COESIONE ITALIA 21-27 PIEMONTE

Cofinanziato dall'Unione europea

REGIONE PIEMONTE

FinTech Software Developer

Programmazione WEB - HTML | CSS | Javascript

Docente: Shadi Lahham

Intervento realizzato da ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

JQuery

Using a library

Shadi Lahham - Web development

Libraries

What's a library?

A collection of reusable methods for a particular purpose.
A math library might have functions like:

```
math.sum(array);
math.pow(num, num);
math.factorial(num);
```

Including a library

Include a script tag to the library on your page.
Call functions from that library

```
<body>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/mathjs/3.11.4/math.js"></script>
  <script>
    let answer = math.sum(2, 2);
    console.log(answer);
  </script>
</body>
```

Intro to jQuery

jQuery

- A very popular library (though lately in decline)
- Write less code for common tasks
- Abstract on top of cross-browser differences
- Open source, big community
- Well documented

Review: DOM & Data

```
// Traversing the DOM with JavaScript
document.getElementById('presentation');
document.getElementsByClassName('future');
document.getElementsByTagName('body');
document.querySelectorAll('a');
document.querySelector('img');


// Traversing the DOM with jQuery
$('#presentation');
$('.future');
$('body');
$('a');
$('img').first();
```

Why jQuery

```
// No library

let elems = document.getElementsByTagName('img');
for (let i = 0; i < elems.length; i++) {
  elems[i].style.display = 'none';
}

// jQuery

$('img').hide();
```

Why jQuery

```
// No library

let p = document.createElement('p');
p.appendChild(document.createTextNode('Welcome!'));
p.style.cssFloat = 'left';
p.style.backgroundColor = 'red';
p.className = 'special';
document.querySelector('div.header').appendChild(p);

// jQuery

let newP = $('<p>Welcome!</p>');
newP.css({ float: 'left', 'background-color': 'red' });
newP.addClass('special');
$('div.header').append(newP);
```

jQuery: Select and manipulate

```
<p>Welcome to jQuery!</p>

$('p').addClass('special');

<p class="special">Welcome to jQuery!</p>
```

jQuery: Finding Elements

```
<p>Welcome!</p>
$('p')

<div id="main">Welcome!</div>
$('#main')

<p class="intro">Welcome!</p>
$('.intro')

<div id="main">
  <p class="intro">Welcome!</p>
</div>
$('#main .intro')
```

All CSS selectors are valid, plus more here:
jQuery Selectors

jQuery: Reading Elements

```
// HTML
<a id="yahoo" href="http://www.yahoo.com" style="font-size:20px;">Yahoo!</a>

// find it
$('a#yahoo');

// store it
let myLink = $('a#yahoo');

// get info about it
myLink.html(); // 'Yahoo!'

myLink.attr('href'); // 'http://www.yahoo.com'

myLink.css('font-size'); // '20px'
```

jQuery: Changing Elements

```
// If we start with this HTML:
<a href="http://www.google.com">Google</a>

// We can use this jQuery:
$('a').html('Yahoo!');
$('a').attr('href', 'http://www.yahoo.com');
$('a').css({'color': 'purple'});

// And we'll get this:
<a href="http://www.yahoo.com" style="color:purple">Yahoo</a>
```

jQuery: Create, Manipulate and Inject

```
// Create element and store a reference
let p = $('<p>')

// Use a method to manipulate
p.addClass('special');

// Inject into your HTML
$('body').append(p);
```

jQuery: Create and Store

```
// Pass any HTML string to jQuery. It will create and return a jQuery object.

$('<p>');
<p></p>

$('<p>Welcome!</p>');
<p>Welcome!</p>

$('<p class="intro">Welcome!</p>');
<p class="intro">Welcome!</p>

// store in a variable
let myParagraph = $('<p class="intro">Welcome!</p>');
```

jQuery: Manipulate and Inject

```
// store a reference to make changes to an element
let myParagraph = $('<p class="intro">Welcome!</p>');
myParagraph.css('font-size','4em');


// the stored reference can be injected in another element
$('body').append(myParagraph);
$('body').prepend(myParagraph);
```

Regular DOM nodes to jQuery objects

```
let paragraphs = $('p'); // an array
let myParagraph = paragraphs[0]; // a regular DOM node
let $myParagraph = $(paragraphs[0]); // a jQuery Object

// can also use loops
for(let i = 0; i < paragraphs.length; i++) {
  let element = paragraphs[i];
  let paragraph = $(element);
  paragraph.html(paragraph.html() + ' wowee!!!!');
};
```

Including jQuery

Ways to include jQuery (or any library)

Download and link the file:

```
<head>
<script src="jquery-3.2.1.min.js"></script>
</head>
```

Use a CDN:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
```

```
<head>
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.min.js"></script>
</head>
```

When using a popular CDN like Google or Microsoft, there is a high likelihood that the user already has the .js file in their cache already!

Content Delivery Network

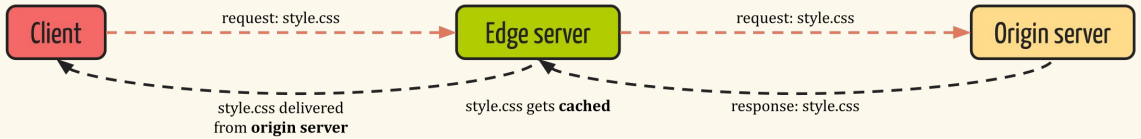
CDN - Content Delivery Network

- A large distributed system of servers deployed in multiple data centers across the Internet
- Serve content to end-users with high availability and high performance
- CDNs serve a large fraction of the Internet content today
  - web objects: text, graphics and scripts
  - downloadable objects: media files, software, documents
  - applications: e-commerce, portals
  - live streaming media
  - on-demand streaming media
  - social networks

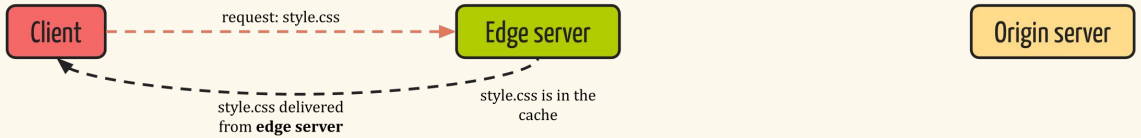
How CDNs work

Content Delivery Network

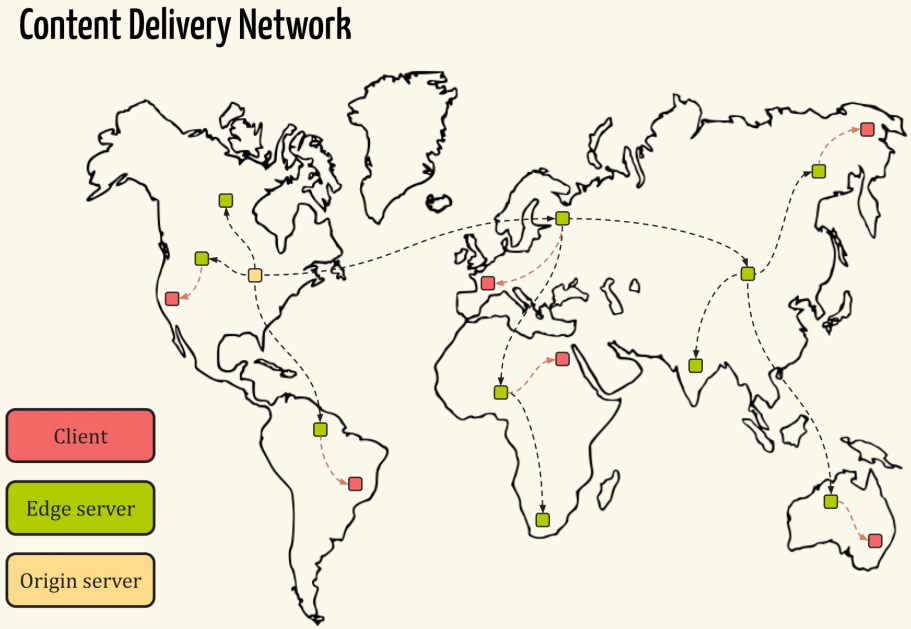
First request



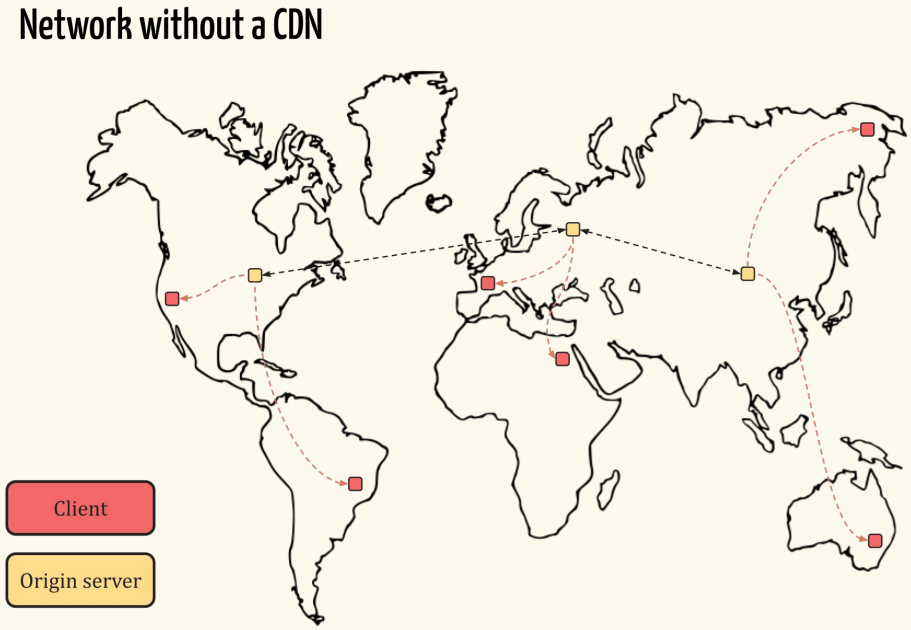
Subsequent requests



How CDNs work



How CDNs work



Using a CDN

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

<script crossorigin src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js"></script>
```

Some known CDNs

Cloudflare CDN

Google cloud CDN

Amazon CloudFront CDN

UNPKG

CDN package search

cdnjs

Your turn

1.VidList

Open the included youtube zip file for this exercise and read the code and the documentation

- Create a **complete** HTML page with main.js, style.css and the library youtube.js
- Add a list of at least 20 youtube videos that you like as links

```
<ul>
      <li><a href="https://www.youtube.com/watch?v=C0DPdy98e4c">vid 1</a></li>
      <li><a href="https://www.youtube.com/watch?v=C0DPdy98e4c">vid 2</a></li>
</ul>
```

- Add a "transform" button to the page

Continues on next page >>>

1.VidList

- When the "transform" button is clicked, it transforms each element of the list into a containing:
  - The title of the video
  - The ID of the video
  - The url of the video (clickable)
  - An image of the video (also clickable)
  - When hovering the image should display a play button and a play icon
- Clicking the button again returns the list to the original form

note:
Do not use bootstrap, jQuery or any other Framework, just plain HTML and JS!

2.jQ VidList

- Repeat the previous exercise but use jQuery instead of plain javascript
- Add any bonus features that you think would make the application better and show that you have understood how to use jQuery

Important!
Do not use bootstrap or any other Framework, just plain HTML and jQuery

Bonus

3.jQ DOM Detective

- Install the following chrome extension https://goo.gl/HJiHEi
- Go to www.gog.com and click on the extension icon and click "add new"
- Click on options and check the box jQuery 3
- In the JS editor on the left add the following code:

```
$( document ).ready(function() {
    console.log( "Page ready!" );
});
```

- Open the chrome inspector (F12)
- The gog.com page outputs a lot of test to the console
- Look for Our console log that says "Page ready" in the console output
- If you see it you are ready to start, go to the next slide

Continues on next page >>>

3.jQ DOM Detective

Use jQuery to find the following elements of the page:

- Every image on the page
- The main menu at the top of the page
- All the news items under "News"
- The footer
- All the social media links at the bottom of the page

For each of the previous elements:

- Hide it
- Change it’s appearance using .css
- Remove it from the page
- Move it to another DOM element on the page
- Add your own new DOM element and add some new content to the page

Submit the code that you used in a file called jq-dom-detective.js

References

Selecting Elements – You Don't Need jQuery!

You Might Not Need jQuery

10 Tips for Writing JavaScript without jQuery

plainJS - The Vanilla JavaScript Repository

I Still Love jQuery — And You Should, Too.