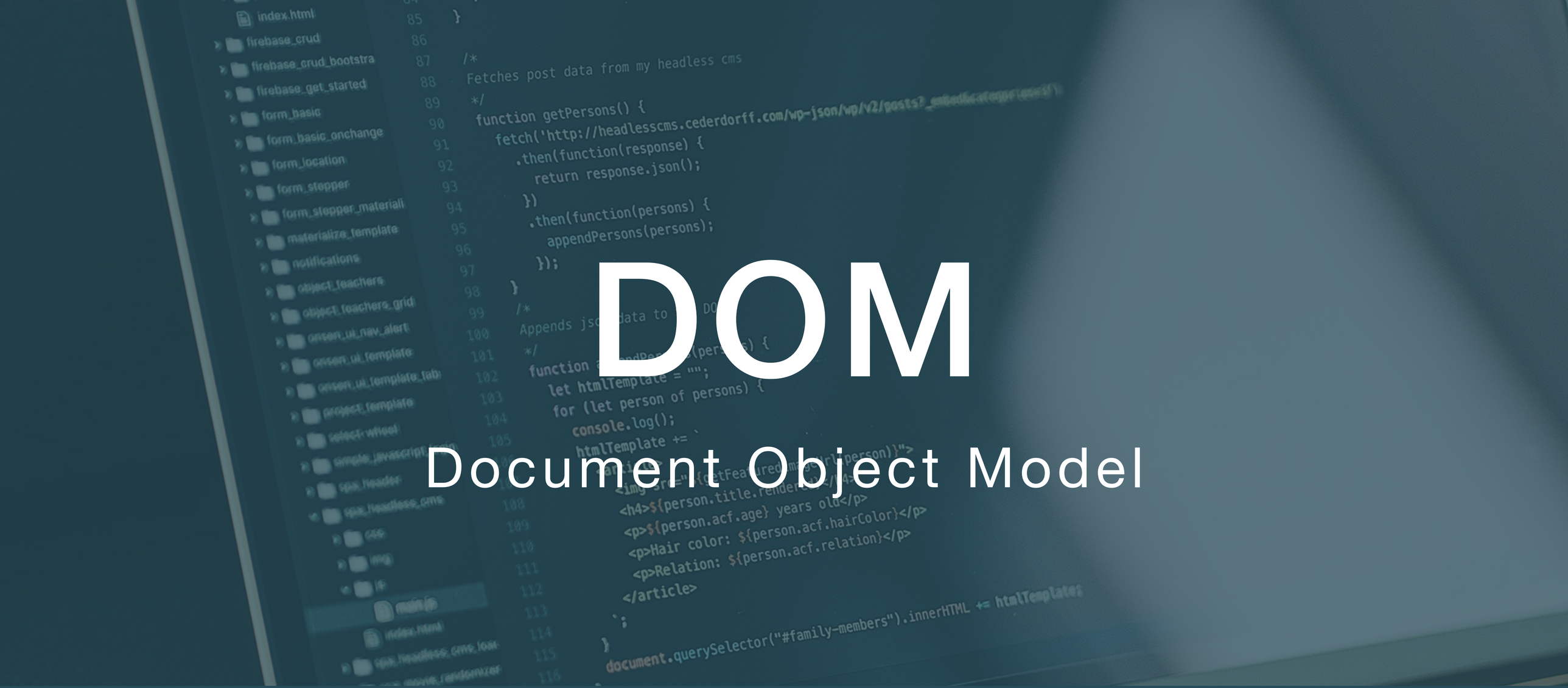


DOM

Document Object Model

Datamatiker

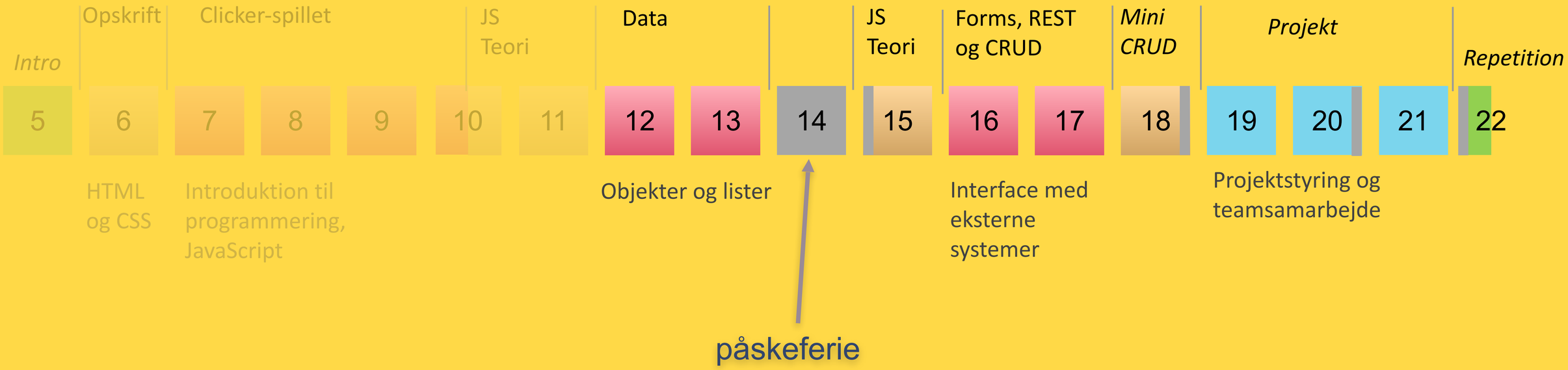
KØBENHAVNS ERHVERVSAKADEMI

Agenda

- Data-forløbet
- The DOM
- DOM-Manipulation
- Øvelser og Opbygning af UI

1. SEMESTERS STRUKTUR

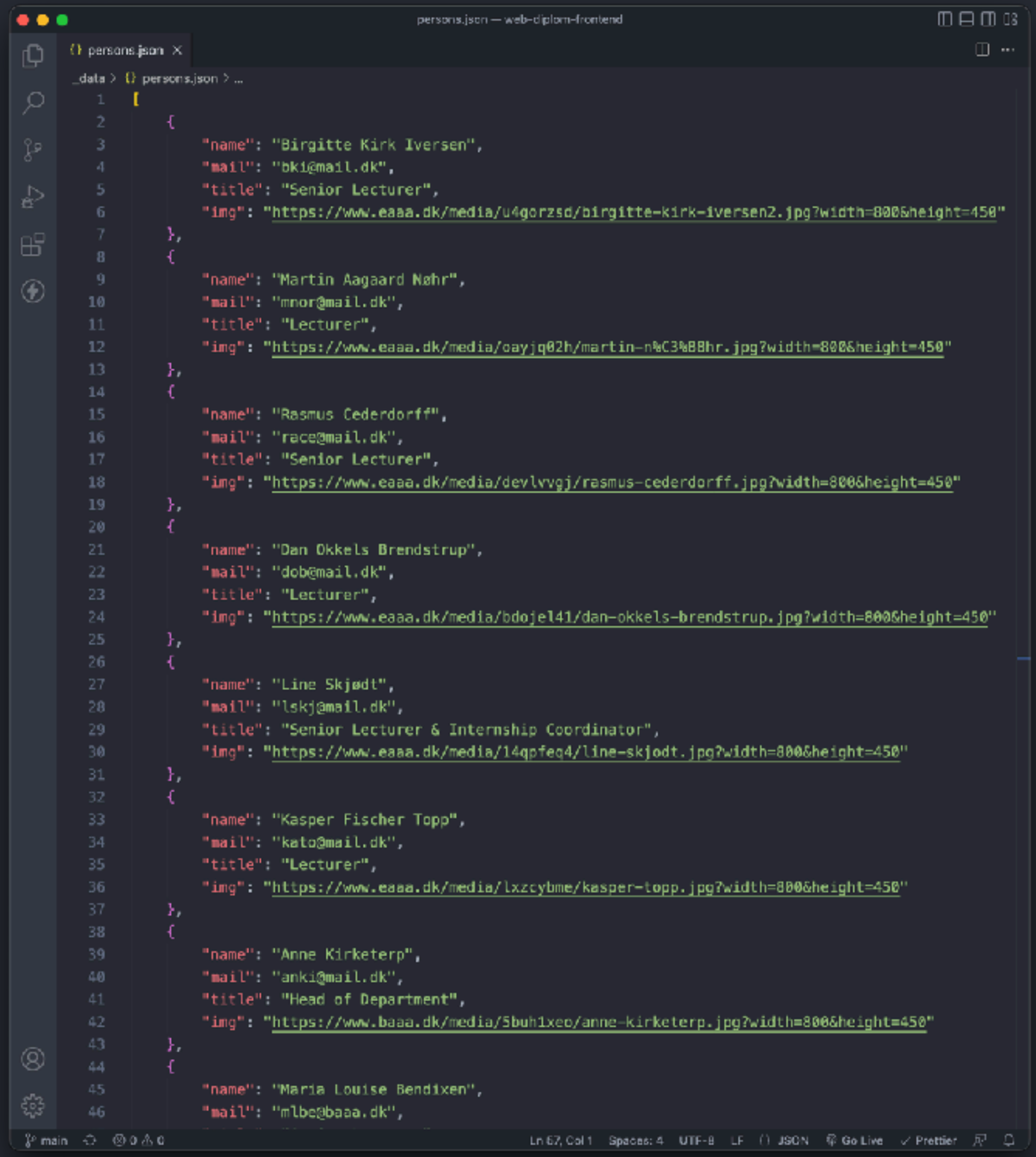
Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)

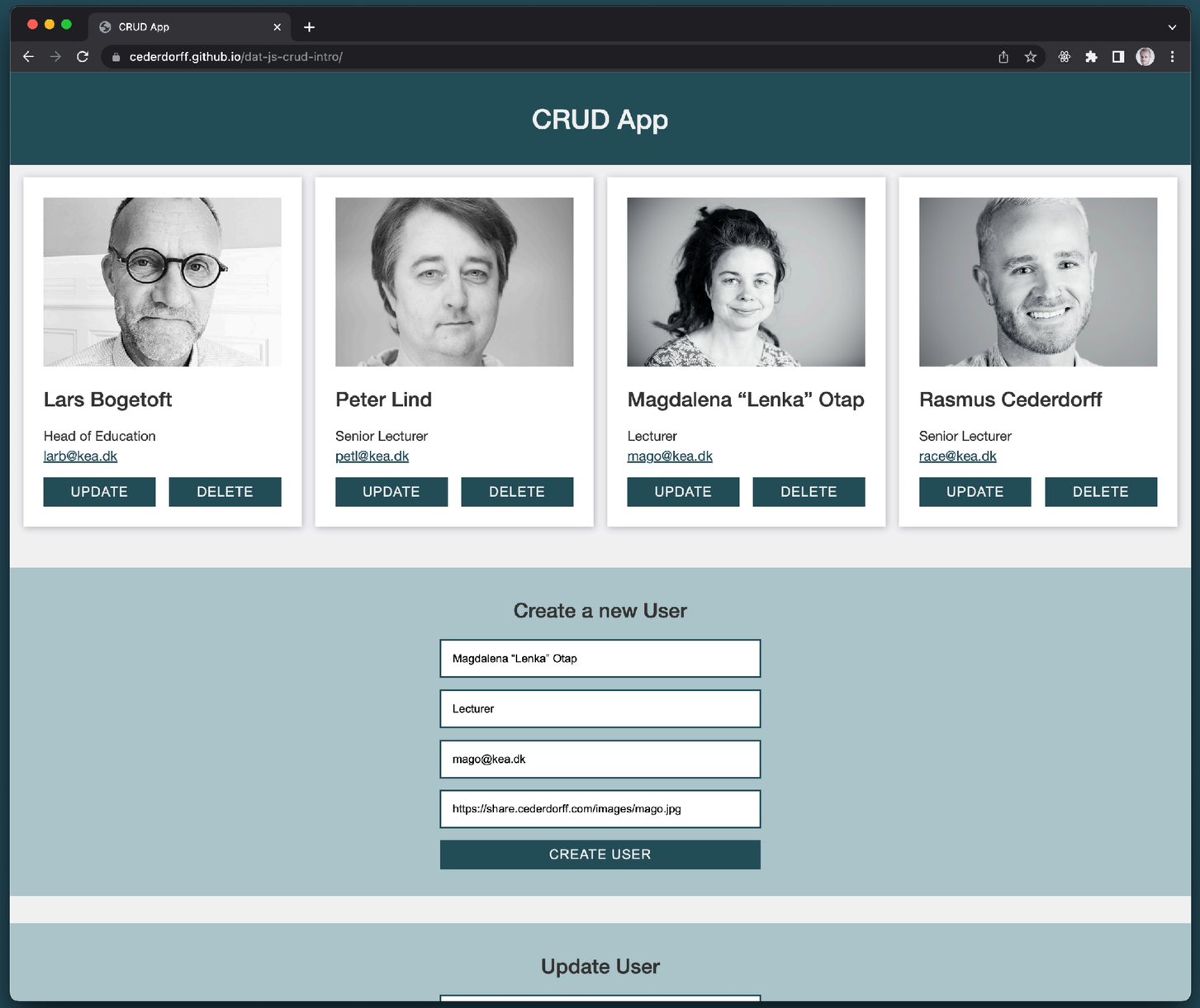


Eksamen ligger 12.-14. juni

Data

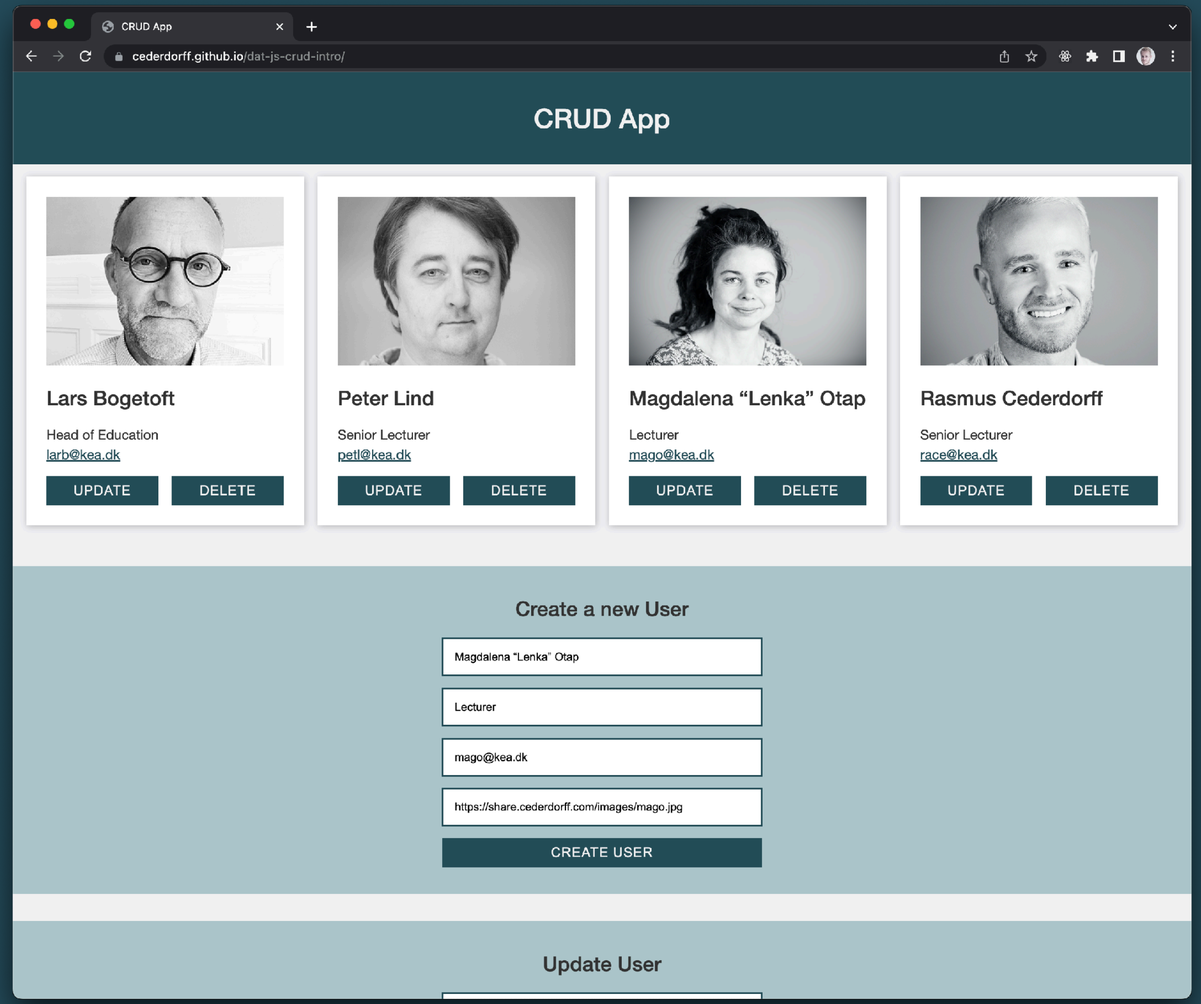
JavaScript, lister, objekter, JSON, fetch, DOM-manipulation og meget mere…





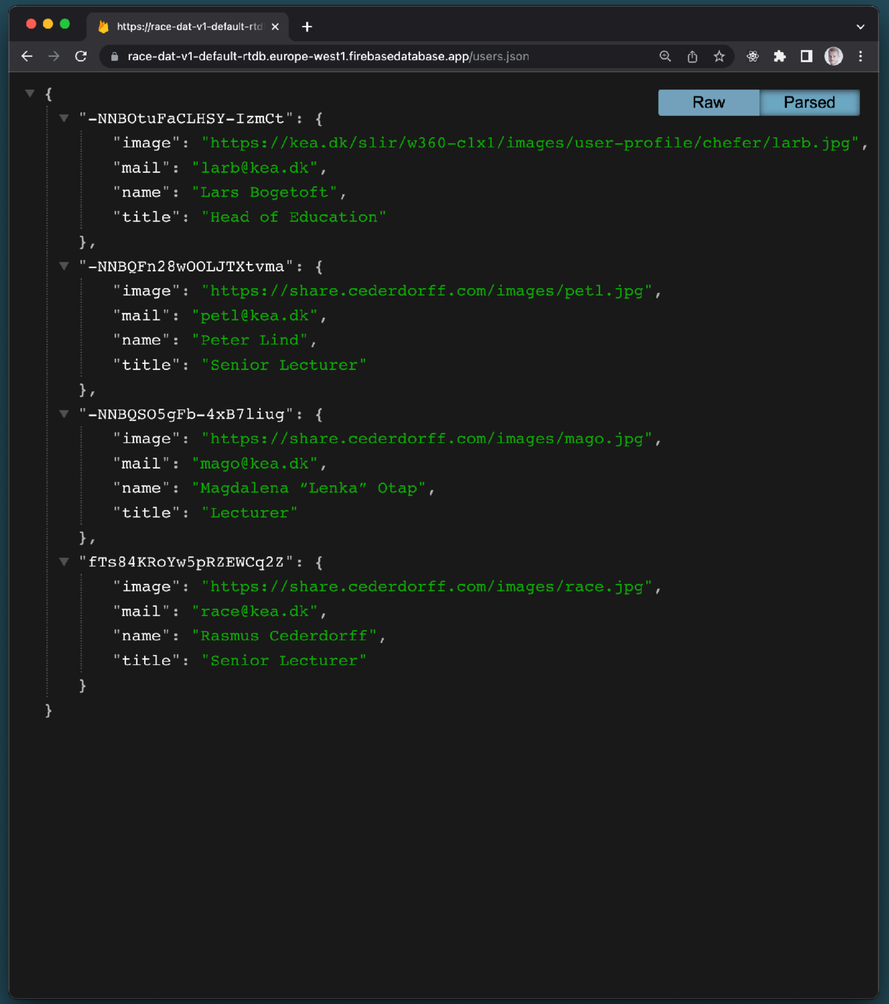
https://cederdorff.github.io/dat-js-crud-intro/

Frontend (client)

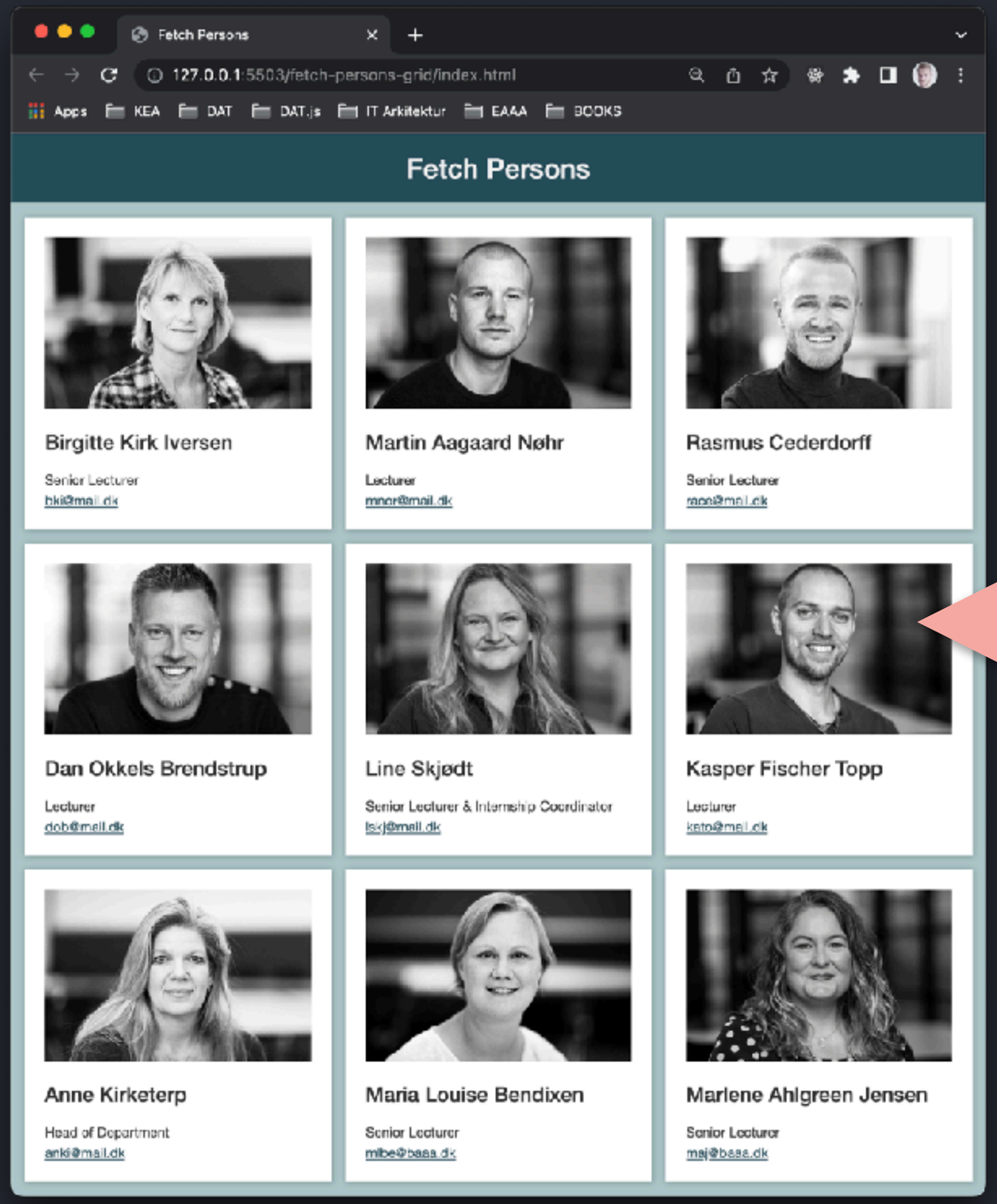


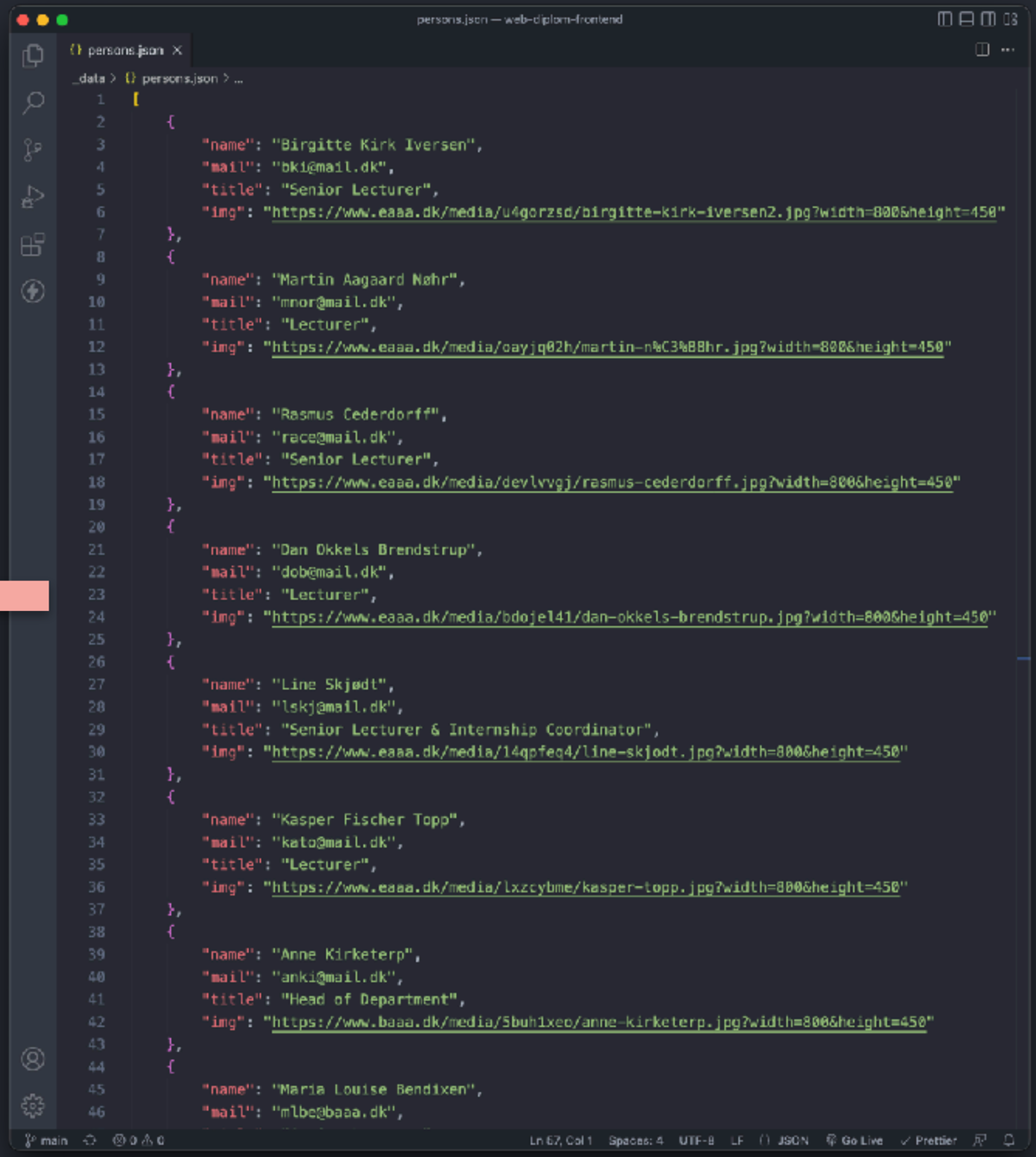
https://cederdorff.github.io/dat-js-crud-intro/

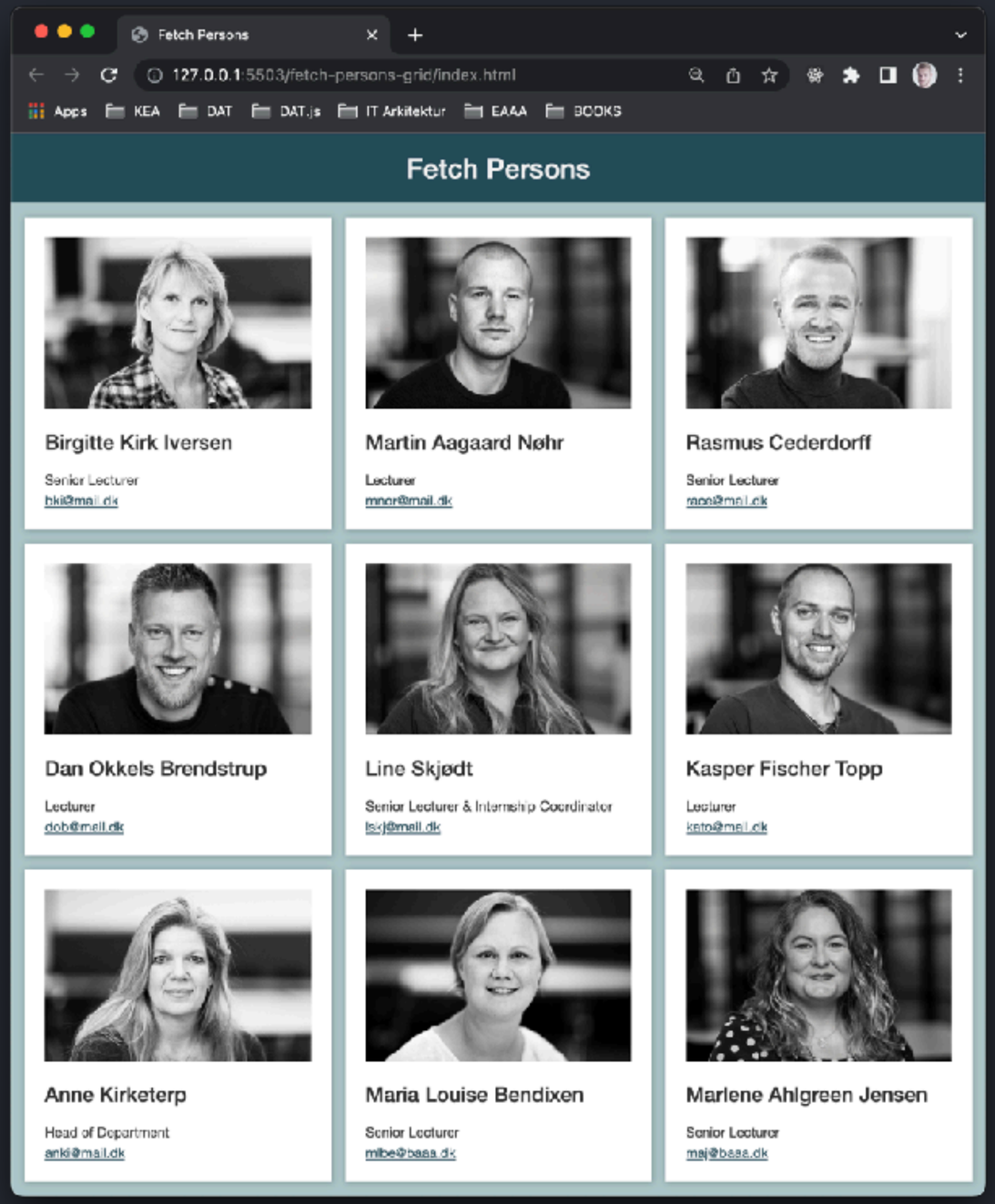
Backend (Server)



https://race-dat-v1-default-rtdb.europe-west1.firebasedatabase.app/users.json


Fetch Persons
127.0.0.1:5503/fetch-persons-grid/index.html
Apps
KEA
DAT
DAT.js
IT Arkitektur
EAAA
BOOKS
Fetch Persons
Birgitte Kirk Iversen
Senior Lecturer
hki@mail.dk
Martin Aagaard Nøhr
Lecturer
mnor@mail.dk
Rasmus Cederdorff
Senior Lecturer
race@mail.dk
Dan Okkels Brendstrup
Lecturer
dob@mail.dk
Line Skjødt
Senior Lecturer & Internship Coordinator
lskj@mail.dk
Kasper Fischer Topp
Lecturer
kato@mail.dk
Anne Kirketerp
Head of Department
anki@mail.dk
Maria Louise Bendixen
Senior Lecturer
mlbe@baaa.dk
Marlene Ahlgreen Jensen
Senior Lecturer
maj@baaa.dk

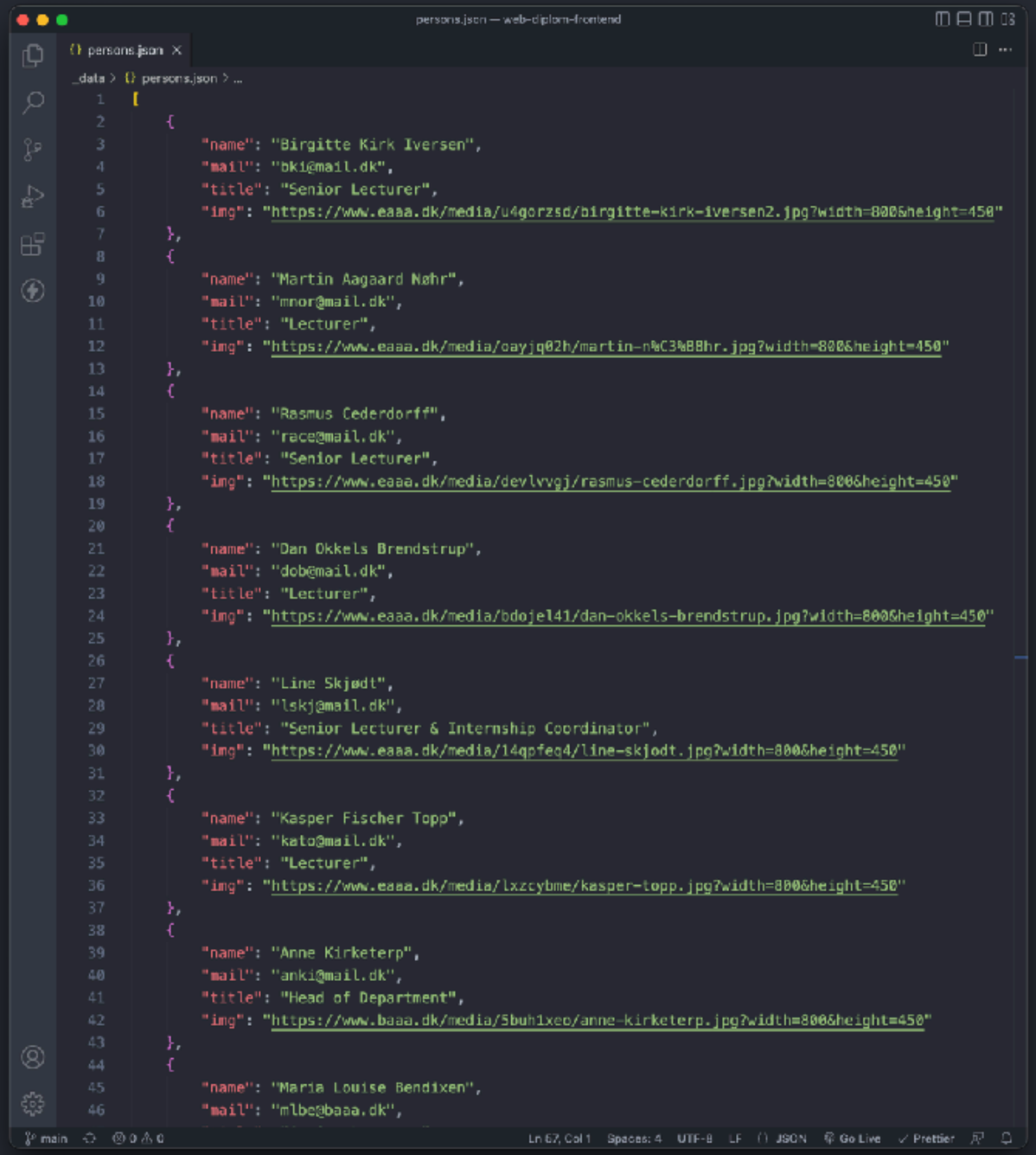

persons.json — web-diplom-frontend
persons.json
_data > persons.json > ...
[
{
"name": "Birgitte Kirk Iversen",
"mail": "bki@mail.dk",
"title": "Senior Lecturer",
"img": "https://www.eaaa.dk/media/u4gorzsd/birgitte-kirk-iversen2.jpg?width=800&height=450"
},
{
"name": "Martin Aagaard Nøhr",
"mail": "mnor@mail.dk",
"title": "Lecturer",
"img": "https://www.eaaa.dk/media/oayjq02h/martin-n%C3%B8hr.jpg?width=800&height=450"
},
{
"name": "Rasmus Cederdorff",
"mail": "race@mail.dk",
"title": "Senior Lecturer",
"img": "https://www.eaaa.dk/media/devlvvgj/rasmus-cederdorff.jpg?width=800&height=450"
},
{
"name": "Dan Okkels Brendstrup",
"mail": "dob@mail.dk",
"title": "Lecturer",
"img": "https://www.eaaa.dk/media/bdojel41/dan-okkels-brendstrup.jpg?width=800&height=450"
},
{
"name": "Line Skjødt",
"mail": "lskj@mail.dk",
"title": "Senior Lecturer & Internship Coordinator",
"img": "https://www.eaaa.dk/media/14qpfeq4/line-skjodt.jpg?width=800&height=450"
},
{
"name": "Kasper Fischer Topp",
"mail": "kato@mail.dk",
"title": "Lecturer",
"img": "https://www.eaaa.dk/media/lxzcybme/kasper-topp.jpg?width=800&height=450"
},
{
"name": "Anne Kirketerp",
"mail": "anki@mail.dk",
"title": "Head of Department",
"img": "https://www.baaa.dk/media/5buh1xeo/anne-kirketerp.jpg?width=800&height=450"
},
{
"name": "Maria Louise Bendixen",
"mail": "mlbe@baaa.dk",
main
Ln 67, Col 1
Spaces: 4
UTF-8
LF
JSON
Go Live
Prettier



Forståelse for objekter og arrays

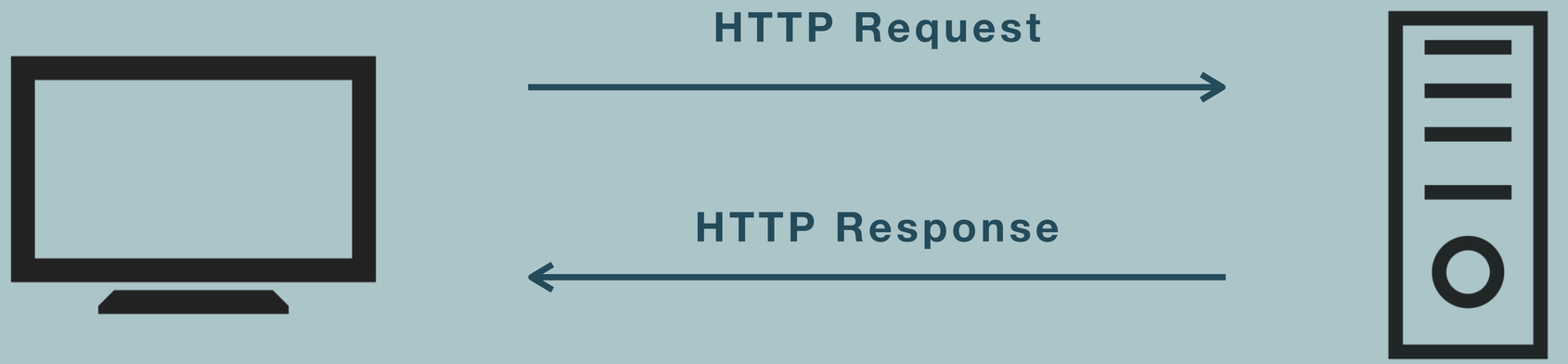
Hvordan viser vi denne JSON-data, så det giver mening for brugerne?

- Denne rette visualisering
- Generere HTML
- DOM-manipulation



Client-Server Model

Communication between web **clients** and web **servers**.



Client-Server Model

Communication between web **clients** and web **servers**.

Hyper Text Transfer Protocol

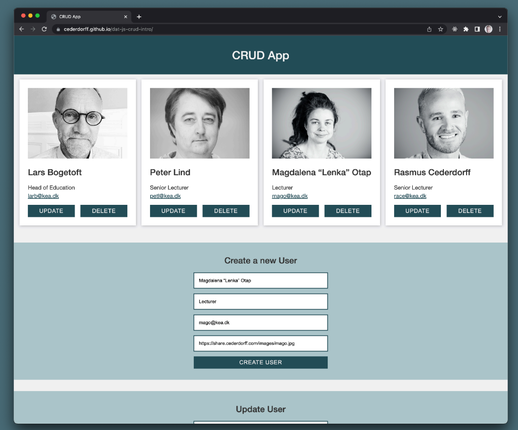
- A protocol and standard for fetching data, HTML and other resources (text, images, videos, scripts, JSON).
- The foundation of the web.

Web Development

WEB APP

REST API

DATA SOURCE



FETCH(GET, POST, PUT, DELETE)

REQUEST

JSON DATA

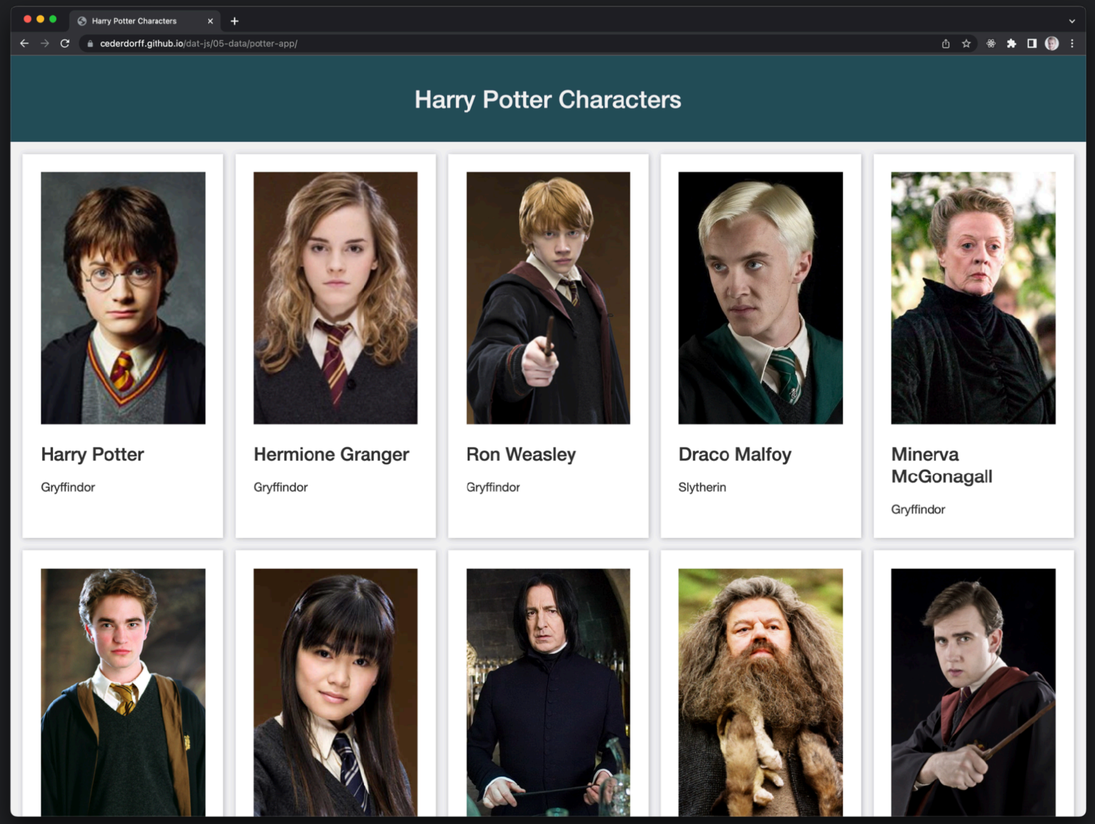
RESPONSE

users

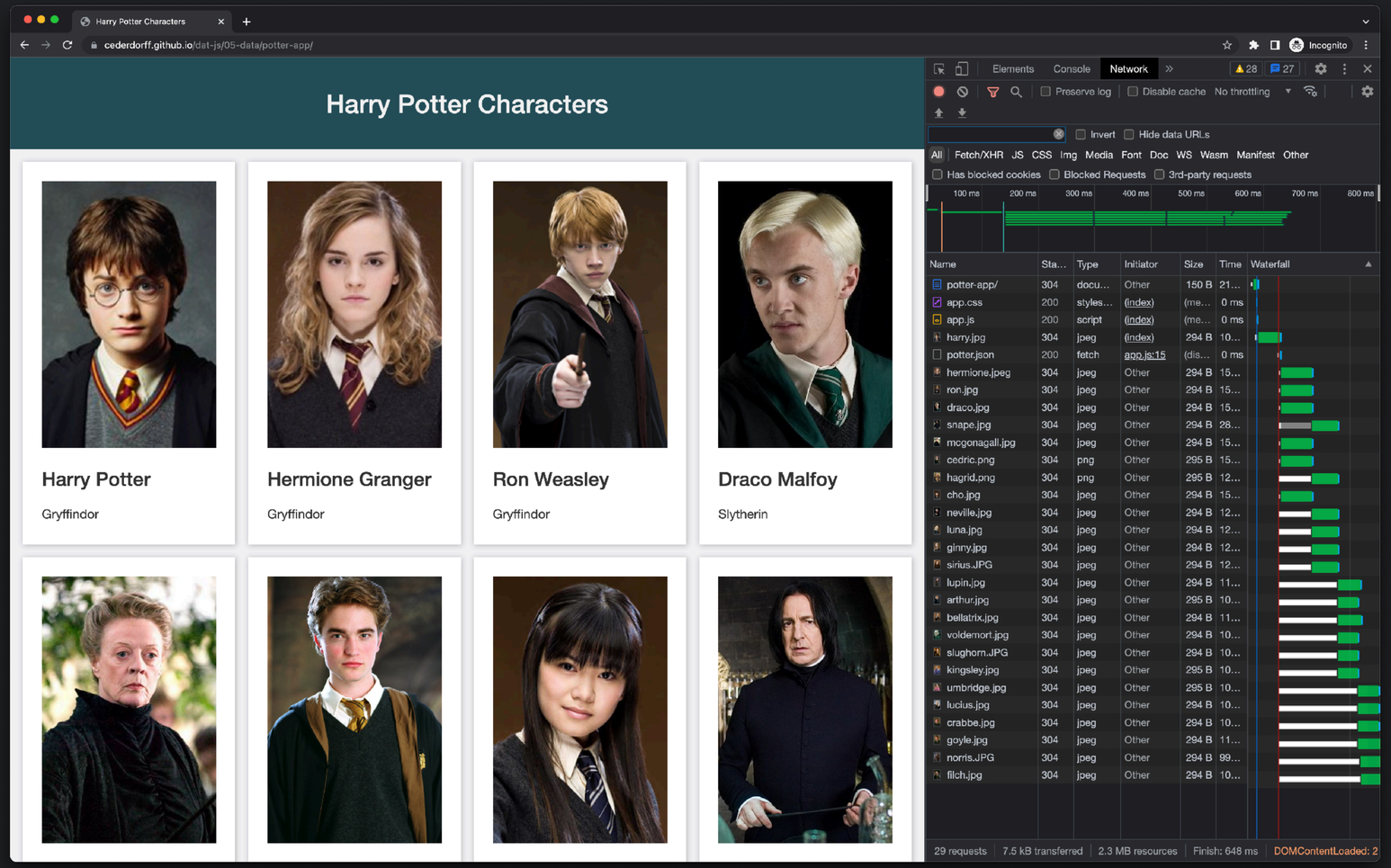
CLIENT

SERVER

The Potter App



Hvad sker der? Tjek Network-tabben!

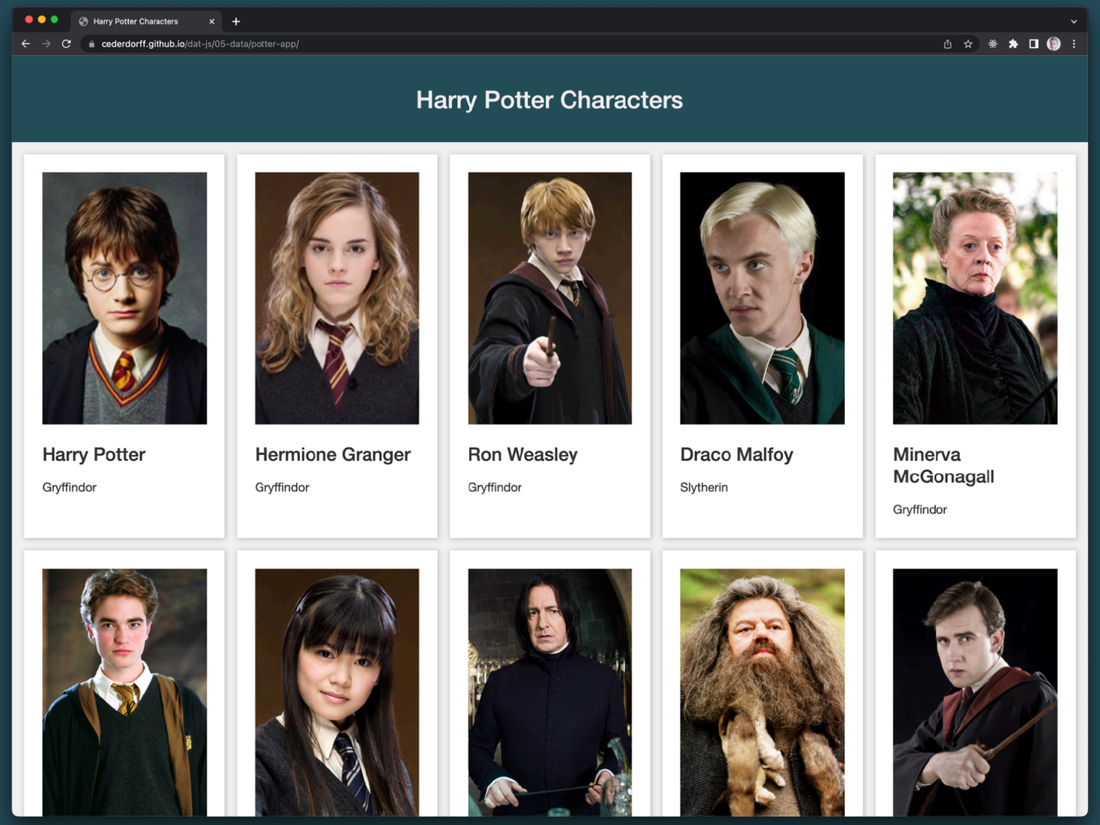


https://cederdorff.github.io/dat-js/05-data/potter-app/

1. Brug Network-tabben til at undersøge:
    1. https://cederdorff.github.io/dat-js/05-data/peters-potter-app/
    2. https://cederdorff.github.io/dat-js/05-data/potter-app/
2. Åben websitet i Chrome (eller en anden browser).
3. Åben Developer Tool (se: How to open the dev tool in your browser) og gå til Network/ Netværk.
4. Genindlæs siden imens du står i Network-tabben og undersøg hvilke ressourcer, der bliver hentet.
5. Hvilke(n) type ressourcer er der tale om?
6. Overvej hvordan ressourcerne hentes.
7. Kan du se hvilken data, der hentes?

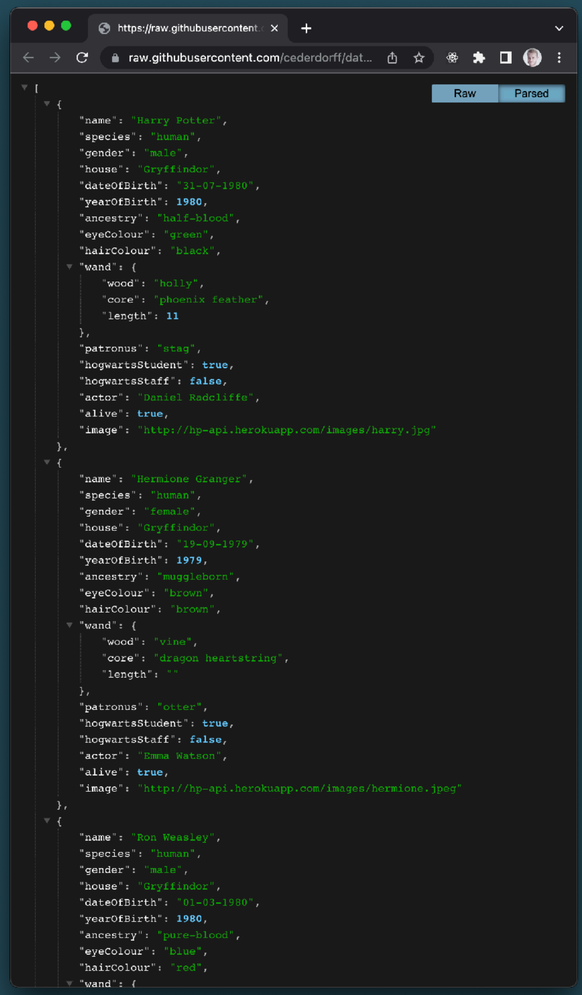
Network-tabben

Frontend (client)



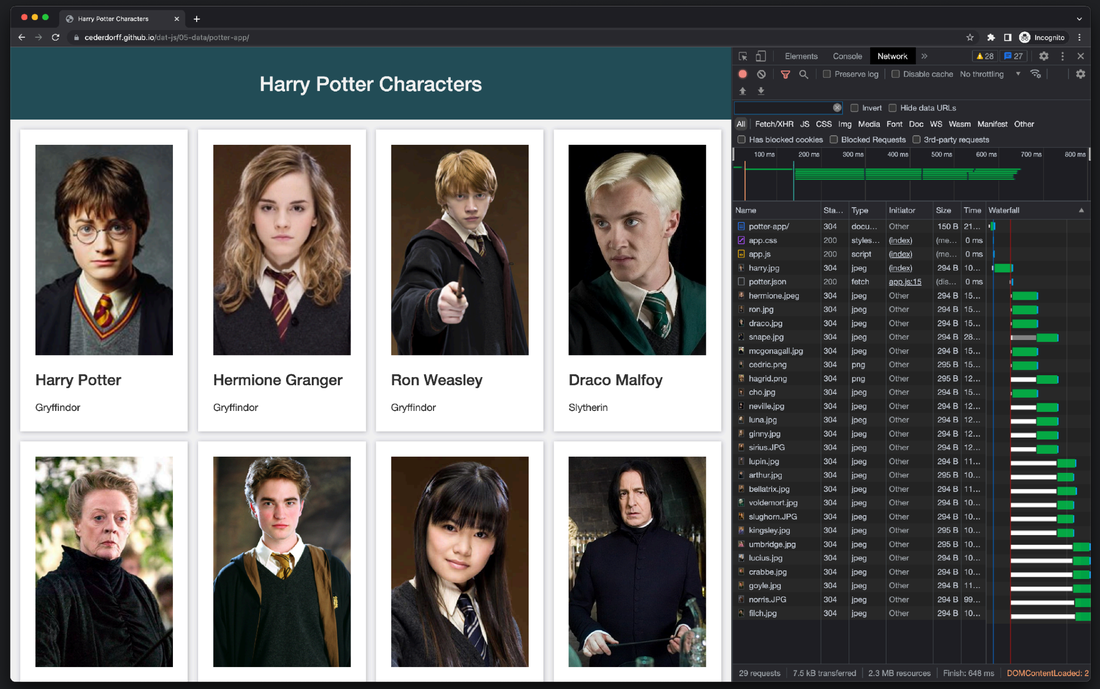
https://cederdorff.github.io/dat-js/05-data/potter-app/

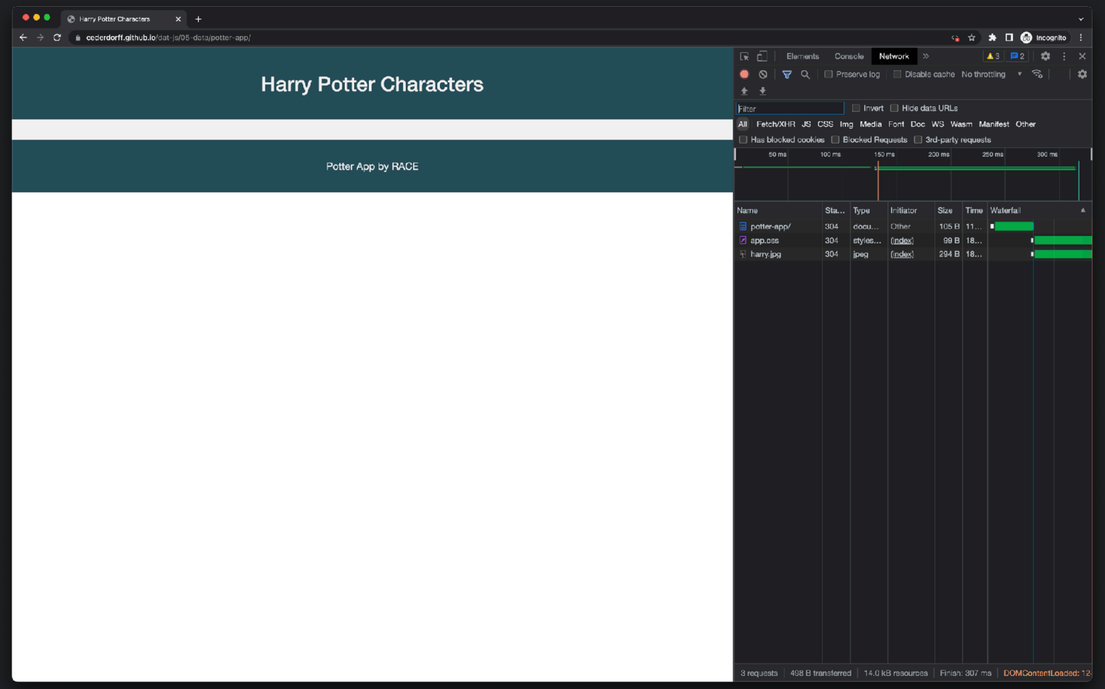
JSON Data Source (Server)



https://raw.githubusercontent.com/cederdorff/dat-js/main/data/potter.json

Hvad sker der uden JavaScript?





How to Disable JavaScript

Med JavaScript

```
…<!DOCTYPE html> == $0
<html lang="en">
  ▶<head>…</head>
  ▼<body>
    ▶<header>…</header>
    ▼<main>
      ▼<section id="characters" class="grid-container"> grid
        ▼<article>
            <img src="http://hp-api.herokuapp.com/images/harry.jpg">
            <h2>Harry Potter</h2>
            <p>Gryffindor</p>
          </article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
        ▶<article>…</article>
```

Uden JavaScript

```
…<!DOCTYPE html> == $0
<html lang="en">
  ▶<head>…</head>
  ▼<body>
    ▶<header>…</header>
    ▼<main>
        <section id="characters" class="grid-container"></section> grid
      </main>
    ▶<dialog id="dialog-character">…</dialog>
    ▶<footer>…</footer>
      <script src="app.js"></script>
    </body>
</html>
```

DOM

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="app.css">
    <link rel="shortcut icon" href="https://cederdorff.github.io/img/favicon.webp" type="image/x-icon">
    <title>Harry Potter Characters</title>
</head>

<body>
    <header>
        <h1>Harry Potter Characters</h1>
    </header>
    <main>
        <section id="characters" class="grid-container">
            <article>
                <img src="http://hp-api.herokuapp.com/images/harry.jpg">
                <h2>Harry Potter</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/hermione.jpeg">
                <h2>Hermione Granger</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/ron.jpg">
                <h2>Ron Weasley</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/draco.jpg">
                <h2>Draco Malfoy</h2>
                <p>Slytherin</p>
            </article>
```

Source Code (HTML)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="app.css">
    <link rel="shortcut icon" href="https://cederdorff.github.io/img/favic
    <title>Harry Potter Characters</title>
</head>

<body>
    <header>
        <h1>Harry Potter Characters</h1>
    </header>
    <main>
        <section id="characters" class="grid-container"></section>
    </main>

    <dialog id="dialog-character">…
    </dialog>
    <footer>
        <p>Potter App by RACE</p>
    </footer>

    <script src="app.js"></script>
</body>
```

The DOM

“

The Document Object Model (DOM) is a programming interface for web documents.

It represents the page so that programs can change the document structure, style, and content.

”

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

The browser fetches the source code (HTML).

When the source code is loaded the browser creates The HTML DOM.

Source Code (HTML)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="app.css">
    <link rel="shortcut icon" href="https://cederdorff.github.io/img/favic
    <title>Harry Potter Characters</title>
</head>

<body>
    <header>
        <h1>Harry Potter Characters</h1>
    </header>
    <main>
        <section id="characters" class="grid-container"></section>
    </main>

    <dialog id="dialog-character">…
    </dialog>
    <footer>
        <p>Potter App by RACE</p>
    </footer>

    <script src="app.js"></script>
</body>
```

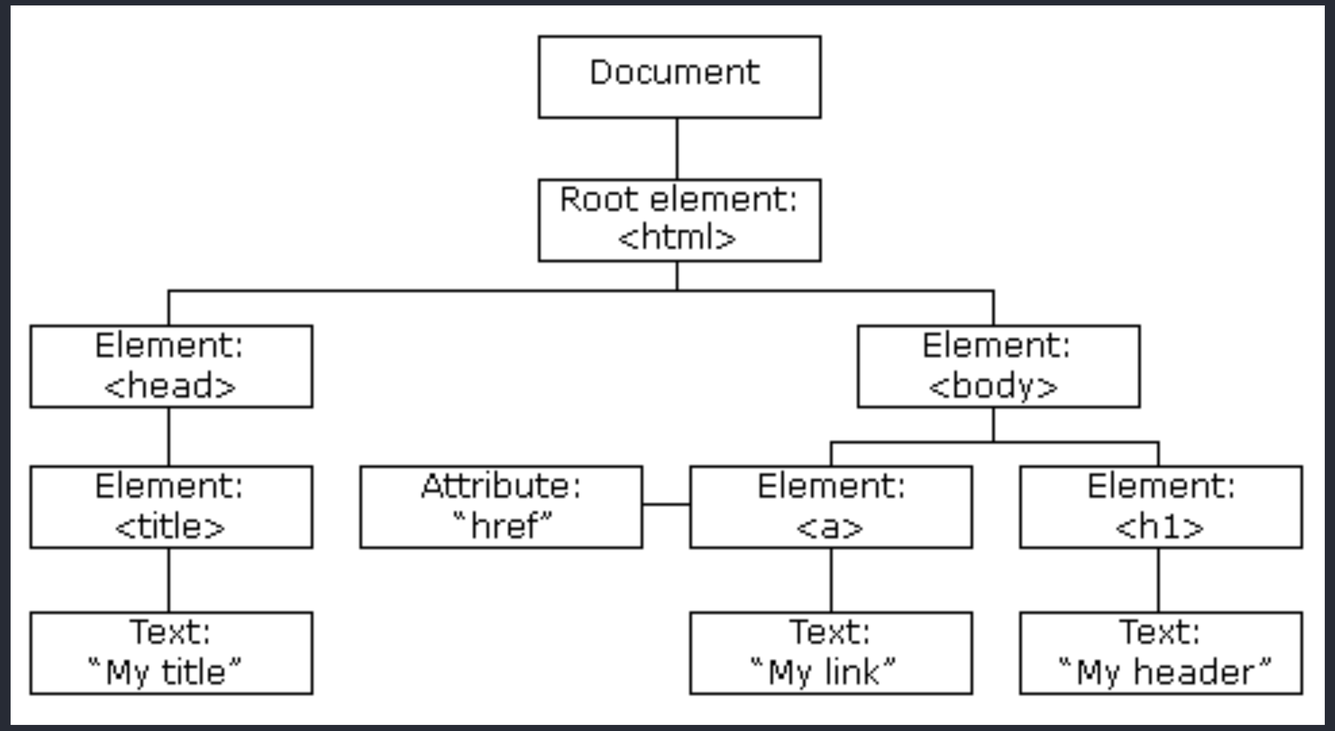
DOM

```html
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="app.css">
    <link rel="shortcut icon" href="https://cederdorff.github.io/img/favicon.webp" type="image/x-icon">
    <title>Harry Potter Characters</title>
</head>

<body>
    <header>
        <h1>Harry Potter Characters</h1>
    </header>
    <main>
        <section id="characters" class="grid-container">
            <article>
                <img src="http://hp-api.herokuapp.com/images/harry.jpg">
                <h2>Harry Potter</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/hermione.jpeg">
                <h2>Hermione Granger</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/ron.jpg">
                <h2>Ron Weasley</h2>
                <p>Gryffindor</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/draco.jpg">
                <h2>Draco Malfoy</h2>
                <p>Slytherin</p>
            </article>

            <article>
                <img src="http://hp-api.herokuapp.com/images/mcgonagall.jpg">
```

The DOM is represent the updated version of the (page) content, which can be updated and manipulated by JavaScript.

The HTML DOM (Document Object Model)

A tree of objects



- Object Model for HTML:
  - HTML elements as objects
  - Properties for all HTML elements
  - Methods for all HTML elements
  - Events for all HTML elements

https://www.w3schools.com/whatis/whatis_htmldom.asp

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <button onclick="tryMe()">Try me</button>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  document.body.style.backgroundColor = "red";
  document.body.style.color = "white";
}
```

What is JavaScript?

.. is the world's most popular programming language.

... is the programming language of the Web.

... is easy to learn.

... can change content of a webpage (HTML content).

... can change styling of HTML.

https://www.w3schools.com/js/default.asp

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <button onclick="tryMe()">Try me</button>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  document.body.style.backgroundColor = "red";
  document.body.style.color = "white";
}
```

With JavaScript we are able to

... build dynamic web pages and web apps.

... fetch content/ data from a backend (web service, data source, etc.) through an API.

... do DOM-manipulation.

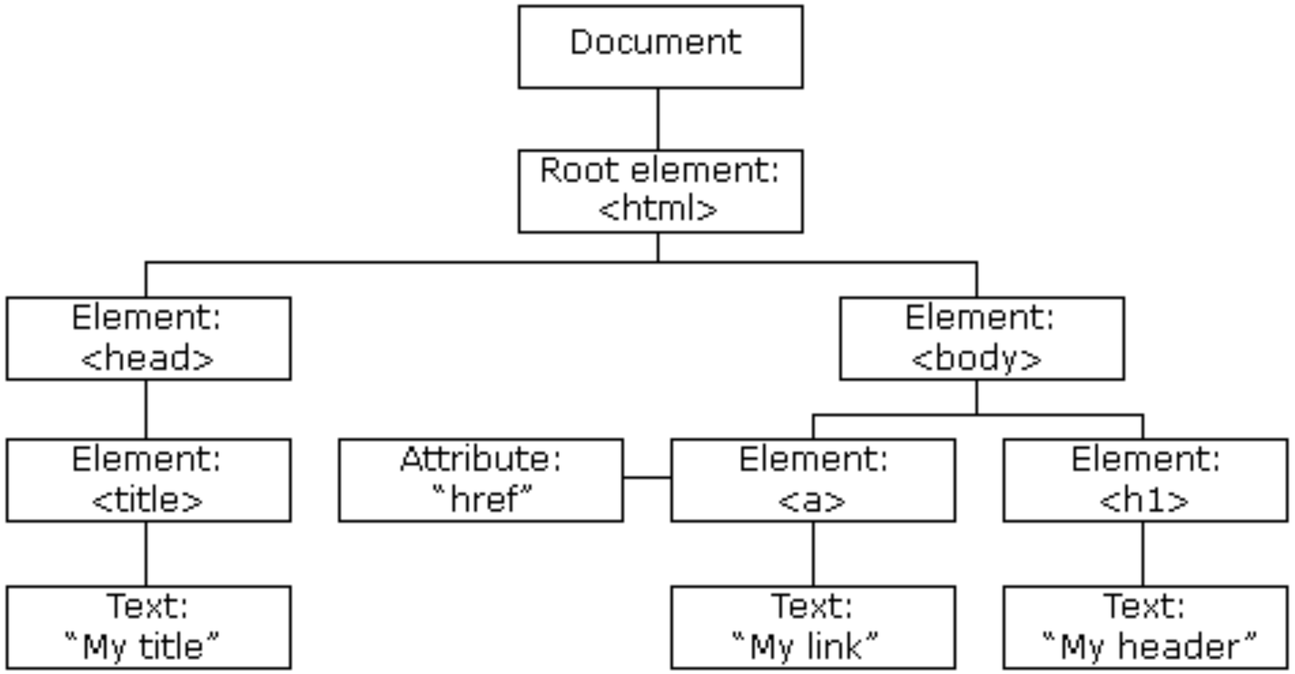
... build and develop anything 🤷

https://www.w3schools.com/js/default.asp

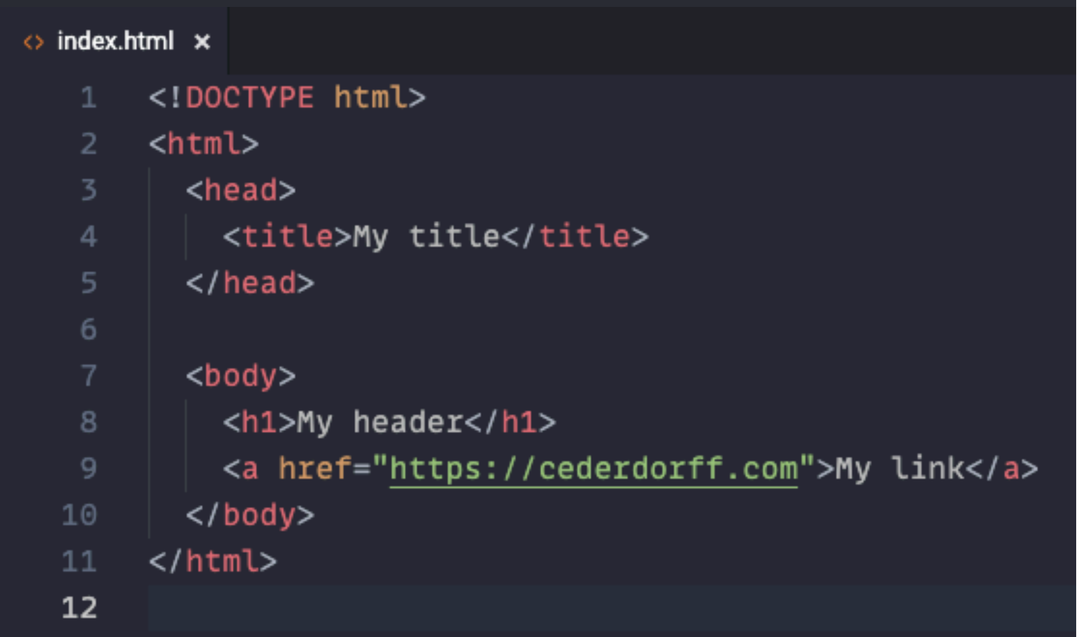
JavaScript HTML DOM

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My title</title>
  </head>

  <body>
    <h1>My header</h1>
    <a href="https://cederdorff.com">My link</a>
  </body>
</html>
```



https://www.w3schools.com/js/js_htmldom.asp
https://javascript.info/dom-nodes
https://javascript.info/dom-navigation



JavaScript HTML DOM

Document Object Model

The HTML document as an object

Gives us the power to create dynamic HTML and manipulate with the HTML (the DOM).

JavaScript can:

… change all the HTML elements in the page
… change all the HTML attributes in the page
… change all the CSS styles in the page
… remove existing HTML elements and attributes
… add new HTML elements and attributes
… react to all existing HTML events in the page
… create new HTML events in the page

https://www.w3schools.com/js/js_htmldom.asp
https://javascript.info/dom-nodes
https://javascript.info/dom-navigation

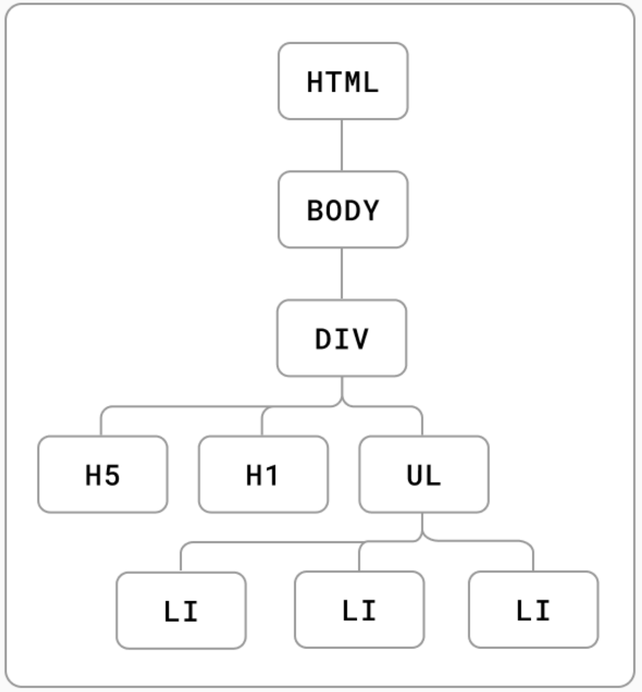
Example?

HTML

index.html

```
1  <html>
2    <body>
3      <div>
4        <h5>Meet the</h5>
5        <h1>Engineers</h1>
6        <ul>
7          <li>A. Lovelace</li>
8          <li>G. Hopper</li>
9          <li>M. Hamilton</li>
10       </ul>
11     </div>
12   </body>
14 </html>
15
16
```

DOM



UI

acme.com/engineers

Meet the

Engineers

- A. Lovelace
- G. Hopper
- M. Hamilton

https://nextjs.org/learn/foundations/from-javascript-to-react

When a user visits a web page, the server returns an HTML file to the browser that may look like this:

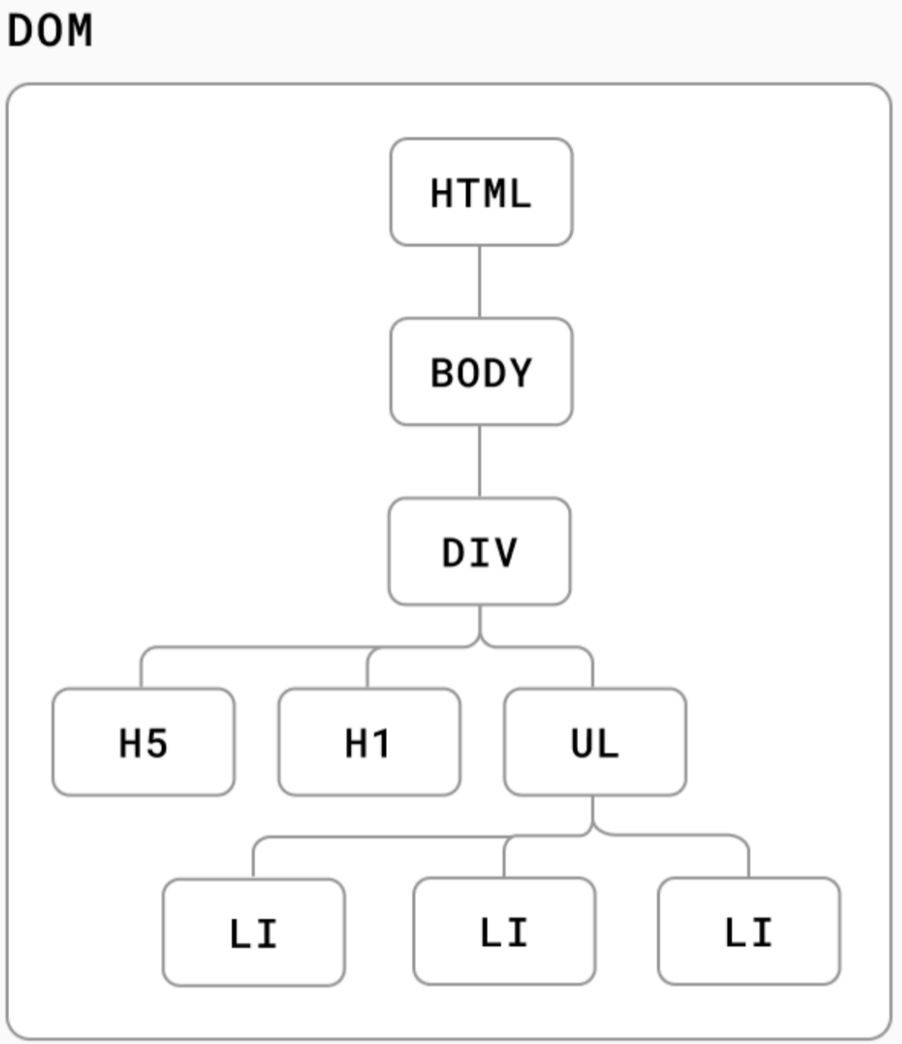
HTML

index.html

```html
<html>
  <body>
    <div>
      <h5>Meet the</h5>
      <h1>Engineers</h1>
      <ul>
        <li>A. Lovelace</li>
        <li>G. Hopper</li>
        <li>M. Hamilton</li>
      </ul>
    </div>
  </body>
</html>
```

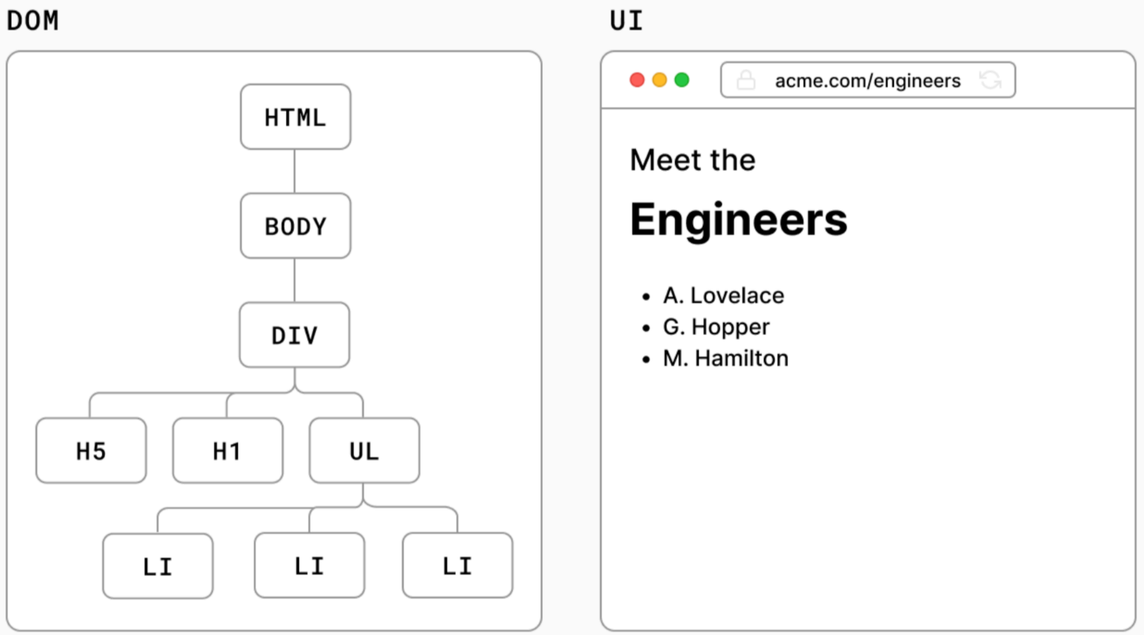
https://nextjs.org/learn/foundations/from-javascript-to-react

The browser then reads the HTML and constructs the Document Object Model (DOM).



https://nextjs.org/learn/foundations/from-javascript-to-react

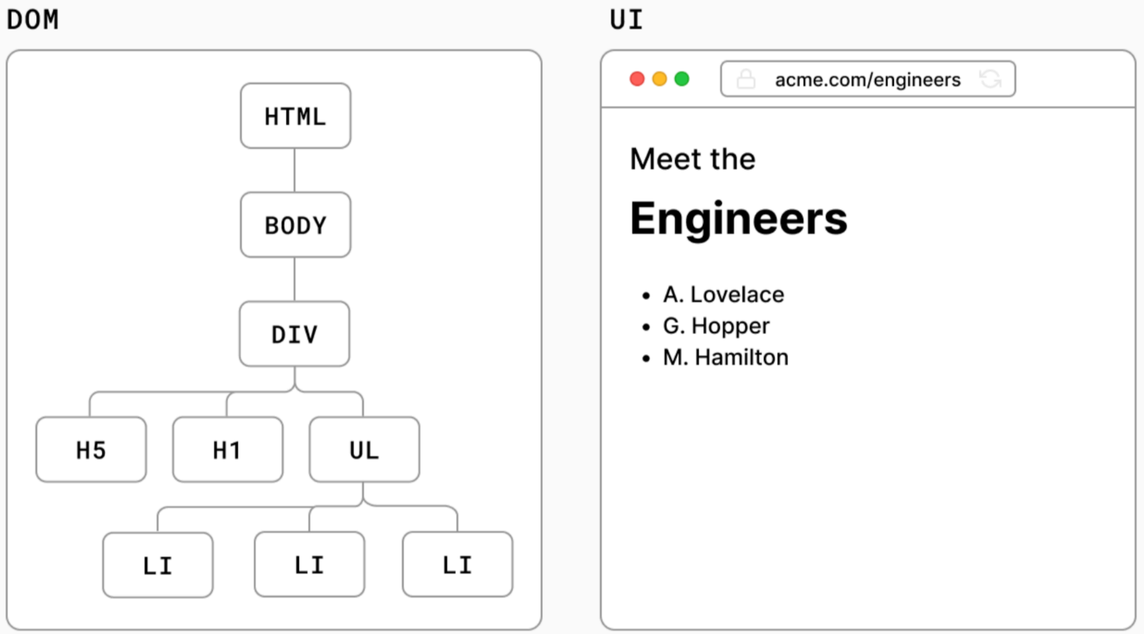
The DOM is an object representation of the HTML elements. It acts as a bridge between your code and the user interface, and has a tree-like structure with parent and child relationships.



https://nextjs.org/learn/foundations/from-javascript-to-react

You can use DOM methods and a programming language, such as JavaScript, to listen to user events and manipulate the DOM by selecting, adding, updating, and deleting specific elements in the user interface. DOM manipulation allows you to not only target specific elements, but also change their style and content.

https://nextjs.org/learn/foundations/from-javascript-to-react



Updating the UI with JavaScript and DOM Methods

```
// Create a new H1 element
const header = document.createElement("h1");

// Add content to H1 element
header.textContent = "Develop. Preview. Ship. 🚀";

// add the newly created element into the DOM
document.querySelector("body").appendChild(header);
```

Let's try it out!

https://nextjs.org/learn/foundations/from-javascript-to-react/updating-ui-with-javascript

DOM VS Source Code

DOM

Elements

```
1  <html>
2    <head></head>
3    <body>
4      <div id="app">
5        <h1>Develop. Preview.
6  Ship. 🚀</h1>
7      </div>
8      <script type="text/
9  javascript">...</script>
10   </body>
11 </html>
```

SOURCE CODE (HTML)

index.html

```
1  <html>
2    <head></head>
3    <body>
4      <div id="app"></div>
5      <script type="text/
6  javascript">...</script>
7    </body>
8  </html>
9
10
11
```

https://nextjs.org/learn/foundations/from-javascript-to-react/updating-ui-with-javascript

DOM Manipulation

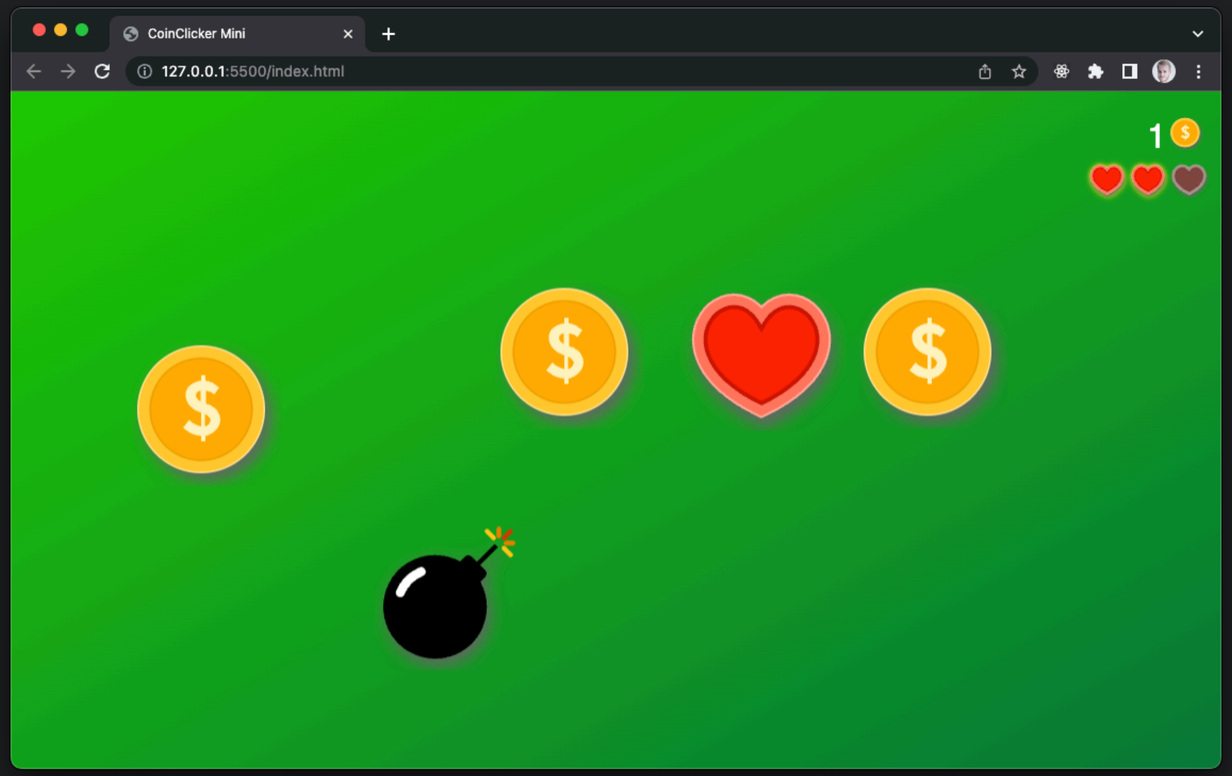
“When you use JavaScript to add, remove, and modify elements of a website”

- document.querySelector(selector)
- document.querySelectorAll(name)
- document.createElement(name)
- parentNode.appendChild(node)
- element.append(node, node, …)
- element.insertAdjacentHTML(position, html)
- element.innerHTML
- element.textContent
- element.style.xxx
- element.setAttribute()
- element.getAttribute()
- element.addEventListener()
- element.classList
- window.content
- Window.onload
- window.scrollTo()

Methods for DOM Manipulation

Common Methods

I Clicker-spillet?



DOM Manipulation

```
let fullName = "Peter Lind";
console.log(fullName);
document.querySelector("#fullName_container").textContent = fullName;
```

```
<body>
    <header>
        <h1 id="fullName_container"></h1>
    </header>
    <script src="app.js"></script>
</body>
```

DOM Manipulation

```
let fullName = "Peter Lind";
console.log(fullName);
document.querySelector("#fullName_container").textContent = fullName;
```

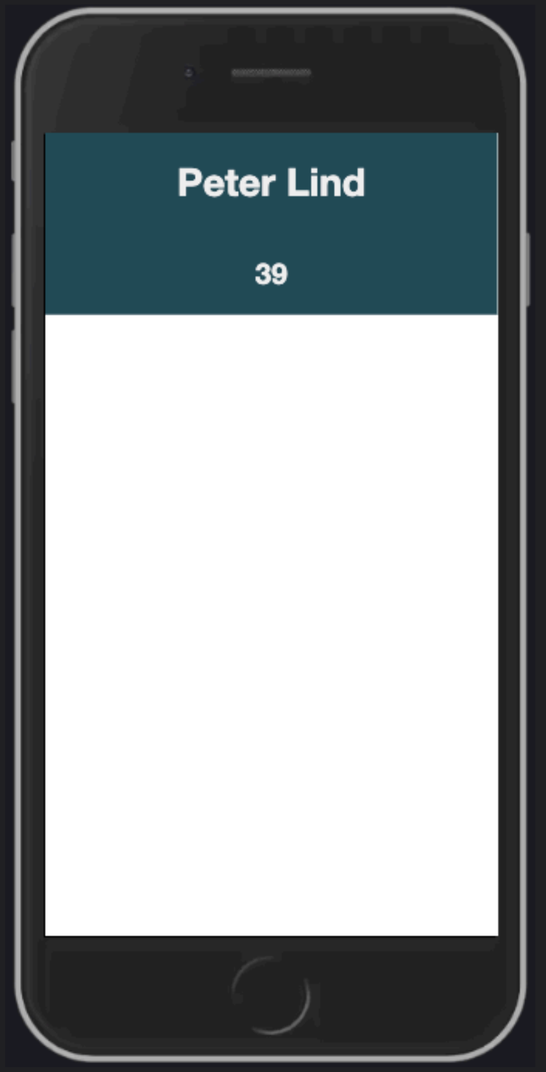
.textContent

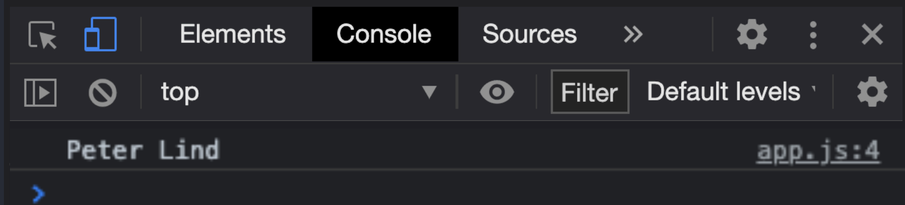
Get or set the text content of a given (HTML) element

https://www.w3schools.com/jsref/prop_node_textcontent.asp

Variables & UI

```
let fullName = "Peter Lind";
let age = 39;
console.log(fullName, age);
document.querySelector("#fullName_container").textContent = fullName;
document.querySelector("#age_container").textContent = age;
```

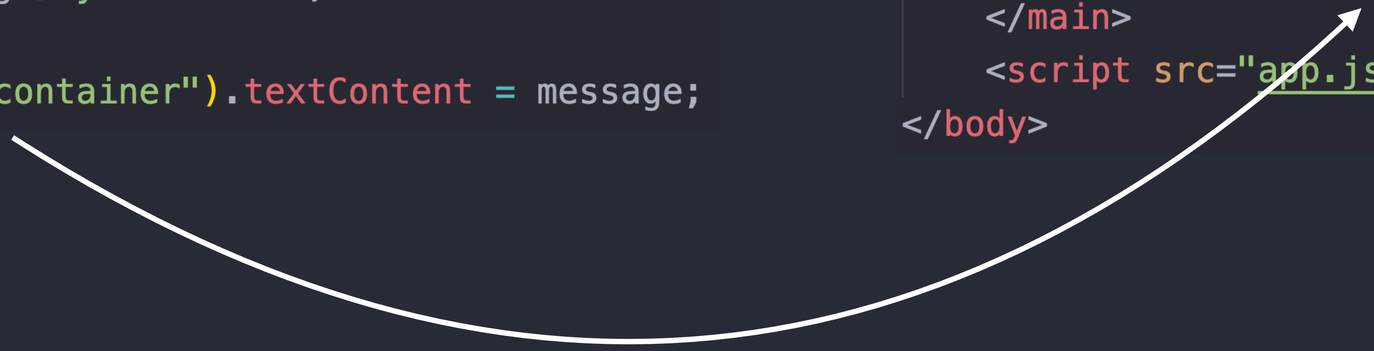




DOM Manipulation

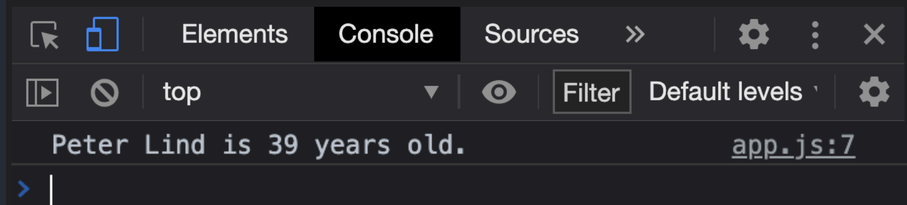
```
let fullName = "Peter Lind";
let age = 39;

let message = `${fullName} is ${age} years old.`;
console.log(message);
document.querySelector("#message_container").textContent = message;
```

```
<body>
    <main>
        <p id="message_container"></p>
    </main>
    <script src="app.js"></script>
</body>
```



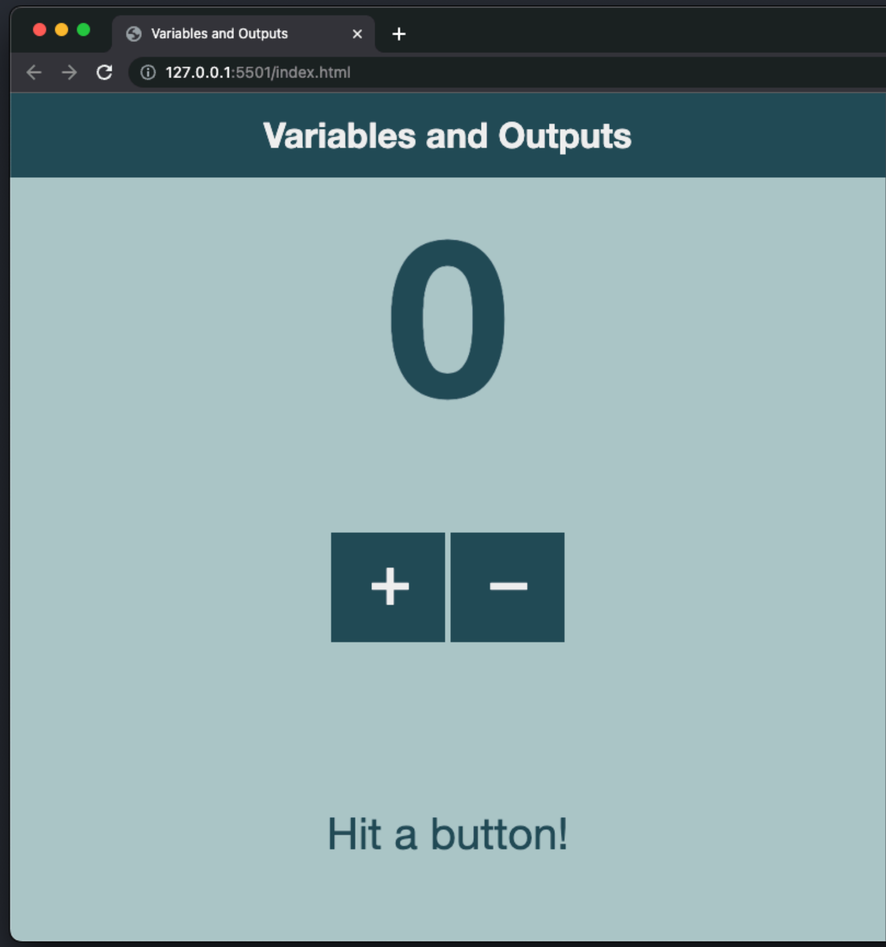
DOM Manipulation

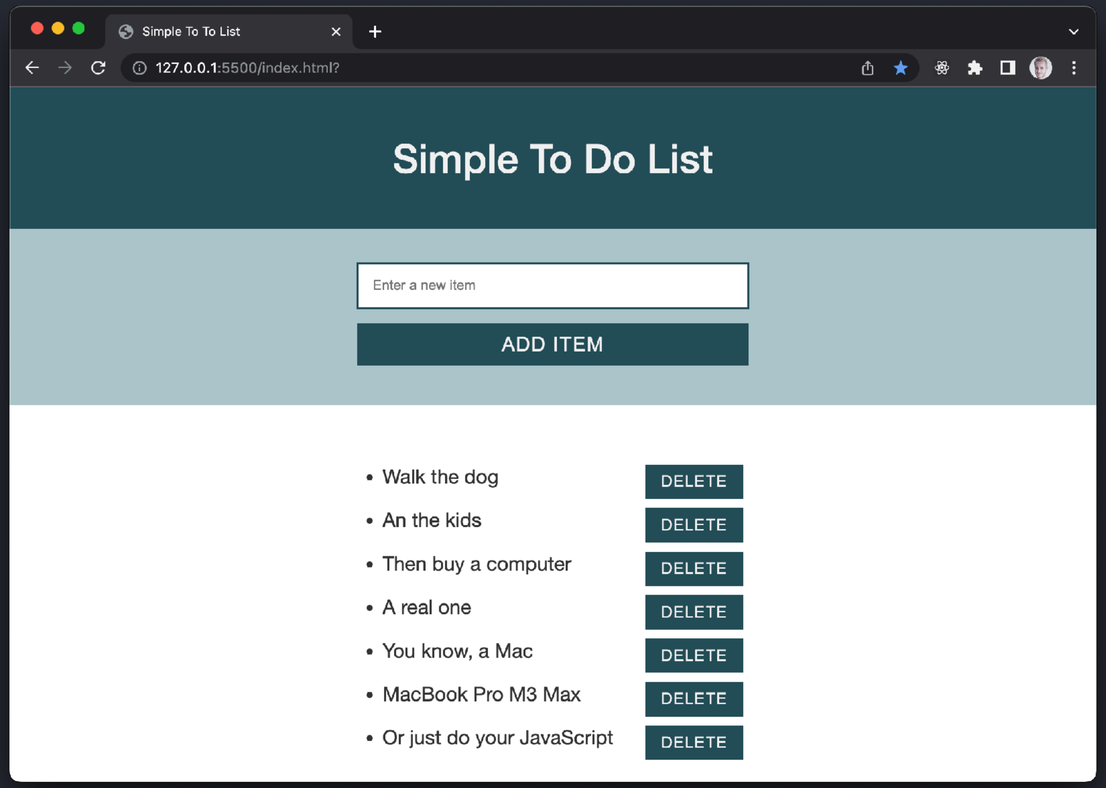
```
let fullName = "Peter Lind";
let age = 39;

let message = `${fullName} is ${age} years old.`;
console.log(message);
document.querySelector("#message_container").textContent = message;
```



Udskrive til DOM'en

```
let variableName = 0;
variableName++; //lægger 1 til den nuværende værdi
variableName--; //trækker 1 fra den nuværende værdi
variableName += x; //lægger værdien x til den nuværende værdi
variableName -= x; //trækker værdien x fra den nuværende værdi

document.querySelector("#mitElement").textContent = variableName;
```





DOM Manipulation Simple To Do List

Link til øvelse

document.createElement()

```
// create a new h1 element
const newTitle = document.createElement("h1");

// and give it some content
newTitle.textContent = "Hi there and greetings!";

// add the newly created element into the DOM
document.querySelector("body").appendChild(newTitle);
```

.createElement

Creates an HTML element specified by tagName.

.appendChild

Appends an element as last child of an element.

https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement
https://developer.mozilla.org/en-US/docs/Web/API/Node/appendChild

insertAdjacentHTML(position, html)

```
// 1

// create a new h1 with content using a backtick
const newTitle = /*html*/ `<h1>Hi there and greetings!</h1>`;

// add the newly created element into the DOM
document.querySelector("body").insertAdjacentHTML("beforeend", newTitle);

// 2

// variable holding username
const username = "RACE";

// create a new p with content using a backtick and username variable
const newParagraph = /*html*/ `<p>Welcome, ${username}!</p>`;

// add the newly created element into the DOM
document.querySelector("body").insertAdjacentHTML("beforeend", newParagraph);
```

Backtick string

Create multiline strings (templates) with embedded expressions and tags.

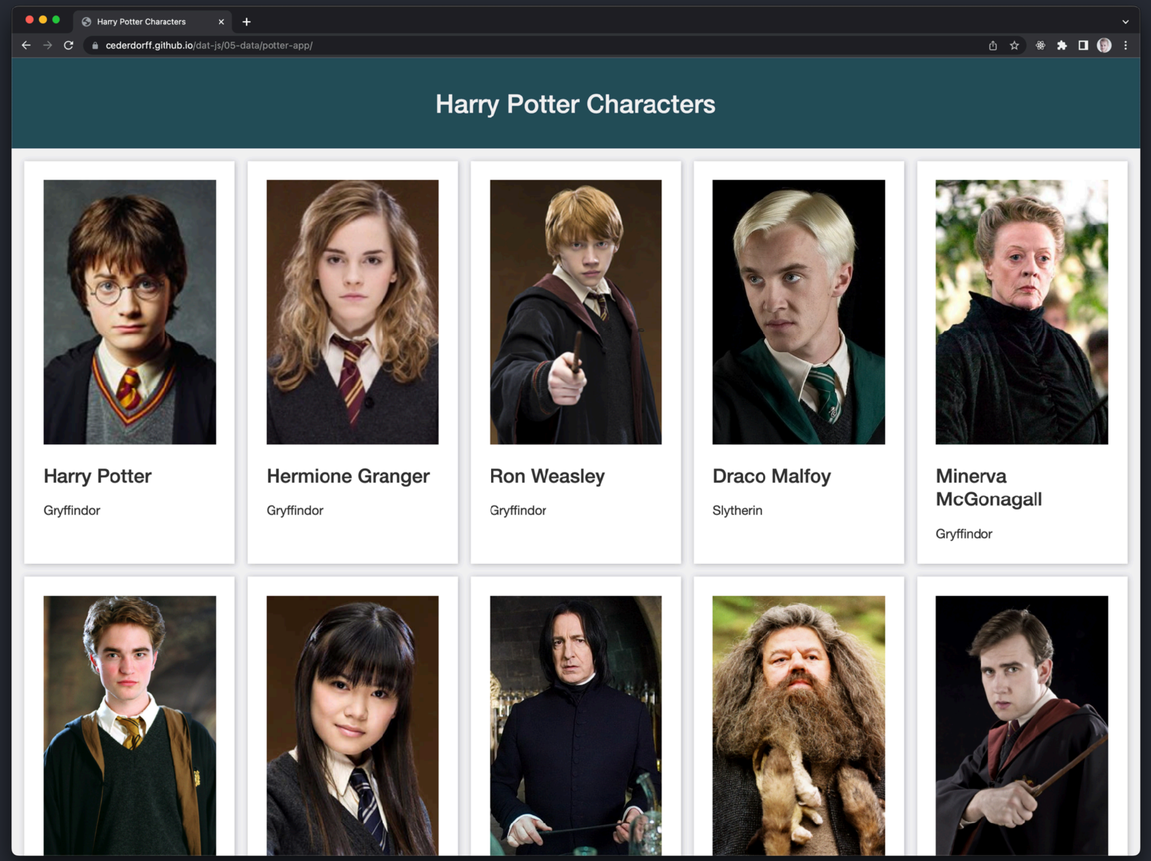
.insertAdjacentHTML

Inserts HTML into a specified position.

https://www.w3schools.com/js/js_string_templates.asp
https://www.w3schools.com/jsref/met_node_insertadjacenthtml.asp

DOM Manipulation
Potter App

Link til øvelse



The DOM & DOM Manipulation

- Manipulating documents: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents
- Introduction to the DOM: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- Using the Document Object Model: https://developer.mozilla.org/en-US/docs/Web/API/Document_object_model/Using_the_Document_Object_Model
- Modifying the document: https://javascript.info/modifying-document

Opgaver

- Simple To Do List
- DOM Manipulation - Potter App
- Active learning: Basic DOM manipulation
- Active learning: A dynamic shopping list


Code
Every
Day