FAKULTAS
ILMU
KOMPUTER

Web Interactivity

Tim Dosen PBP

Interactivity with JavaScript

Client Side Programming

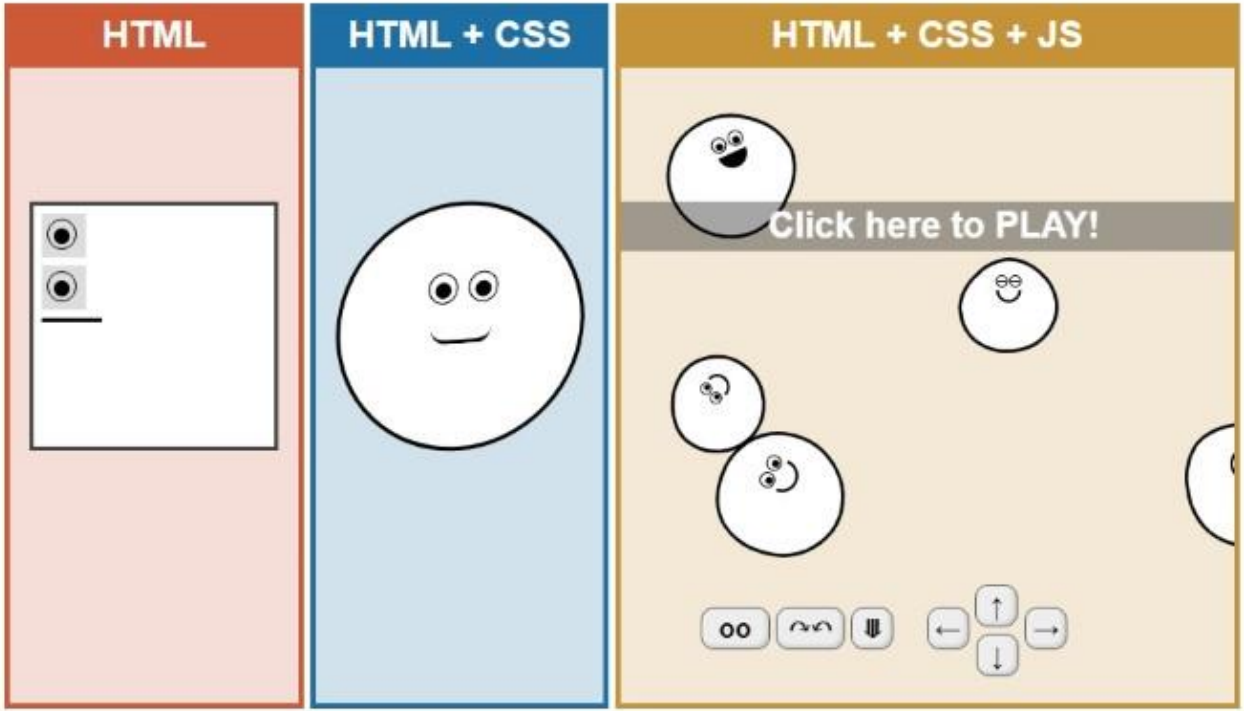


Image from: https://miro.medium.com/max/700/1*n36jCJi0bg3dBeXLogViXg.jpeg

JavaScript

```
<!DOCTYPE html>
<html>
<body>
<h2>What Can JavaScript Do?</h2>
<p>JavaScript can change HTML attribute values.</p>
<p>In this case JavaScript changes the value of the src (source) attribute of an image.</p>
<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button>
<img id="myImage" src="pic_bulboff.gif" style="width:100px">
<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the light</button>
</body>
</html>
```

Motivation

- **usability**: can modify a page without having to post back to the server (faster UI)
- **efficiency**: can make small, quick changes to page without waiting for server
- **event-driven**: can respond to user actions like clicks and key presses

JavaScript

- JavaScript is the programming language of HTML and the Web
- Used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - react to events (ex: page load user click)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)
- Technically, it has no relation with Java programming language

Check more tutorial here: https://www.w3schools.com/js

What JavaScript can do for your Website

By using JavaScript, you can:

- **make your website easy to navigate**. e.g: expandable site maps, tooltips, mouse rollover effects, helpful suggestions pop up, displaying a thank you message, loading content into multiple frames etc
- **customize the way your website looks on the fly.** The customization of your website looks maybe based on criteria such as: the browser, the current date/time, user's behaviours, user's preferences, etc
- **create simple (but effective) animated effects.**

How to Code JavaScript

1. Internal JavaScript

- JavaScript can be placed in the <body> and the <head> sections of an HTML page.
- Use **<script>** tag

```
<script>
    document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

How to Code JavaScript

2. External JavaScript

- Code your JavaScript in external files, save with **.js** extension
- You can place an external script reference in <head> or <body> as you like
- The script will behave as if it was located exactly where the <script> tag is located.
- Use **src** attribute to link your JavaScript files.

  Example:

```
<script src="myScript.js"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
```

- Best practice: External JavaScript

How to Display Outputs

- Writing into an alert box, using **window.alert()**.
- Writing into the HTML output using **document.write()**.
- Writing into an HTML element, using **innerHTML**.
- Writing into the browser console, using **console.log()**.

More: https://www.w3schools.com/js/js_output.asp

JavaScript Variables and Data Types

```
var name = expression;
var age = 32;
var weight = 127.4;
var clientName = “Alice”;
```

- Variables are declared with **var** keyword
- JavaScript has dynamic type
- Data type are: number, string, array, object, boolean, etc.

More: https://www.w3schools.com/js/js_datatypes.asp

JavaScript Function

```
function name() {
  statement ;
  statement ;
  ...
  statement ;
}

function myFunction() {
  window.alert("Hello!");
  window.alert("How are you?");
}
```

JavaScript Statement and Comments

- JavaScript statements are separated by **semicolons**
- JavaScript is Case Sensitive
- The first character of identifiers (name of variables, keywords, functions, and labels) must be a letter, an underscore (_), or a dollar sign ($)
- To make comment, use

  //this is comment

  or

  /* this is comment */

JavaScript Array

- You can have different objects in one array
- Array properties : length
- Array methods: sort()
- Adding array

```
var x = cars.length;
cars
var y = cars.sort();
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Lemon");                 // adds a new element (Lemon) to fruits
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[fruits.length] = "Lemon";      // adds a new element (Lemon) to fruits
```

JavaScript Object

- How to create object

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

```
var person = {
    firstName: "John",
    lastName : "Doe",
    id       : 5566,
    fullName : function() {
       return this.firstName + " " + this.lastName;
    }
};
```

properties

methods are stored in properties as **function definitions**.

JavaScript Object

- How to access object properties

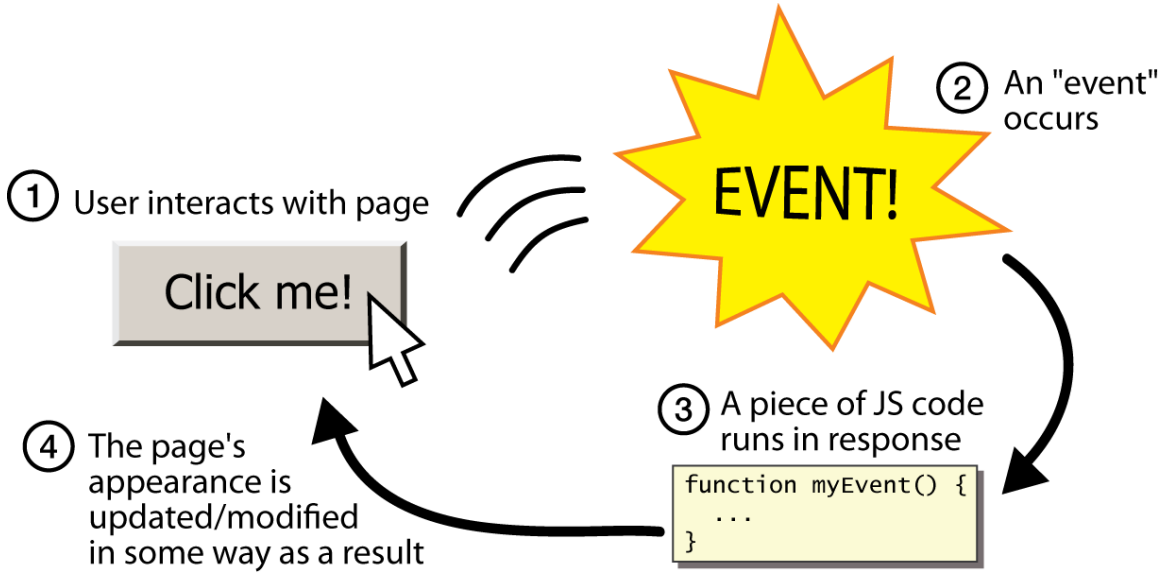
```
objectName.propertyName
```

```
objectName["propertyName"]
```

- How to access object methods

```
objectName.methodName()
```

Event Driven Programming



Source: https://courses.cs.washington.edu/courses/cse190m/11sp/lectures/slides/lecture12-javascript.shtml#slide11

Event Handler

```
<element attributes onclick="function();">...
```

```
<button onclick="myFunction();">Click me!</button>
```

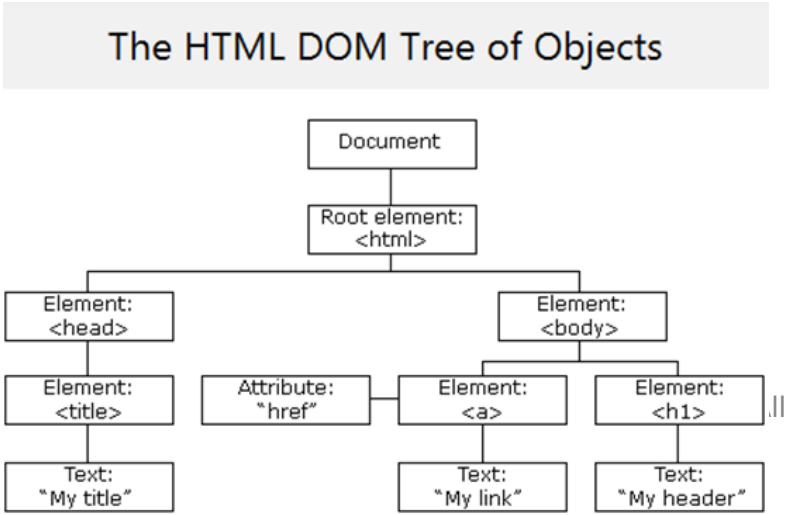
- `onclick` is just one of many event HTML attributes we'll use

More event HTML attributes:

https://www.w3schools.com/tags/ref_eventattributes.asp

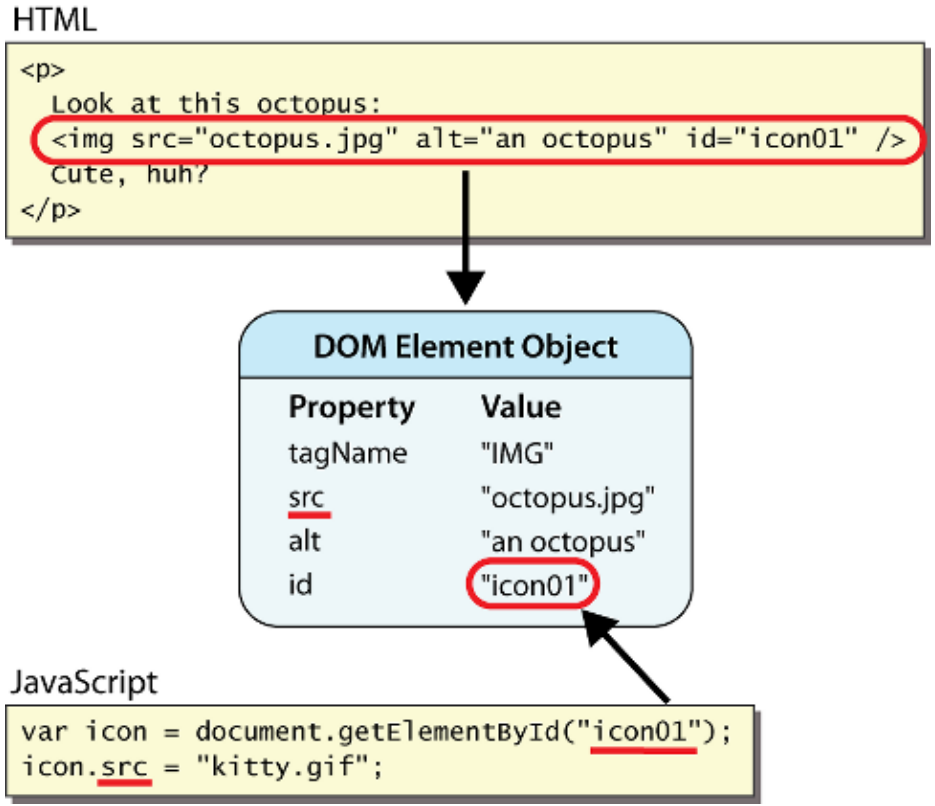
Document Object Model (DOM)

- With the HTML DOM, JavaScript can access and change all the elements of an HTML document.



- http://www.w3schools.com/jsref/default.asp

DOM Element Objects



HTML DOM

- Finding element

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

Method	Description
document.getElementById()	Find an element by element id
document.getElementsByTagName()	Find elements by tag name
document.getElementsByClassName()	Find elements by class name

- Changing element

Method	Description
element.innerHTML=	Change the inner HTML of an element
element.attribute=	Change the attribute of an HTML element
element.setAttribute(*attribute,value*)	Change the attribute of an HTML element
element.style.*property*=	Change the style of an HTML element

HTML DOM

- Create element

Method	Description
document.createElement()	Create an HTML element
document.removeChild()	Remove an HTML element
document.appendChild()	Add an HTML element
document.replaceChild()	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

- Changing style

```
document.getElementById(id).style.property=new style
```

HTML

```
<div>
    <h1 style = "text-align: center">Hello World</h1>
    <p style = "text-align: center"> Welcome to Java Script World</p>
</div>
<div id = "kotaklang">
   <table>
    <tr>
        <th>Java Script</th>
        <th>HTML</th>
        <th>Django</th>
    </tr>

    <tr>
         <td><img id ="jscript"
             src = "https://www.yudana.id/wp-content/uploads/2018/04/javascript-788x394.jpg" height=30% width=50%></td>
        <td><img id ="html"
             src = "https://cdn-images-1.medium.com/max/1096/1*28-1lYrYTQoLhi87mllgBw.png" height=40% width=60%></td>
        <td><img id ="django"
             src = "https://cdn-images-1.medium.com/max/1600/1*u_Jr6FozmyMCi3pe9ZsoFg.png" height=30% width=50%></td>
    </tr>
   </table>
</div>
```

JS Functions

```
function ganti1(){
     var js1 = document.getElementById('jscript');
     js1.src =
'https://cdn1.macworld.co.uk/cmsdata/features/3640222/learn_javascript_on_mac_thumb800.jpg';
}

function ganti2(){
      var html1 = document.getElementById('html');
      html1.style.width = '200px';
}

function ganti3(){
       var django1 = document.getElementById ('django');
       django1.setAttribute('style','transform:rotate(180deg)');
}
```

Link to JS File

Put your external Java Script file to Static Folder

```
<script src="{% static 'js/hello.js' %}"></script>
```

Don’t forget to collectstatic and load staticfiles

Add event onclick to your HTML tag

```
<img id="jscript" onclick="ganti1()" src = ……>
```

Create function ganti1 on your JavaScript Files

JavaScript Library

Introduction

JavaScript is a powerful language, but it has many flaws:

- the DOM can be clunky to use
- the same code doesn't always work the same way in every browser
  - code that works great in Chrome, Firefox, Safari, ... will fail in IE and vice versa
- many developers work around these problems with hacks (checking if browser is IE, etc.)
- JavaScript libraries out there: jQuery, D3.JS, React.JS
- (Bonus) JavaScript Frameworks: Node.JS, Angular

jQuery Features

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

How to use?

- Download the jQuery library from jQuery.com

  or

- Include jQuery from a CDN, like Google

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/j
query.min.js"></script>
```

Example: window.onload JavaScript vs jQuery

- The DOM way (using Pure JavaScript)

```
window.onload = function() {
  // do stuff with the DOM
}
```

vs

- The direct jQuery translation

```
$(document).readv(function() {
  // do stuff with the DOM
});
```

Selecting DOM Objects using $

Basic syntax is: **`$(selector).action()`**

- Finding by id: `$('#idname')`
- Finding by class: `$('.classname')`
- Finding by css selector: `$('div.imgholder')`

More:

https://www.w3schools.com/jquery/jquery_selectors.asp

DOM and jQuery comparison

DOM method	jQuery equivalent
`getElementById("id")`	`$("#id")`
`getElementsByTagName("tag")`	`$("tag")`
`getElementsByName("somename")`	`$("[name='somename']")`
`querySelector("selector")`	`$("selector")`
`querySelectorAll("selector")`	`$("selector")`

<input name="animal" type="checkbox" value="Cats">

jQuery Event Methods

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

https://www.w3schools.com/jquery/jquery_events.asp

jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **`$(selector).action()`**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

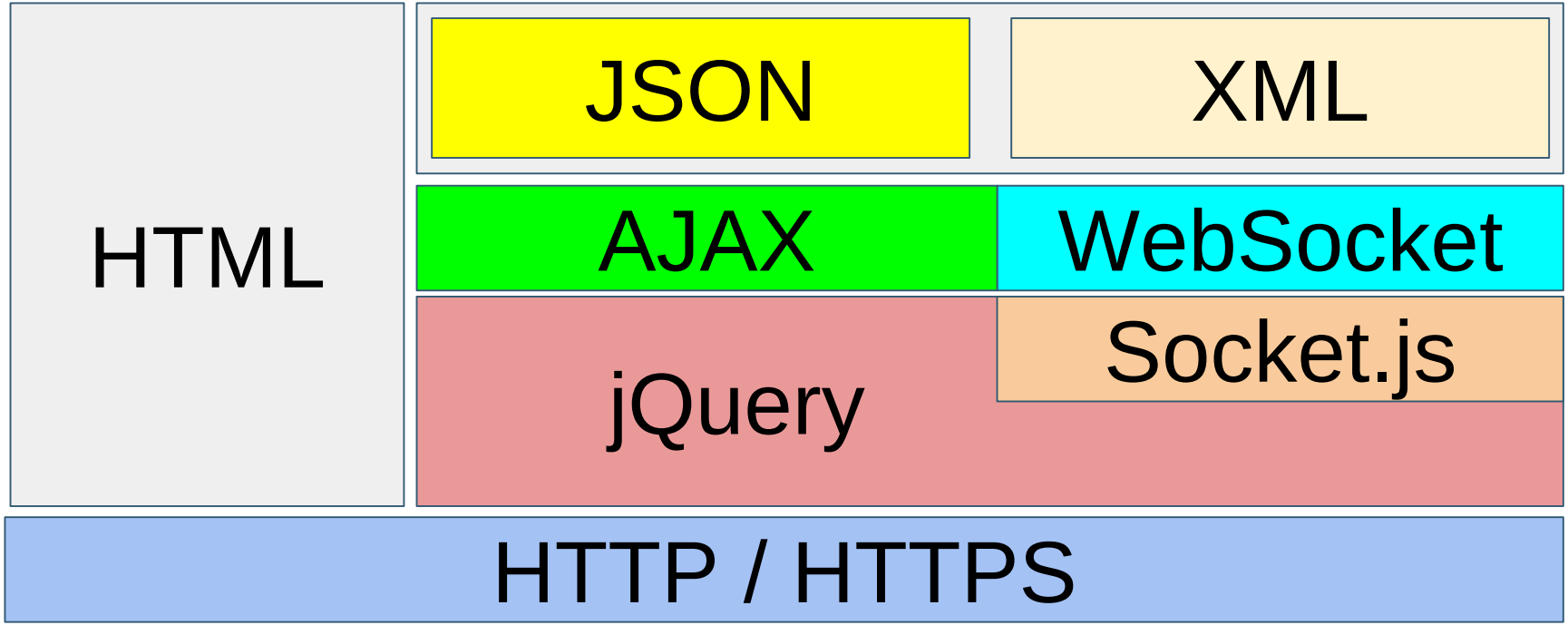
Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me to hide paragraphs</button>

</body>
</html>
```

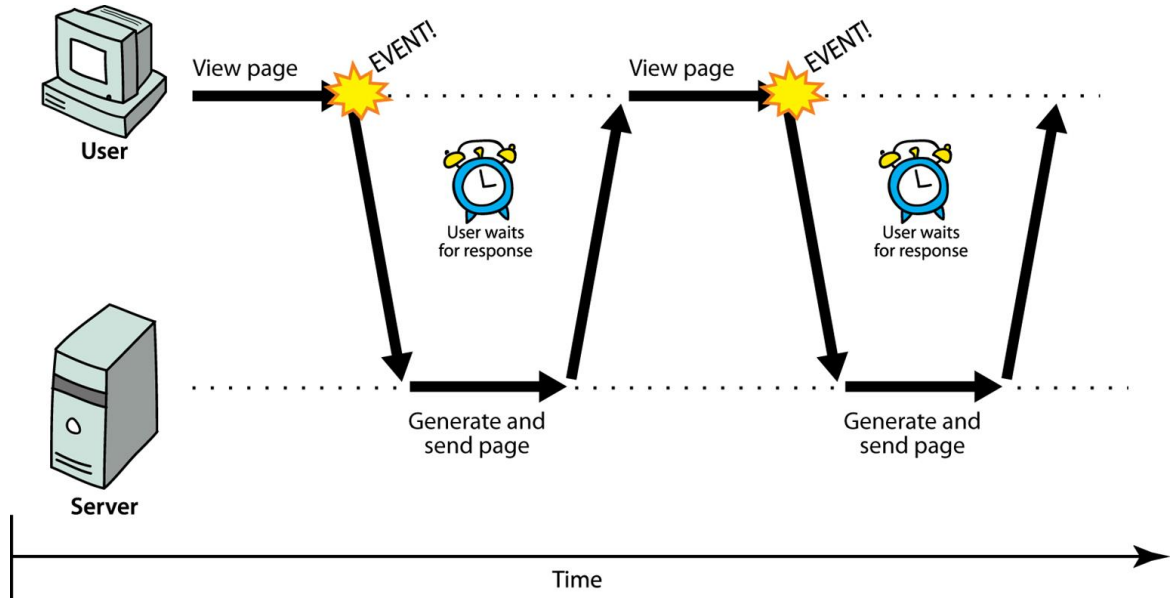
https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_p

Ajax

Data Delivery



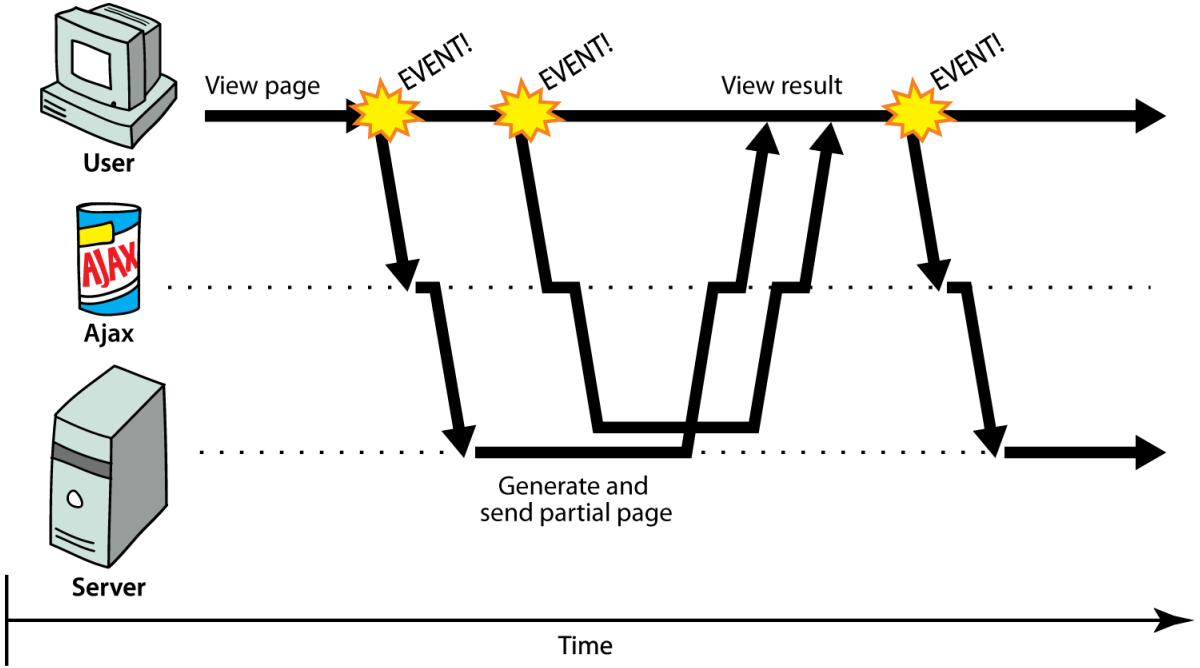
Synchronous Web Communication



user must wait while new pages load (click, wait, refresh)

Source: https://courses.cs.washington.edu/courses/cse154/12au/lectures/slides/lecture22-ajax.shtml#slide1

Asynchronous Web Communication



user can keep interacting with page while data loads

Source: https://courses.cs.washington.edu/courses/cse154/12au/lectures/slides/lecture22-ajax.shtml#slide3

Ajax: Asynchronous JavaScript and XML

- AJAX is a new technique for creating better and more interactive web applications with the help of XML, HTML, CSS, and JavaScript.
- Not a programming language; a particular way of using JavaScript
- Downloads data from a server in the background
- Allows dynamically updating a page without making the user wait
- Avoids the "click-wait-refresh" pattern
- Examples:
  https://www.w3schools.com/xml/tryit.asp?filename=tryajax_first
  https://www.w3schools.com/js/js_ajax_http.asp

Ajax with jQuery

```
$.ajax({
  url: url,        //Specifies the URL to send the request to
  type: GET/POST,  //Specifies the type of request.
  data: data,      //Specifies data to be sent to the server
  success: success,         //A function to be run when the request succeeds
  dataType: type of data,   //The data type expected of the server response.
});
```

Selengkapnya:

https://www.w3schools.com/jquery/ajax_ajax.asp

Ajax with jQuery

```
$(document).ready(function(){
  $("button").click(function(){
    $.ajax({
        url: "https://www.googleapis.com/books/v1/volumes?q=web+application",
        success: function(result){
            for(i=0; i < result.items.length; i++){
                $("#list").append("<div>"+result.items[i].volumeInfo.title+"</div>");
            }
    }});
  });
});
```

JavaScript Fetch API

- Ajax alternative using JavaScript
- Perform all tasks as XMLHttpRequest (XHR) object does
- Requires only one parameter which is the URL of the resource

Fetch()

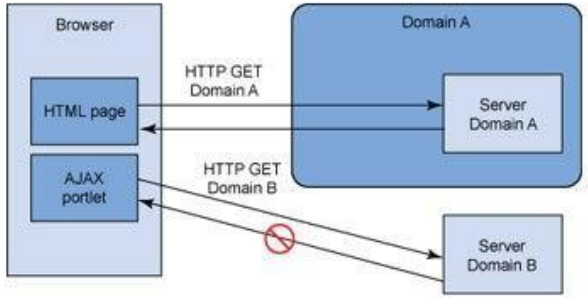
```
<html>

<body>
      <h1> Test Fetch API </h1>
      <p id="fetch"></p>
<script>

   async function fetchText() {
      let response = await fetch('requirements.txt');

      console.log(response.status); // 200
      console.log(response.statusText); // OK

      if (response.status === 200) {
          let data = await response.text();
          // handle data
          document.getElementById("fetch").innerHTML = data;
      }
  }
</script>

   <button type="button" onclick = fetchText();> Watch Your Console Log</button>
</body>
</html>
```

Cross Origin Request

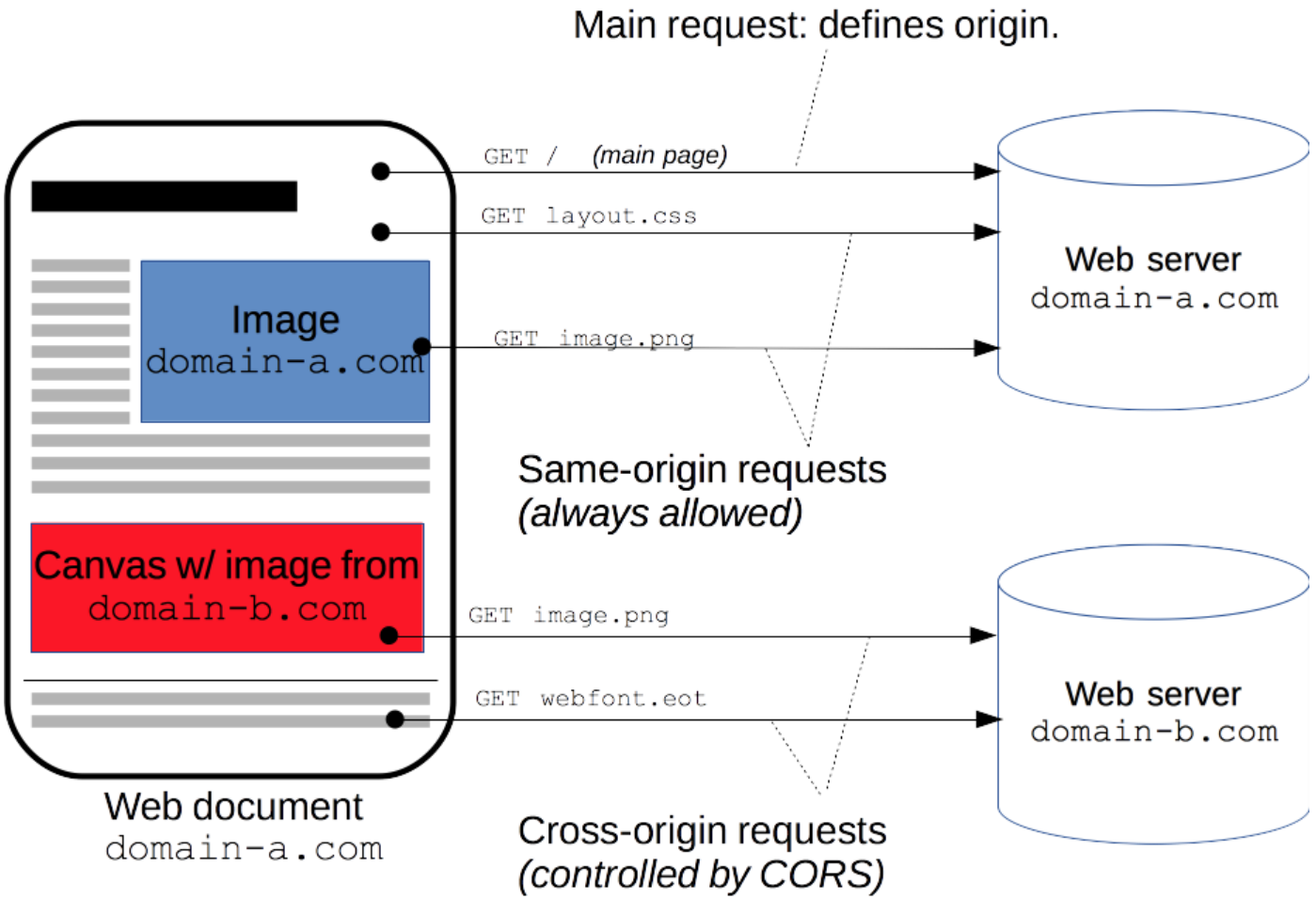
Cross Origin Request

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. A web page may freely embed cross-origin images, stylesheets, scripts, iframes, and videos.

```
DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfchfphceeifdbepaooicaho/sourceMap/chrome/content.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
Access to XMLHttpRequest at 'https://www.w3schools.com/xml/ajax_info.txt' from origin 'http://localhost:8000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.
▶ GET https://www.w3schools.com/xml/ajax_info.txt jquery.min.js:2
net::ERR_FAILED 200
>
```



1. Cross Domain Request


Main request: defines origin.
GET / (main page)
GET layout.css
GET image.png
Image
domain-a.com
Web server
domain-a.com
Same-origin requests
(always allowed)
Canvas w/ image from
domain-b.com
GET image.png
GET webfont.eot
Web server
domain-b.com
Web document
domain-a.com
Cross-origin requests
(controlled by CORS)

JavaScript, JSON, and XML

Work with JSON data

```
<!DOCTYPE html>
<html>
<body>

<h2>JSON Object Creation in JavaScript</h2>

<p id="demo"></p>

<script>
var text = '{"name":"John Johnson","street":"Oslo West 16","phone":"555 1234567"}'

var obj = JSON.parse(text);

document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.street + "<br>" +
obj.phone;
</script>

</body>
</html>
```

- More: https://www.w3schools.com/js/js_json_objects.asp

Work with XML

- More: https://www.w3schools.com/xml/dom_nodes_get.asp

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        myFunction(this);
    }
};
xhttp.open("GET", "books.xml", true);
xhttp.send();

function myFunction(xml) {
    var xmlDoc = xml.responseXML;
    var x = xmlDoc.getElementsByTagName('title')[0];
    var y = x.childNodes[0];
    document.getElementById("demo").innerHTML = y.nodeValue;
}
</script>

</body>
</html>
```

References

- jQuery API: http://api.jquery.com/
- W3C jQuery tutorial:
  https://www.w3schools.com/jquery/default.asp
  https://www.w3schools.com/jquery/ajax_ajax.asp
- Slide course by Washington:
  https://courses.cs.washington.edu/courses/cse190m/12su/lectures/slides/lecture15-dom-jquery.shtm