JavaScript, DOM, AJAX, CORS and SVG

General part

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

- The DOM (Document Object Model) is an interface that represents how your HTML and XML documents are read by the browser. It allows a language (JavaScript) to manipulate, structure, and style your website. After the browser reads your HTML document, it creates a representational tree called the Document Object Model and defines how that tree can be accessed.

- By manipulating the DOM, you have infinite possibilities. You can create applications that update the data of the page without needing a refresh. Also, you can create applications that are customizable by the user and then change the layout of the page without a refresh. You can drag, move, and delete elements.

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

- Javascript events

  JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc. Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable. Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

- Event bubbling

  Event bubbling relates to the order in which event handlers are called when one element is nested inside a second element, and both elements have registered a listener for the same event (a click, for example).

```
document.getElementById("a").onclick = hiFrom;
document.getElementById("b").onclick = hiFrom;

function hiFrom(event)
{
   var id = event.target.id;
   console.log("Hi from " + id);
   document.getElementById("txt").innerText = "Hi from " + id;
```

- }

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

- JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.
- Json and XML allows us to overcome cross-domain issues

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

- AJAX stands for Asynchronous JavaScript And XML and is not a programming language, instead it uses a combination of: A browser built-in XMLHttpRequest object (to request data from a web server), and JavaScript and HTML DOM (to display or use the data).
  AJAX makes it possible to:
    a. Update a web page without reloading the page
    b. Request data from a server - after the page has loaded
    c. Receive data from a server - after the page has loaded
    d. Send data to a server - in the background

- Technology has always been in a state of continuous evolution. Today we can see websites for everything. As we keep on adding stuff to the websites, we keep on making them slower. AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
    e.

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

- Same Origin Policy, also called Single Origin Policy, is a security measure used in Web browser programming languages such as JavaScript and Ajax to protect the condifentiality of information. Same Origin Policy prevents a websites´s scripts from accessing and interacting with scripts used on other sites

- CORS is a security mechanism that allows a web page from one domain or *Origin* to access a resource with a different domain (a *cross-domain request*). CORS is a relaxation of the same-origin policy implemented in modern browsers. Without features like CORS, websites are restricted to accessing resources from the same origin through what is known as same-origin policy.

Practical part

- **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.**

    - It is possible because both applications have the same origin (**Origin:** http://localhost:3456), and because the Access control allows GET method. This is all includes in the response and request headers