

JavaScript, DOM, AJAX, CORS and SVG

General part

1) Explain about the Document 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.
- The importance of JavaScript and DOM as a web technology can be determined from the fact that it is currently used by 94.5% of all websites. As a client-side programming language, JavaScript helps web developers to make web pages dynamic and interactive by implementing custom client-side scripts.

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 page loads, it is called an event. 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).

- Example of Event Bubbling:

```
9  document.getElementById("East").onclick = hearts;
10 document.getElementById("West").onclick = hearts;
11 document.getElementById("South").onclick = hearts;
12 document.getElementById("North").onclick = hearts;
13
24
25 function hearts(event)
26 {
27     document.getElementById("heart").innerHTML = event.currentTarget.id;
28 }
29
```

The concept of event bubbling was introduced to deal with situations where a single event, such as a mouse click, may be handled by two or more event handlers defined at different levels of the DOM hierarchy. If this is the case, the event bubbling process starts by executing the event handler defined for individual elements at the lowest level (on the picture above, a button). From there, the event “bubbles up” to the containing elements (on the picture above the function), then up to even higher-level elements (the body element of the page). Finally, the event ends up being handled at the highest level in the DOM hierarchy, the document element itself.

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:
 - Update a web page without reloading the page
 - Request data from a server - after the page has loaded
 - Receive data from a server - after the page has loaded
 - 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.

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

- Same Origin Policy (SOP), also called Single Origin Policy, is a security measure used in Web browser programming languages such as JavaScript and Ajax to protect the confidentiality and integrity of information. Same Origin Policy prevents a web site's scripts from accessing and interacting with scripts used on other sites.

- **How to work around it:** CORS is a security mechanism that allows a web page from 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

2. **Use Chrome Developer tools to explain (with focus on the Same Origin Policy) why this is possible to obtain data right from *restcountries.eu* via an AJAX call made from within your Browser**
 - It is possible because both applications have the same “origin”:
http://localhost:3456 and because of “Access-Control-Allow-Method”s: GET.
 - This is included in response and request headers.