	stylesheet	（索引）:39	115 kB	66 ms
styles.css?ver=6.0.6	200	stylesheet	（索引）:50	3.2 kB	62 ms
jquery-ui.css?ver=6.7.2	200	stylesheet	（索引）:54	6.4 kB	44 ms
decisiontree_local_style.css?ver=6.7.2	200	stylesheet	（索引）:55	4.3 kB	75 ms
font-awesome.min.css?ver=4.0.3	200	stylesheet	（索引）:56	4.6 kB	41 ms
style.css?ver=3.0.2	200	stylesheet	（索引）:117	1.6 kB	66 ms
fonts.css	200	stylesheet	（索引）:118	1.3 kB	62 ms
style.css?ver=2.3.2	200	stylesheet	（索引）:119	51.6 kB	91 ms
jquery.min.js?ver=3.7.1	200	script	（索引）:120	87.9 kB	93 ms
jquery-migrate.min.js?ver=3.4.1	200	script	（索引）:121	13.9 kB	91 ms
local-storage.js	200	script	content-script-utils.js:18	2.3 kB	56 ms
stray_ajax.js?ver=6.7.2	200	script	（索引）:127	1.9 kB	89 ms

第 34 项请求，共 38 项 | 已传输 780 kB，共 829 kB | 所选资源大小为 1.1 MB，共 1.2 MB | 完成用时：1.17 秒 | DOMContentLoaded：589 ms | 加载时间：929 ms

adhok.bk.tudelft.nl (URL: https://adhok.bk.tudelft.nl/)
style.min.css?ver=6.7.2
(URL: https://adhok.bk.tudelft.nl/wp-includes/css/dist/block-library/style.min.css?ver=6.7.2)
styles.css?ver=6.0.6
(URL:
https://adhok.bk.tudelft.nl/wp-content/plugins/contact-form-7/includes/css/styles.css?ver=6.0.6)
jquery-ui.css?ver=6.7.2
(URL: https://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css?ver=6.7.2)
decisiontree_local_style.css?ver=6.7.2
(URL:
https://adhok.bk.tudelft.nl/wp-content/plugins/sidecar-decision-tree/decisiontree_local_style.css?ver=6.7.2)

4. **https://www.google.com/**



名称	状态	类型	启动器	大小	时间
www.google.com	200	document	其他	62.7 kB	141 ms
local-storage.js	200	script	content-script-utils.js:18	2.3 kB	44 ms
m=cdos,hsm,jsa,mb4ZUb,cEt90b,SNUn3,qddgKe,sTsDMc,dtl0hd,eHDfl,YV5bee,d,csi	200	stylesheet	（索引）:14	1.2 kB	54 ms
express-fte.js	200	script	ShowOneChild.js:18	7.0 kB	43 ms
m=cdos,hsm,jsa,mb4ZUb,cEt90b,SNUn3,qddgKe,sTsDMc,dtl0hd,eHDfl,YV5bee,d,csi	200	script	（索引）:14	339 kB	100 ms
single-file-hooks-frames.js	200	script	singlefile.js:112	9.6 kB	14 ms
pageSaving.js	200	script	inject.js:74	22.6 kB	15 ms
express-fte-utils.js	200	script	express-fte.js:18	1.7 kB	10 ms
4UaGrENHsxJlGDuGo1OIlL3Owp5eKQtG.woff2	200	font	（索引）:0	39.5 kB	65 ms
translateWeb.js	200	script	pageSaving.js:26	4.8 kB	60 ms
itemSaver.js	200	script	pageSaving.js:27	20.8 kB	62 ms
AF2bZyhAqvAProSOVEj525EunFlt6-7jDm7jq_gfJx4nJ0eGVQ=s32-c-mo	200	jpeg	（索引）:60	1.4 kB	200 ms
AdobeClean-Regular.otf	200	font	www.google.com/	250 kB	44 ms

71 个请求 | 已传输 1.9 MB | 3.9 MB 项资源 | 完成用时：11.76 秒 | DOMContentLoaded：322 ms | 加载时间：667 ms

www.google.com
(URL: https://www.google.com/)
desktop_searchbox_sprites318_hr.webp
(URL: https://www.google.com/images/searchbox/desktop_searchbox_sprites318_hr.webp)
create
(URL: https://waa-pa.clients6.google.com/$rpc/google.internal.waa.v1.Waa/Create)
GetAsyncData
(URL: https://ogads-pa.clients6.google.com/$rpc/google.internal.onegoogle.asyncdata.v1.AsyncDataService/GetAsyncData)
Ping
(URL: https://waa-pa.clients6.google.com/$rpc/google.internal.waa.v1.Waa/Ping)

b

Type	File	Reponse
document	the main HTML page	It becomes the visible web page.
stylesheet	a CSS file (text/css)	Gives the page its colours, fonts and layout.
script	a JavaScript file (application / javascript)	Adds logic: clicks, data loading, etc.
jpeg	a JPEG picture (image/jpeg)	Normal photo format for images.
webp	a WebP picture (image/webp)	Newer, smaller image format Chrome likes.
font	a web-font file (font/woff2, woff)	Lets the site use special text styles.
xhr	data returned by fetch() / XHR, often JSON	Page asks the server for extra data after it loads.
ping	a tiny beacon request, no body	Used for analytics—just tells the server “a page view happened”.
preflight	CORS OPTIONS check	Browser asks “may I send the real cross-site request?”

c

Code	Name	Meaning
200 OK	Success	The file came back fine.
302 Found	Redirect	“This file moved, go to a new URL.” The browser immediately makes another request to the link in the Location header.

204 No Content	Empty success	Server says "I got your request, nothing to send back." Common for tracking pings or beacons.

e

Similarity for list A and list B link

- Every page begins with one HTML document request and gets back **200 OK**.
- Right after the document, the browser grabs at least one CSS file and several JavaScript files.
- Those same Chrome-extension helper scripts (local-storage.js, express-fte.js, etc.) appear in every list—so both groups show injected resources as script.
- Most responses are successful (status 200); no 4xx or 5xx errors anywhere.

Difference for list A and list B link

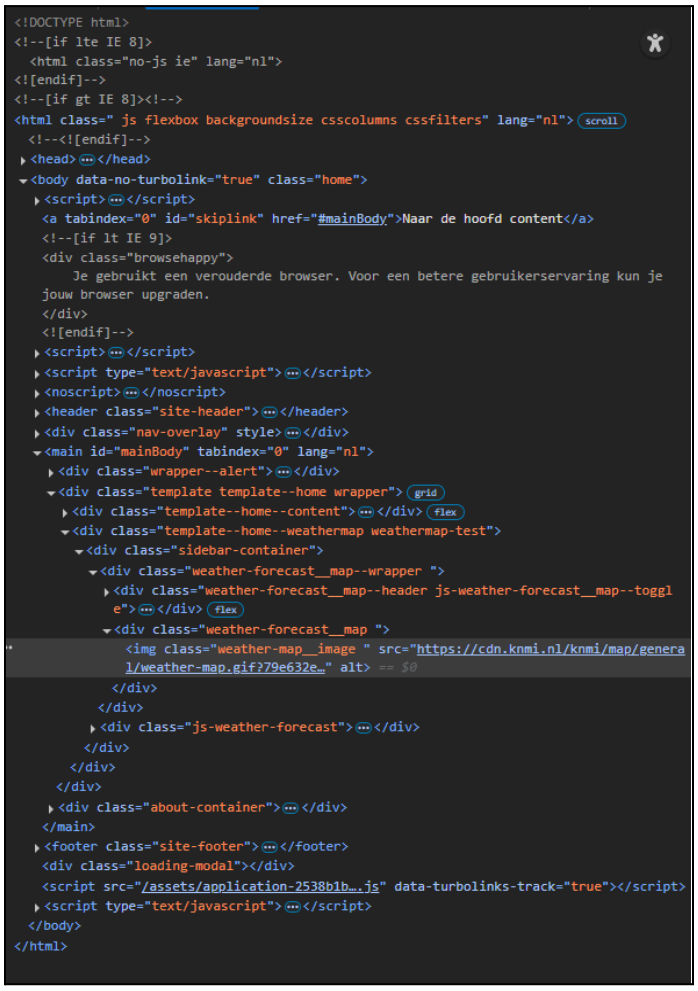
- **Request number:** list A is tiny—only 8–11 lines—while the second group explodes to 30-plus requests.
- **File types:** list A is almost all *document + script* (with a single CSS). List B mixes in lots of extra types—extra stylesheets, web-fonts, PNG/JPEG/WebP images, even a JSONP call and CORS pre-flight.
- **Redirects:** list A has at most one 302 redirect; list B chains 301/302 before the final page or image list.
- **Domains:** list A fetches everything from the local test server plus the extension; list B pulls from remote hosts and CDNs.
- **Load time:** because there is far less to fetch, list A finishes in a couple of hundred milliseconds, whereas list B keeps the network panel busy for a full second or more.

Question 3

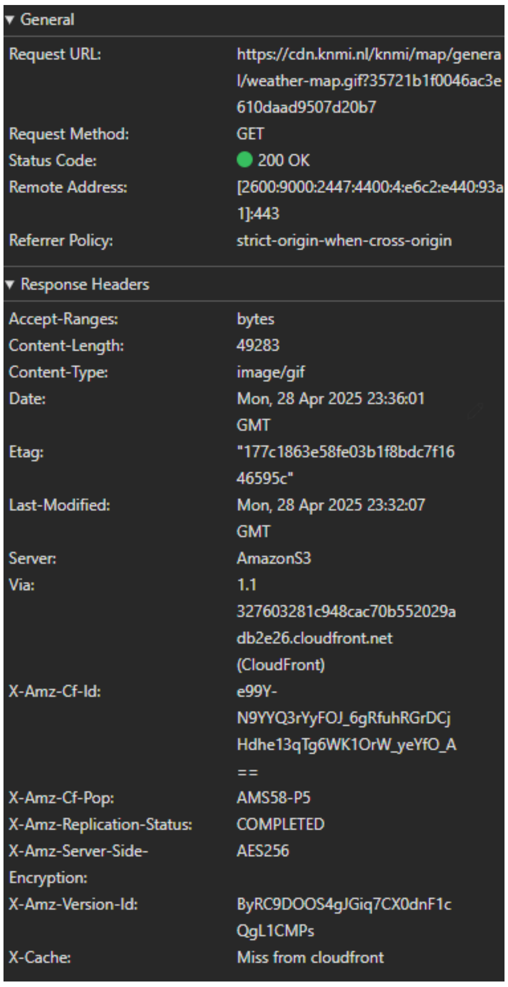
Example: https://www.knmi.nl/home
Using inspect on the picture shows the HTML code using the request method GET to obtain the map.

```
<img class="weather-map__image "
src="https://cdn.knmi.nl/knmi/map/general/weather-map.gif?35721b1f0046ac
3e610daad9507d20b7" alt="">
```

The content type of the map is analysed using the Network tool. The content type of the map is Image/gif.



```
<!DOCTYPE html>
<!--[if lte IE 8]>
  <html class="no-js ie" lang="nl">
<![endif]-->
<!--[if gt IE 8]><!-->
<html class=" js flexbox backgroundsize csscolumns cssfilters" lang="nl"> scroll
  <!--<![endif]-->
 ▸<head>…</head>
 ▾<body data-no-turbolink="true" class="home">
   ▸<script>…</script>
    <a tabindex="0" id="skiplink" href="#mainBody">Naar de hoofd content</a>
    <!--[if lt IE 9]>
    <div class="browsehappy">
        Je gebruikt een verouderde browser. Voor een betere gebruikerservaring kun je
    jouw browser upgraden.
    </div>
    <![endif]-->
   ▸<script>…</script>
   ▸<script type="text/javascript">…</script>
   ▸<noscript>…</noscript>
   ▸<header class="site-header">…</header>
   ▸<div class="nav-overlay" style>…</div>
   ▾<main id="mainBody" tabindex="0" lang="nl">
     ▸<div class="wrapper--alert">…</div>
     ▾<div class="template template--home wrapper"> grid
       ▸<div class="template--home--content">…</div> flex
       ▾<div class="template--home--weathermap weathermap-test">
         ▾<div class="sidebar-container">
           ▾<div class="weather-forecast__map--wrapper ">
             ▸<div class="weather-forecast__map--header js-weather-forecast__map--toggl
              e">…</div> flex
             ▾<div class="weather-forecast__map ">
                <img class="weather-map__image " src="https://cdn.knmi.nl/knmi/map/genera
                l/weather-map.gif?79e632e…" alt> == $0
              </div>
            </div>
           ▸<div class="js-weather-forecast">…</div>
          </div>
        </div>
      </div>
     ▸<div class="about-container">…</div>
    </main>
   ▸<footer class="site-footer">…</footer>
    <div class="loading-modal"></div>
    <script src="/assets/application-2538b1b….js" data-turbolinks-track="true"></script>
   ▸<script type="text/javascript">…</script>
  </body>
</html>
```



```
▾ General
Request URL:              https://cdn.knmi.nl/knmi/map/genera
                          l/weather-map.gif?35721b1f0046ac3e
                          610daad9507d20b7
Request Method:           GET
Status Code:              ● 200 OK
Remote Address:           [2600:9000:2447:4400:4:e6c2:e440:93a
                          1]:443
Referrer Policy:          strict-origin-when-cross-origin
▾ Response Headers
Accept-Ranges:            bytes
Content-Length:           49283
Content-Type:             image/gif
Date:                     Mon, 28 Apr 2025 23:36:01
                          GMT
Etag:                     "177c1863e58fe03b1f8bdc7f16
                          46595c"
Last-Modified:            Mon, 28 Apr 2025 23:32:07
                          GMT
Server:                   AmazonS3
Via:                      1.1
                          327603281c948cac70b552029a
                          db2e26.cloudfront.net
                          (CloudFront)
X-Amz-Cf-Id:              e99Y-
                          N9YYQ3rYyFOJ_6gRfuhRGrDCj
                          Hdhe13qTg6WK1OrW_yeYfO_A
                          ==
X-Amz-Cf-Pop:             AMS58-P5
X-Amz-Replication-Status: COMPLETED
X-Amz-Server-Side-        AES256
Encryption:
X-Amz-Version-Id:         ByRC9DOOS4gJGiq7CX0dnF1c
                          QgL1CMPs
X-Cache:                  Miss from cloudfront
```

Lab 2 - Web pages: Styling and Interactivity

A.1

Type Selectors:
body
header
main
footer

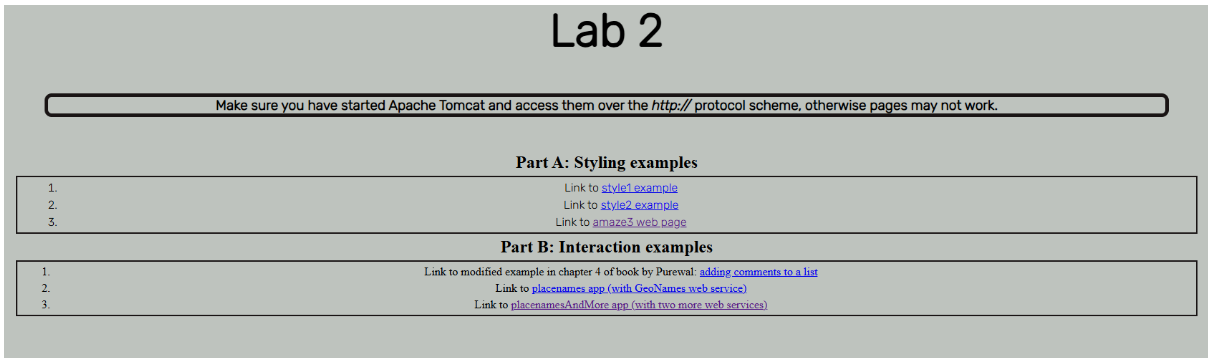
Class selectors:
main .left
header .container

Selector lists:
header .container,
main .container,

footer .container

Descendant combinator:
header h1
header nav

A.2


Lab 2

Make sure you have started Apache Tomcat and access them over the *http://* protocol scheme, otherwise pages may not work.

Part A: Styling examples

1. Link to style1 example
2. Link to style2 example
3. Link to amaze3 web page

Part B: Interaction examples

1. Link to modified example in chapter 4 of book by Purewal: adding comments to a list
2. Link to placenames app (with GeoNames web service)
3. Link to placenamesAndMore app (with two more web services)

B.1

When the webpage is initially loaded HTML is used to define the content and CSS is used to specify the layout and styling. When adding a new TODO Javascript is used to program the behavior of the page. It changes the HTML attribute values or CSS styles. In this instance, First the Javascript obtained the input value and adds it to an element newComment, the Javascript then changes the styling of this comment to hide the newComment, the HTML attribute comments is changed by appending a new child attribute, afterwards the javascript changes the styling of the comment by displaying it again and then the javascript removes the input text from the inputbox.

B.2

a

DOMContentLoaded:

```
document.addEventListener("DOMContentLoaded", allFunctions);
```

keypress:

```
document
  .querySelector("section#geonames input")
  .addEventListener("keypress", function (event) {
    if (event.keyCode === 13) {
      searchFromInput();
    }
  })
```

click:

```
document
 .querySelector("section#geonames button")
 .addEventListener("click", function (event) {
   searchFromInput();
 })
```

b

The keypress and click statements have a querySelector part. In these parts the selectors section#geonames button & section#geonames input are used. Like in CSS these selectors target specific html code. section#geonames button, targets the button in the section with ID geonames. The querySelector is used to target a specific part of the webpage to then check for the event, in this case a click.

c

When the click or keypress events occur the javascript reacts by carrying out the *searchFromInput();* function. This function takes the text in the input

d

The function *getPlacenames_plain_javascript(postalcodeInput, countryInput);* is used to construct the web url to get the place name for a postal code.

e

When the GeoNames server sends back its JSON, the script runs:

1. ***fetch()* finishes → JSON is parsed**

   getPlacenames_plain_javascript() gets the response, checks response.ok, then calls *response.json()*.

   The parsed object is passed to *createTableFromJsonResponse()*.
2. **Debug copy of the raw JSON**

   Inside *createTableFromJsonResponse()* the whole object is string-ified and appended to

   <main> .forDebug — so you can see exactly what GeoNames sent.
3. **Check if the object contains postal code**

```
if (typeof data.postalcodes !== "undefined" && data.postalcodes.length >
0)
```

   Yes → build table rows.

   No → write “no results found” into *<main> .messages* and stop.

4. **Building rows (only if there are results)**

   For every element in data.postalcodes:

   - Create a hidden *<tr>*.
   - Make three *<td>* cells: place name, latitude, longitude.
     - Each cell also gets a data-col attribute (placeName, latitude, longitude).

   - Append the cells to the row, and the row to the `<table id="resultsTable">`.
   - Finally set `row.style.display = ""` so the row becomes visible.
5. **If something wrong with the request**
   Any network or HTTP error is caught by `.catch()`.
   It appends the text “connection error” to `<main> .messages`.

B.3

Similarities

- **Both call a GeoNames service from the browser.**
  One looks up a postal-code list, the other finds the nearest street intersection.
- **Both are fully client-side.**
  They use an HTTP GET, wait for the answer, then update the page without reload.
- **Both show raw data in a “forDebug”**
  Each service response is first dumped (JSON string or plain XML text) so the developer can inspect it, then parsed and turned into <tr> rows that are appended to an existing table.
- **Both can handle empty results and network errors.**
  If the service returns no match the script prints a “no results found” message; if the fetch/XMLHttpRequest fails it shows “connection error”.

Differences

Aspect	Postal-code service	Nearest-intersection service
Transport helper	Uses modern `fetch()` with Promises.	Uses the older `XMLHttpRequest` object.
Data format	Server returns JSON; parsed with `response.json()`.	Server returns XML; parsed via `this.responseXML`.
Parsing logic	Loops over `data.postalcodes` array; each element already a plain JS object with properties (`placeName, lat, lng`).	Looks for the first `<intersection>` element and then loops over its *child nodes*, extracting `nodeName` and text content manually.
Row creation	Builds every cell as a separate `<td>` element with `createElement`, then appends them.	Builds the whole row at once with an `innerHTML` string containing two `<td>` elements (property / value).
Debug output	Dumps `JSON.stringify(data)` into `.forDebug`.	Dumps raw XML text into `.forDebug2` inside a `<textarea>` for readability.
Header row	The postal-code table already exists in the HTML template (only data rows are added).	The XML table creates its own header row programmatically: `<th>Property name</th><th>value</th>`.

B.4

1. **Remove the old map**

```
var getAndDisplayMap = function (wms_request) {
  document.querySelector("main .mapDiv").innerHTML = "";   // NEW

  var img = document.createElement("img");
  img.style.display = "none";
  img.src = wms_request;
  document.querySelector("main .mapDiv").append(img);
  img.style.display = "block";
};
```

Clearing the container (*innerHTML = “”*) delect any *<img>* that was previously added

2. **Remove the old XML dump and intersection table**

```
var anotherGeonamesRequest = function (latitude, longitude) {
  ...
  request.onload = function () {
    if (this.status >= 200 && this.status < 400) {

      document.querySelector("main .forDebug2").textContent = "";   //
NEW
      document.querySelector("#xmlDataAsTable").innerHTML = "";    //
NEW

      var textarea = document.createElement("textarea");
      textarea.rows = "20";
      textarea.cols = "60";
      textarea.style.border = "solid 1px black";
      textarea.textContent = this.responseText;
      document.querySelector("main .forDebug2").append(textarea);

      var xmlData = this.responseXML;
      handleXMLResponse(xmlData);
    } else {
      ...
    }
  };
  ...
};
```

textContext = “” clears the raw-XML area (*.forDebug*)
innerHTML = “” empties the intersection table, letting *handleXMLResponse()* writes a fresh header and new rows.