* **Explain about the Object Model, and why it’s (very) relevant for modern Web-development**

The DOM represents the document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. By using DOM methods and changing DOM properties you can dynamically change a page.

* **Explain (using an example of your own choice) about JavaScript events, and Event Bubbling**

Event propagation – imagine you have a map with a lot of countries. Instead of performing an event handler, such as onclick, on each element, you can assign the event handler to the parent element and then use element.target to reference the child target.

* **Elaborate on how JSON or XML supports communication between subsystems, even when the subsystems are implemented on diﬀerent platforms.**

It is sent through AJAX.

* **Explain the topic AJAX and how it has changed the way modern web-applications are created**

AJAX uses an XMLHttpRequest object to request data from a (different) web server asynchronously, behind the scenes. The front end can fetch data as JSON, plain text or XML from a server without reloading the entire page, making single-page applications possible. Before, an entire document was sent from the server to the browser each time data was requested. Now only a small piece of data is sent.

1. An event occurs in a web page (the page is loaded, a button is clicked)

2. An XMLHttpRequest object is created by JavaScript

3. The XMLHttpRequest object sends a request to a web server

4. The server processes the request

5. The server sends a response back to the web page

6. The response is read by JavaScript

7. Proper action (like page update) is performed by JavaScript

* **Explain the Same Origin Policy (for AJAX), and different ways to work around it**

The same-origin policy is a critical security mechanism that restricts how a document or script loaded from one origin can interact with a resource from another origin, to prevent malicious script from gaining sensitive information. A way to work around it is using CORS.

CORS: Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to tell browsers to give a web application running at one origin, access to selected resources from a different origin.

1. **For the previous task it was possible to obtain data right from *restcountries.eu* via an AJAX call made from within your Browser (as sketched to the right). Use Chrome Developer tools to explain (with focus on the Same Origin Policy) why this is possible.**

This is possible because of the request- and response headers:

The Access-Control-Allow-Headers response header is used in response to **a preflight** request which includes the Access-Control-Request-Headers to indicate which HTTP headers can be used during the actual request.

**A CORS preflight request** is a CORS request that checks to see if the CORS protocol is understood and a server is aware using specific methods and headers.

It is an OPTIONS request, using three HTTP request headers: Access-Control-Request-Method, Access-Control-Request-Headers, and the Origin header.

The Access-Control-Allow-Origin response header indicates whether the response can be shared with requesting code from the given origin.