JavaScript, DOM, AJAX, CORS and SVG

General part

- **Question:** Explain about the Object Model, and why it's (very) relevant for modern Web-development
  Answer: DOM also known as document object model is very relevant to frontend programming as we are able to manipulate HTML tags. The whole idea about working with DOM is to treat the page as an object and therefore change its properties. Also when a client makes a request we can simply change the specific value rather than having to make a response with the whole HTML page again. This makes the performance faster.

- **Question:** Explain (using an example of your own choice) about JavaScript events, and Event Bubbling
  Answer: Generally speaking event bubbling is a term used when we do event handling in JavaScript. An event could be things like 'click', 'mouseover' and so on.
  With an event we attach an id, which we have set on a HTML tag, example:
  .
  By doing so we can sniff out a click on the tag and the event bubbling begins. As a standard it goes from the inner must tag and bubbles out to the HTML tags parents. Meaning if you click on an anchor tag:

  that tag has a parent, perhaps a and so on until we reach .window (the window object represents an open window in a browser) . Which is the equivalent to Object in java language.
  Example from our own code follows below:

```
var outer = document.getElementById("outer");
          outer.onclick = function(e) {
              var target = e.target;
              document.getElementById("text").innerHTML = "You pressed: " +
target.id;
          };
```

```
<div id="outer">
            <div id="id1" class="mydiv"></div>
            <div id="id2" class="mydiv"></div>
            <div id="id3" class="mydiv"></div>
            <div id="id4" class="mydiv"></div>
            <div id="id5" class="mydiv"></div>
            <div id="id6" class="mydiv"></div>
            <div id="id7" class="mydiv"></div>
            <div id="id8" class="mydiv"></div>
            <div id="id9" class="mydiv"></div>
```

```
            <div id="id10" class="mydiv"></div>
        </div>
<p id="text"></p>
```

- **Question:** Elaborate on how JSON or XML supports communication between subsystems, even when the subsystems are implemented on different platforms.
  Answer: When we use JSON or XML it is more readable and the structure is straight forward. This makes it easy to work with as a programmer. This can also help when working across platforms and different systems since it is all in a string form. The string can be in a form of an object, array or just a string value. During the time we've been working with XML, our tomcat server had to read which java file was a servlet. We've now switched to JSON when working with Javascript in correlation with java.

- **Question:**Explain the topic AJAX and how it has changed the way modern web-applications are created
  Answer: Ajax (Asynchronous Javascript And XML) is a web development technique designed to make web applications more interactive. This is done by exchanging small amounts of data between the client and the server, so that the entire site doesn't have to be reloaded every time the user makes a request.

  With ajax, an application can send and retrieve data from a server, without interfering with the display and behavior of the existing page. This allows a web page to change content dynamically without reloading the page.

  Ajax is not a new technology, but a new way of utilizing already existing technologies.

  From early to mid 1990, most web sites were based on HTML pages. Any user action required a complete new page to be loaded from the server, making it highly inefficient.

  In 1996 the iframe tag was introduced, making loading or fetching content asynchronously possible
  In 1998 Microsoft developed the concept behind XMLHttpRequest(the precursor to fetch()), but was barely seen until Outlook web access (2000) and Oddpost(2002) was released. But it wasn't til 2005 that the term "Ajax" was first used
  In 2017 the fetch() function was implemented and mostly replaced XMLHttpRequest.

  Since the first mention of AJAX, there have been several changes to the technologies used in ajax applications. One of these is that XML is no longer required for data interchange and XSLT is no longer required for manipulation of data. Instead an alternative (JSON) is often used for data interchange.

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

  Answer: 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. It helps isolate potentially malicious documents, reducing possible attack vectors.
  A way to work around it is by using Cross-origin Resource Sharing.

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. A web application executes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, or port) from its own.