
00200

Loops

... meget mere end bare `.forEach(…)`

Datamatiker

KØBENHAVNS ERHVERVSAKADEMI

Agenda

- Status
  - Hvad kan vi nu, og hvor skal vi hen?
- Loops og iterating
  - forEach
  - For
  - For Of
  - For In
  - While
- Kort om Array-metoder
  - .filter
  - .find
  - .sort

1. SEMESTERS STRUKTUR

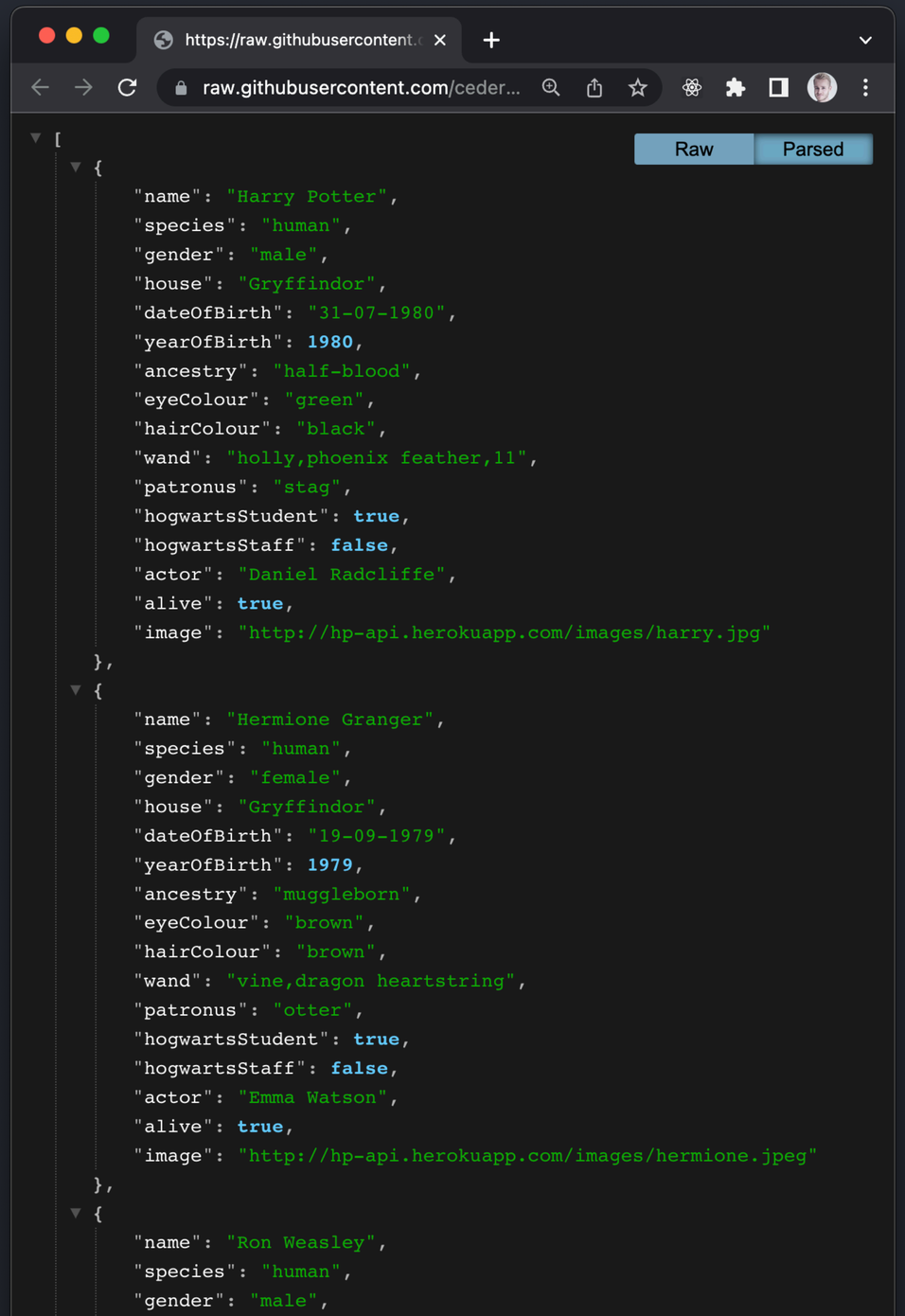
Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)

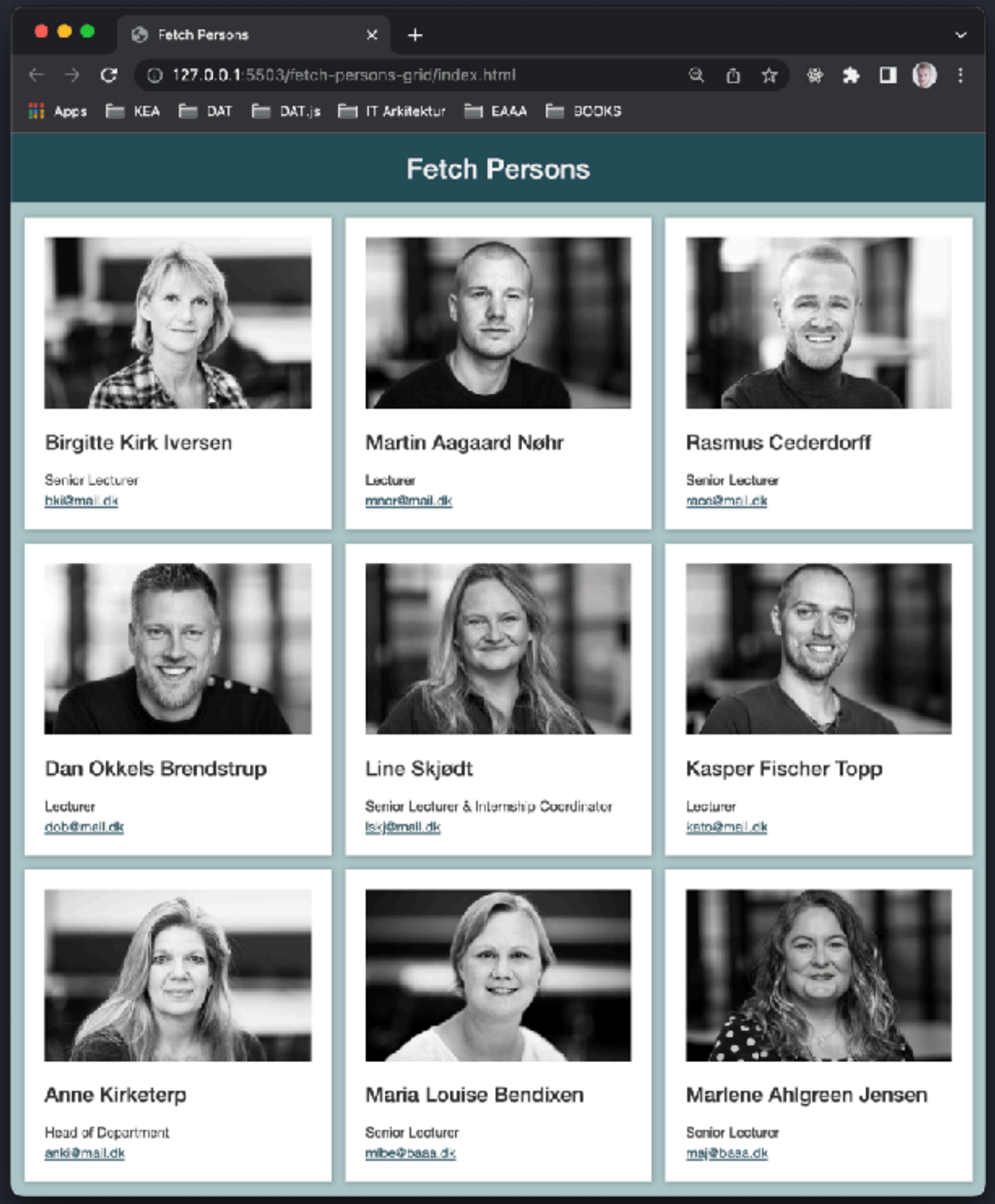
Eksamen ligger 12.-14. juni

kea
KØBENHAVNS ERHVERVSAKADEMI

Data

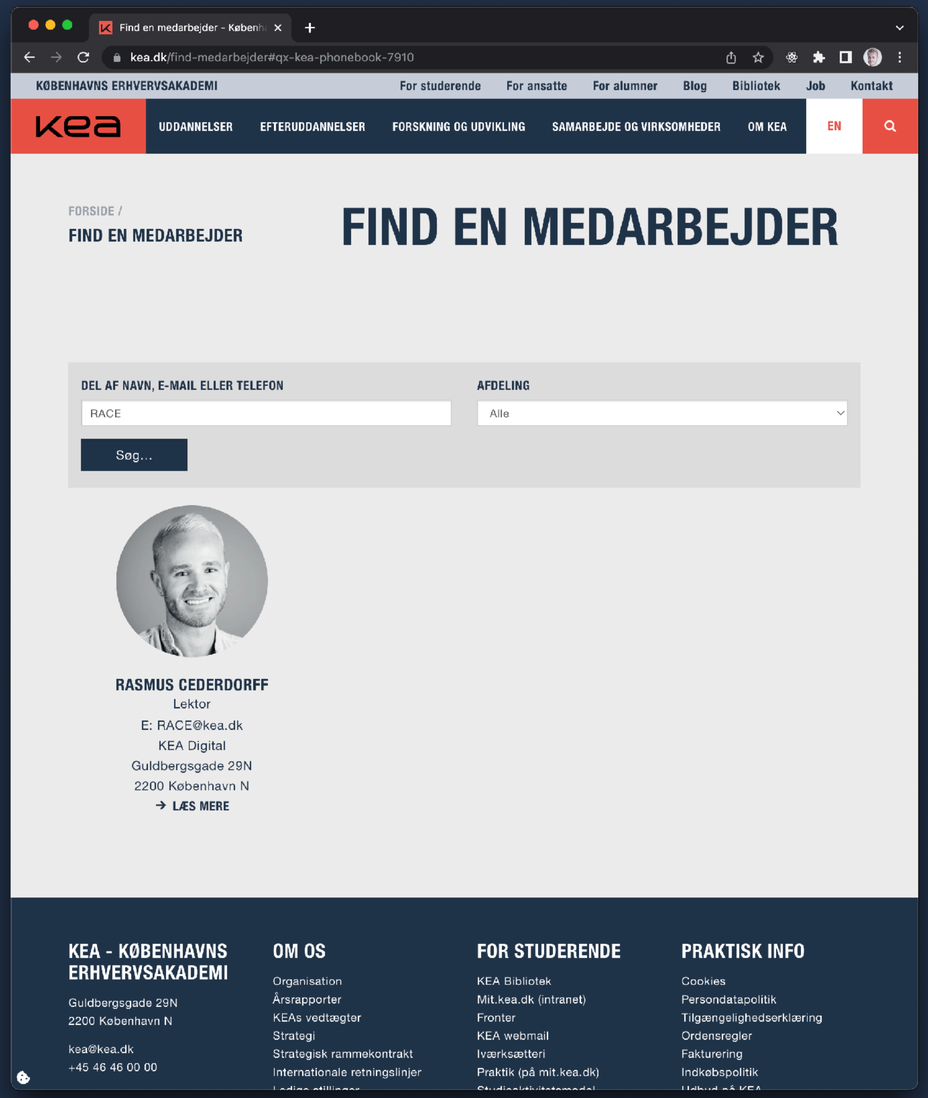
JavaScript, lister, objekter, JSON, fetch, DOM-manipulation og meget mere…





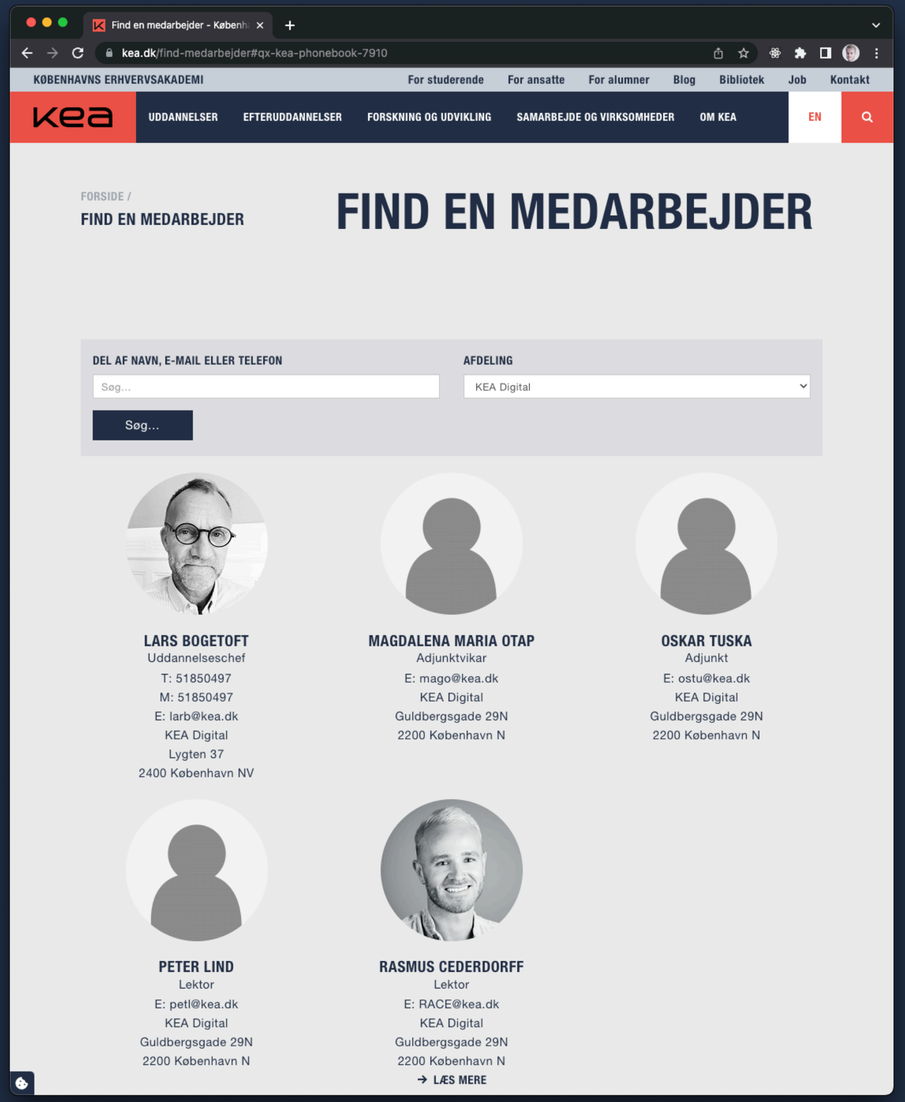
Forståelse for objekter og arrays

Hvad ser du?

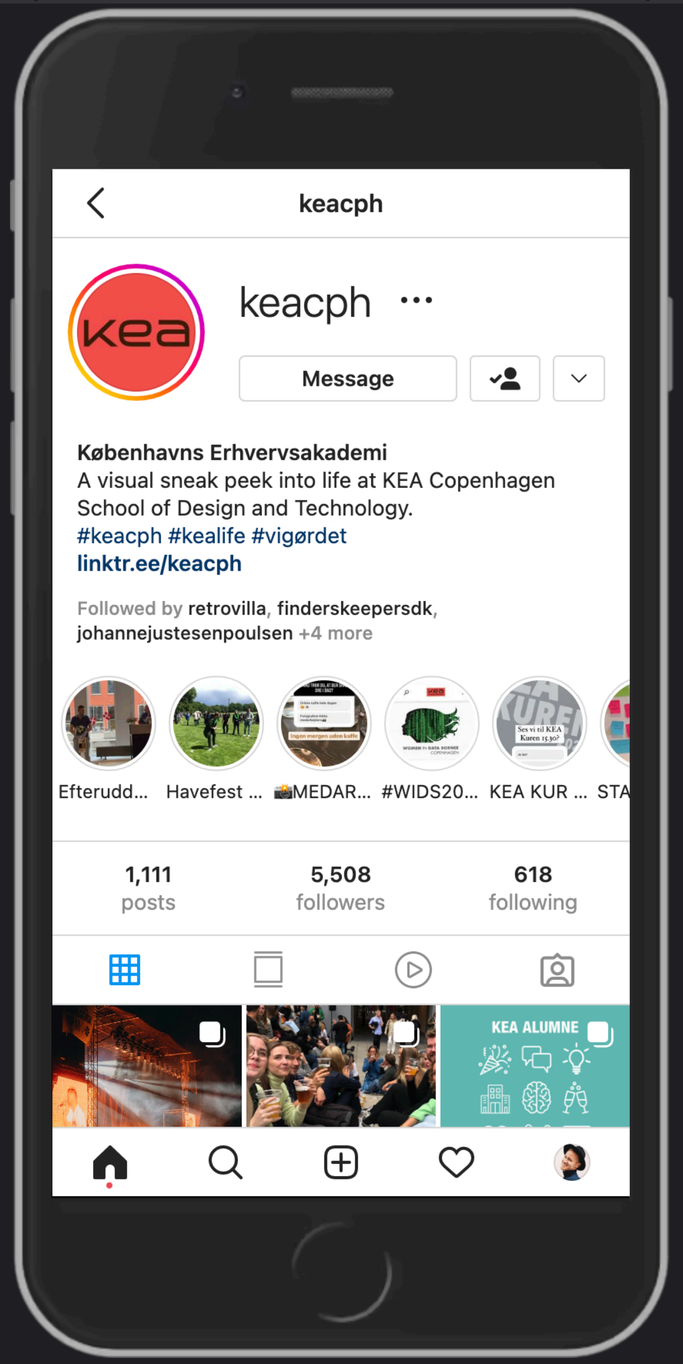



```
const race = {
    name: "Rasmus Cederdorff",
    position: "Lektor",
    mail: "race@kea.dk",
    department: "KEA Digital",
    address: "Guldbergsgade 29N, 2200 København N",
    image: "https://kea.dk/slir/w180-c1x1/images/us
};
```

Hvad ser du?




```
const lecturers = [
    {
        name: "Peter Lind",
        position: "Lektor",
        mail: "petl@kea.dk",
        department: "KEA Digital",
        address: "Guldbergsgade 29N, 2200 København
        image: "https://share.cederdorff.com/images/
    },
    {
        name: "Rasmus Cederdorff",
        position: "Lektor",
        mail: "race@kea.dk",
        department: "KEA Digital",
        address: "Guldbergsgade 29N, 2200 København
        image: "https://kea.dk/slir/w180-c1x1/images
    }
];
```



Dimensions: iPhone ... 375 x 667 100%

Elements Components Console Performance insights Recorder Sources Network

Preserve log Disable cache No throttling

Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

Has blocked cookies Blocked Requests 3rd-party requests

2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms 14000 ms

Name

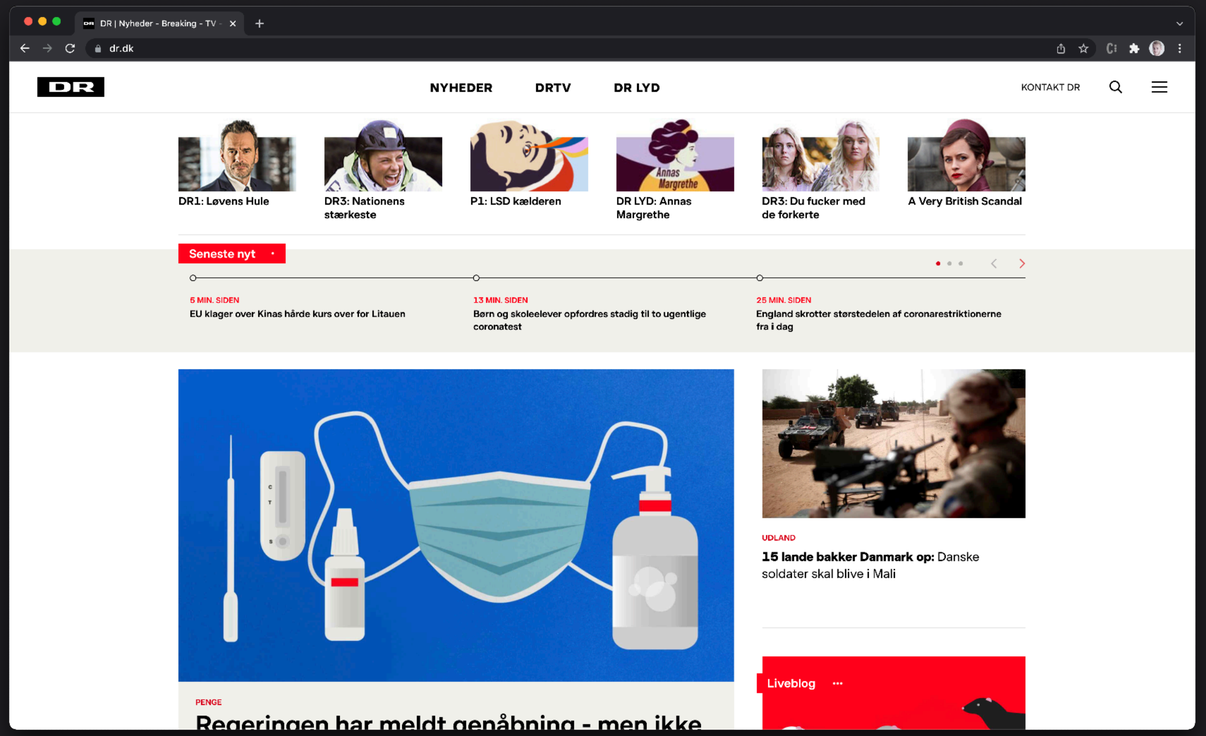
?modules=PolarisBD...
?content_type=PROF...
?username=keacph
reels_tray/
timeline/
ig_sso_users/
logging_client_events
bz?__d=dis
falco
bz?__a=1&__ccg=G...
batch_fetch_web/
get_encrypted_crede...
highlights_tray/
?target_user_id=140...
badge/
story/
bulk-route-definitions/
bulk-route-definitions/
?query_hash=69cba4...
bz?__a=1&__ccg=G...
bz?__a=1&__ccg=G...
bulk-route-definitions/
bz?__a=1&__ccg=G...
logging_client_events
bz?__d=dis

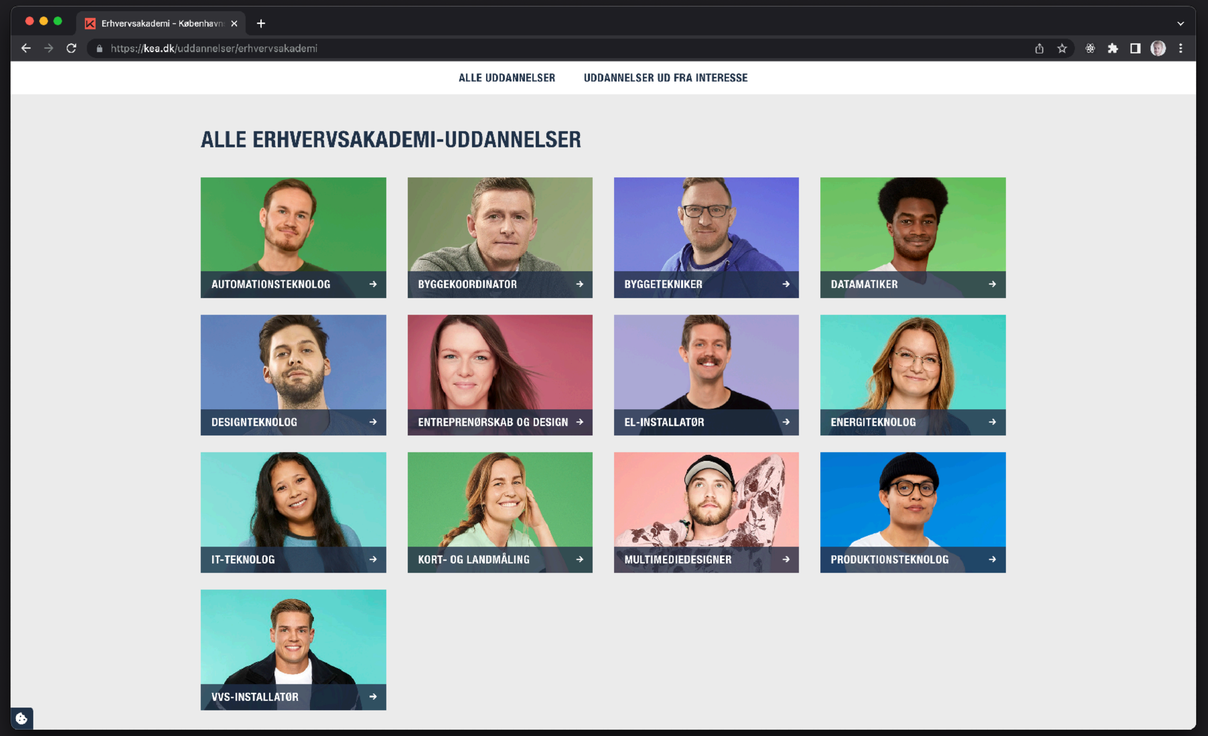
26 / 96 requests 257 kB

Headers Payload Preview Response Initiator Timing Cookies

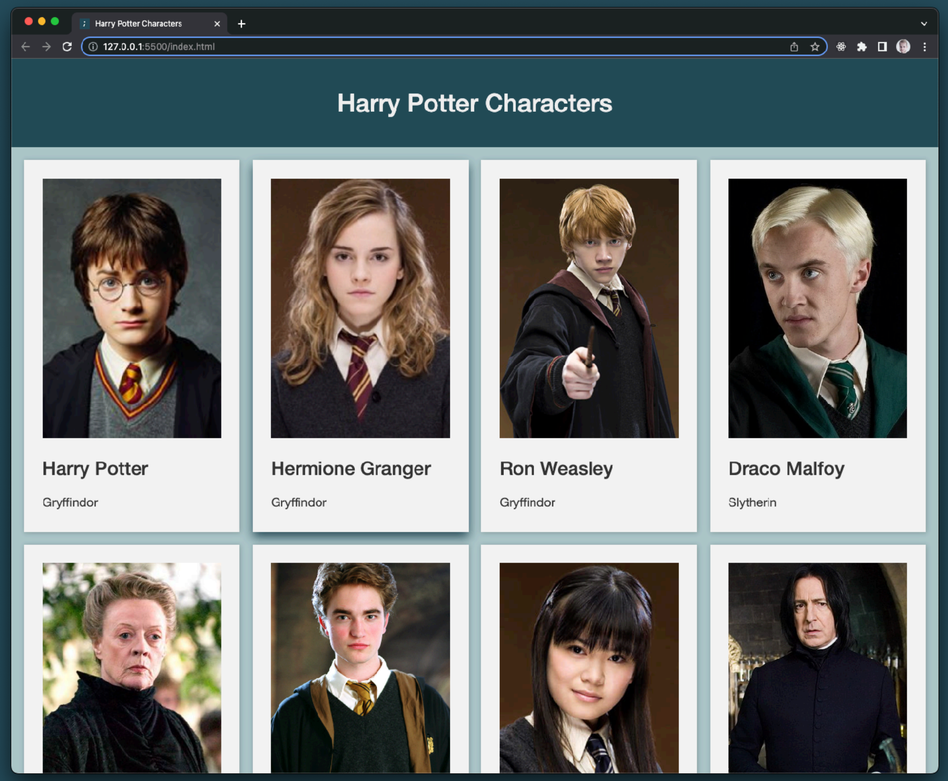
{data: {user: {,…}}, status: "ok"}
data: {user: {,…}}
user: {,…}
bio_links: []
biography: "A visual sneak peek into life at KEA Copenhagen School of Design and Technology. \n#keacph
biography_with_entities: {,…}
blocked_by_viewer: false
business_address_json: null
business_category_name: "Home Services"
business_contact_method: "CALL"
business_email: null
business_phone_number: null
category_enum: "LOCAL"
category_name: null
connected_fb_page: null
country_block: false
edge_felix_video_timeline: {count: 116, page_info: {has_next_page: true,…},…}
edge_follow: {count: 618}
edge_followed_by: {count: 5508}
edge_media_collections: {count: 0, page_info: {has_next_page: false, end_cursor: null}, edges: []}
edge_mutual_followed_by: {count: 7, edges: [{node: {username: "retrovilla"}}, {node: {username: "finde
edge_owner_to_timeline_media: {count: 1111, page_info: {has_next_page: true,…},…}
edge_saved_media: {count: 0, page_info: {has_next_page: false, end_cursor: null}, edges: []}
external_url: "https://linktr.ee/keacph"
external_url_linkshimmed: "https://l.instagram.com/?u=https%3A%2F%2Flinktr.ee%2Fkeacph&e=ATNF7APjyvCLD
fbid: "17841400899781985"
followed_by_viewer: true
follows_viewer: true
full_name: "Københavns Erhvervsakademi"
guardian_id: null
has_ar_effects: false
has_blocked_viewer: false
has_channel: false
has_clips: false

It’s all objects & arrays!



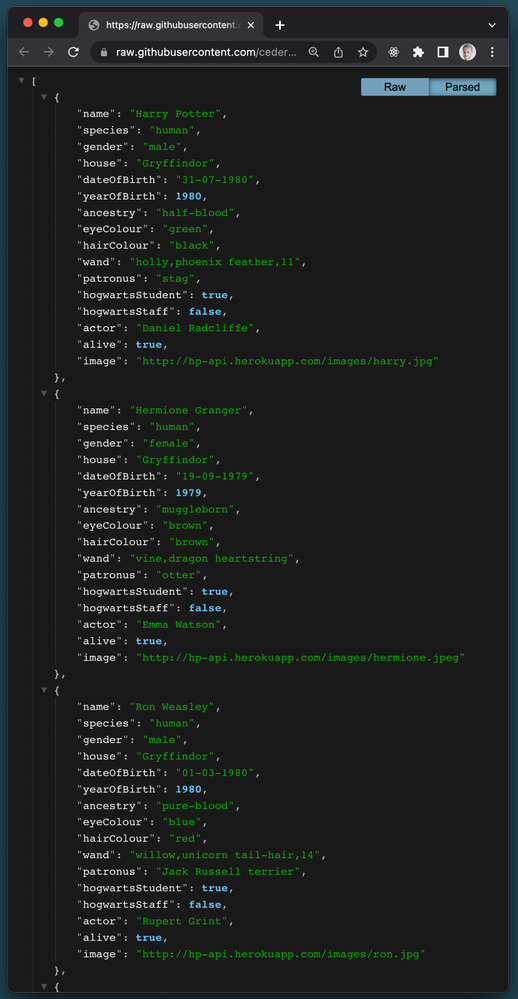


Frontend (client)



https://cederdorff.github.io/potter-app/

JSON Data Source (Server)



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

fetch(...)

HTTP Requests in JavaScript.

A way to get and post data from and to data sources.

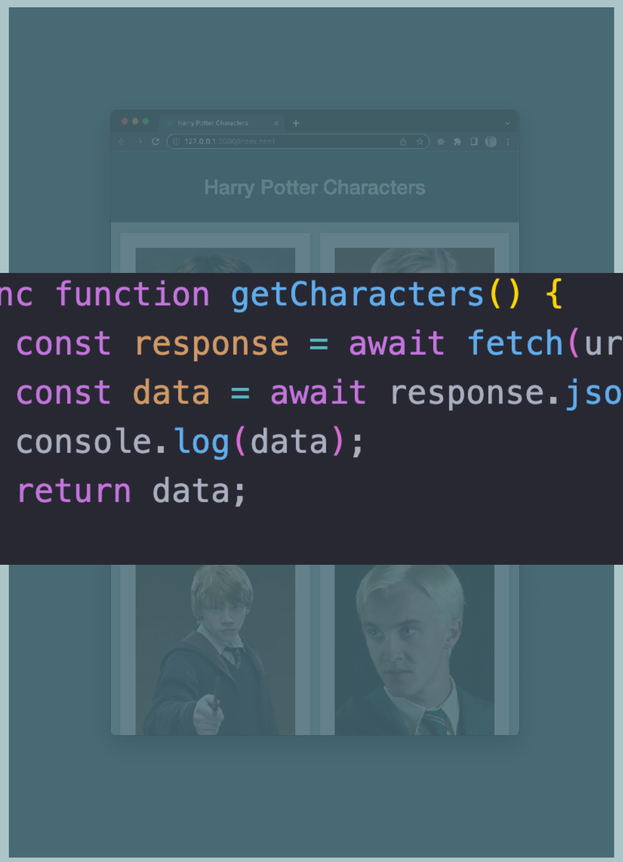
`getCharacters` fetches a list of characters from a JSON data source, parses the JSON to JS and returns the data.

```
async function getCharacters() {
    const response = await fetch(url);
    const data = await response.json();
    console.log(data);
    return data;
}
```

Fetch, HTTP Request & Response

WEB APP

DATA SOURCE




```
async function getCharacters() {
    const response = await fetch(url);
    const data = await response.json();
    console.log(data);
    return data;
}
```

FETCH

REQUEST (GET)

JSON DATA

RESPONSE

potters

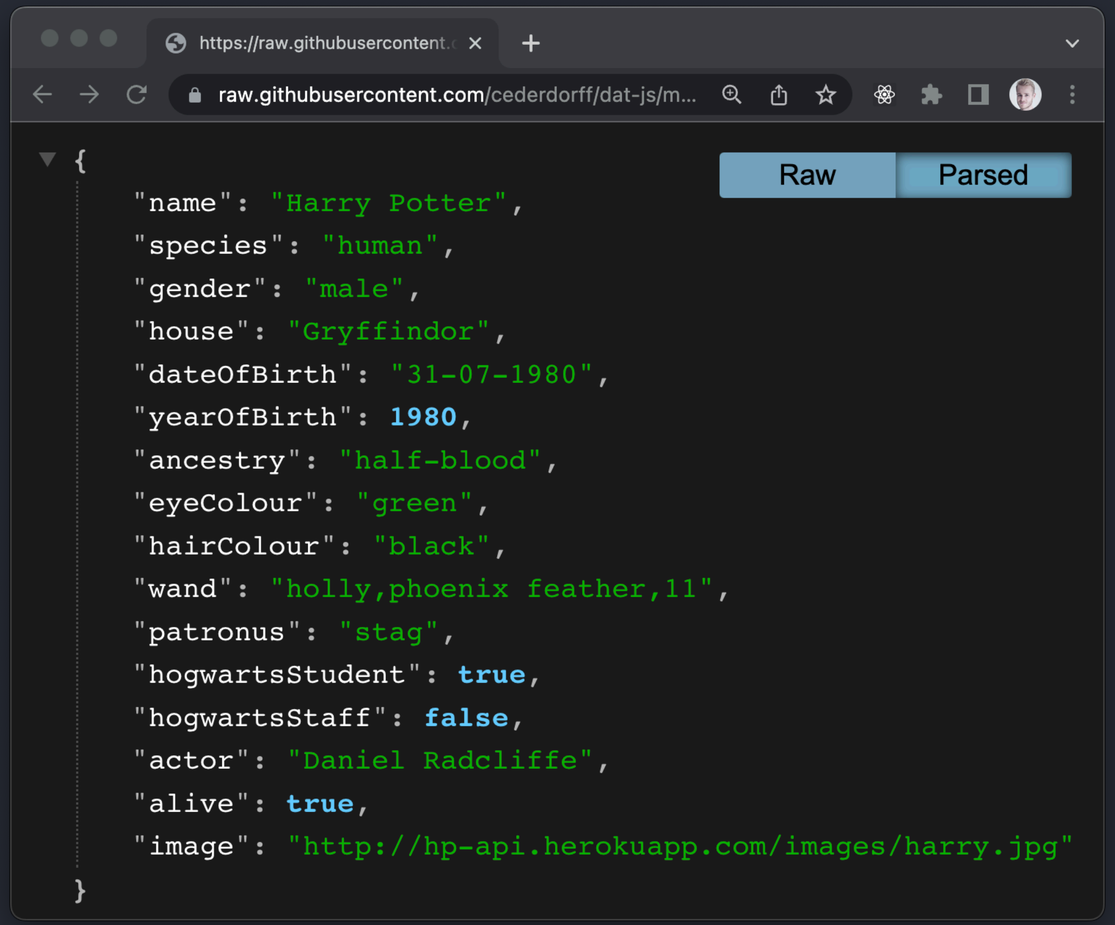
pokemons

southpark

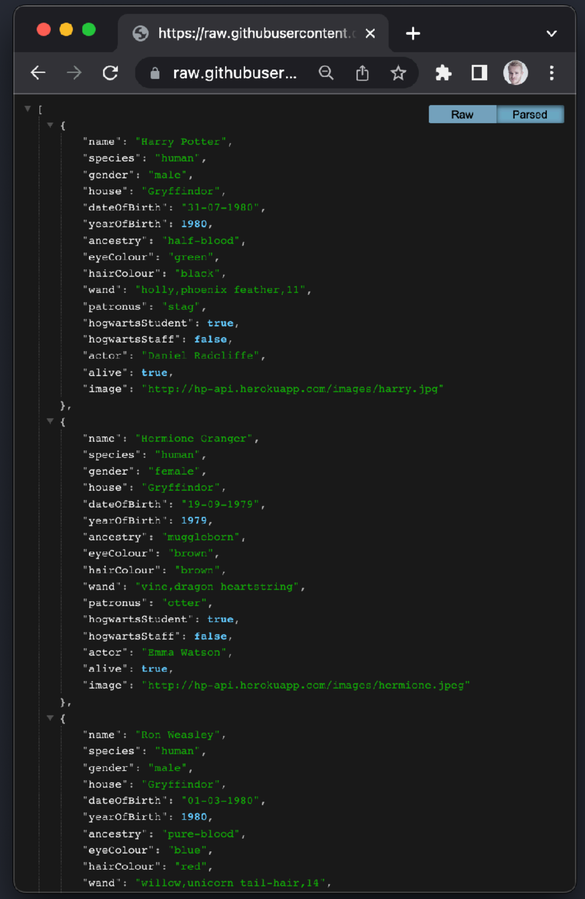
CLIENT

SERVER

JSON



JSON OBJECT



LIST OF JSON OBJECTS

https://www.instagram.com/p/CVqbCzgsZUF/

Frontend (client)

JSON Data Source (Server)

Så hvad kan *du* så nu?

https://cederdorff.github.io/potter-app/

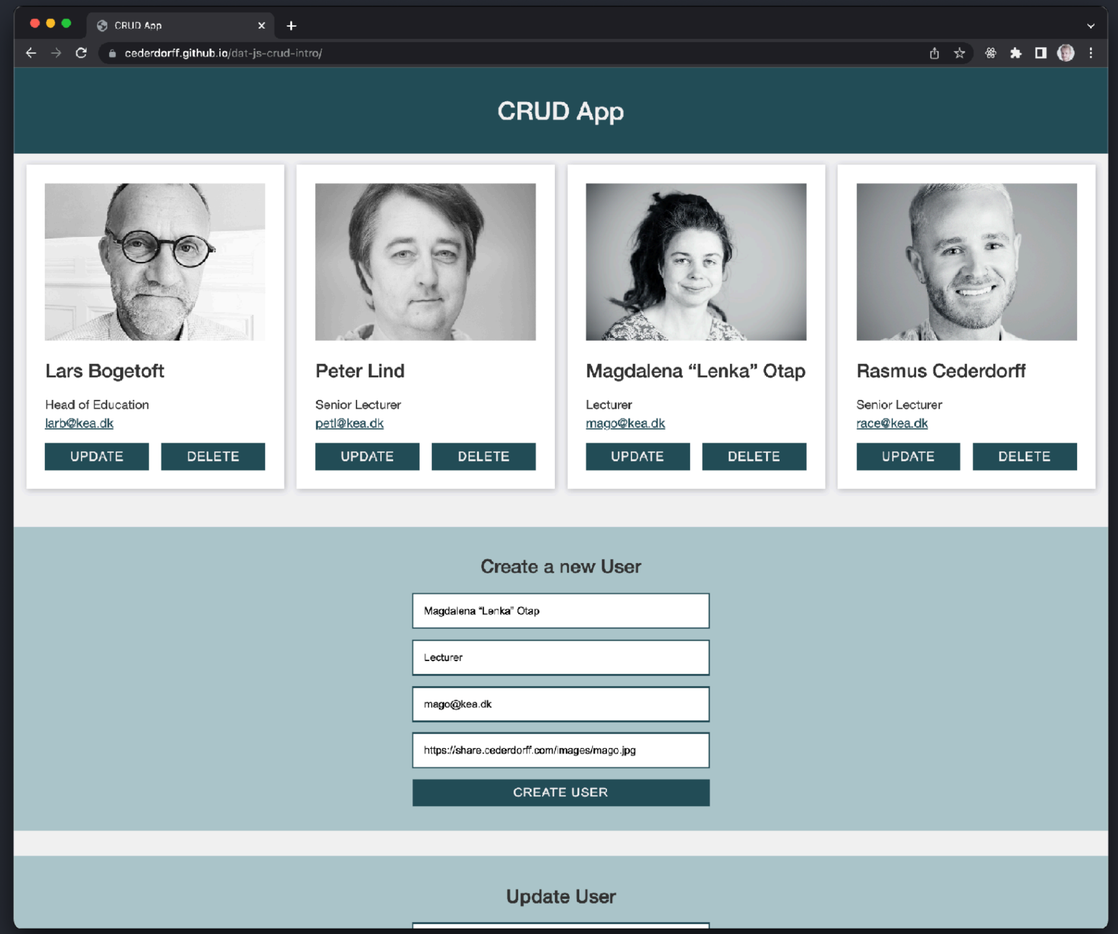
https://raw.githubusercontent.com/cederdorff/dat-js/main/data/potter.json

Og, hvor skal vi hen, *du*?

Forms, REST & CRUD

Formularer, API'er, REST, Fetch, HTTP-metoder, CRUD, fejlhåndtering & validering

CRUD-projekt



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

Frontend (client)

JSON Data Source (Server)

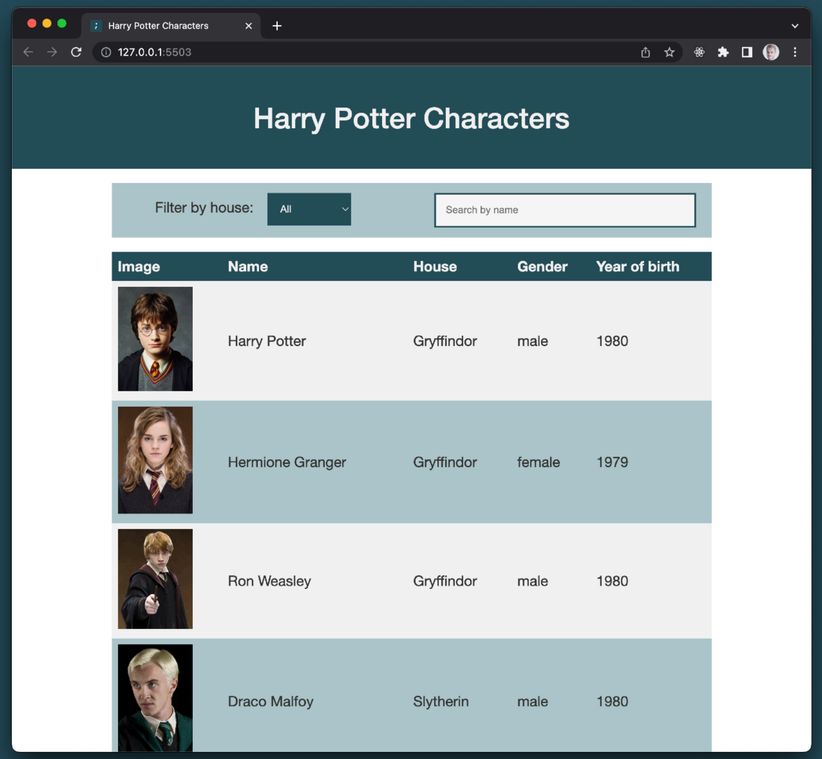
Hvad mangler du?

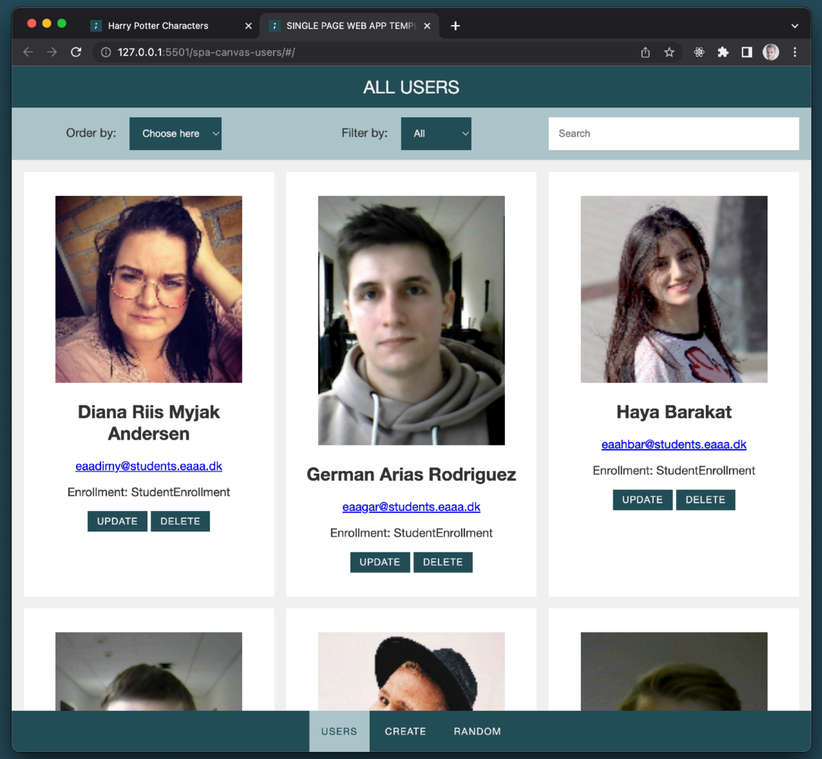
Er der noget, du gerne ville kunne have gjort med dit data-projekt, som du ikke kunne?

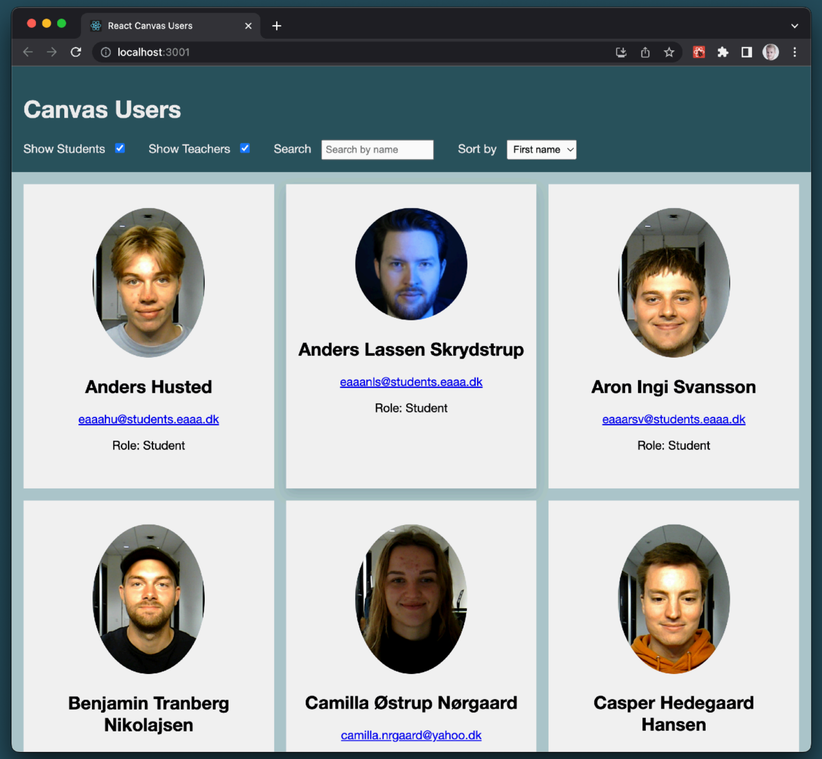
https://cederdorff.github.io/potter-app/

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

Sortering, filtrering, søgning ...







Loops

Iterate over arrays or other iterable objects.

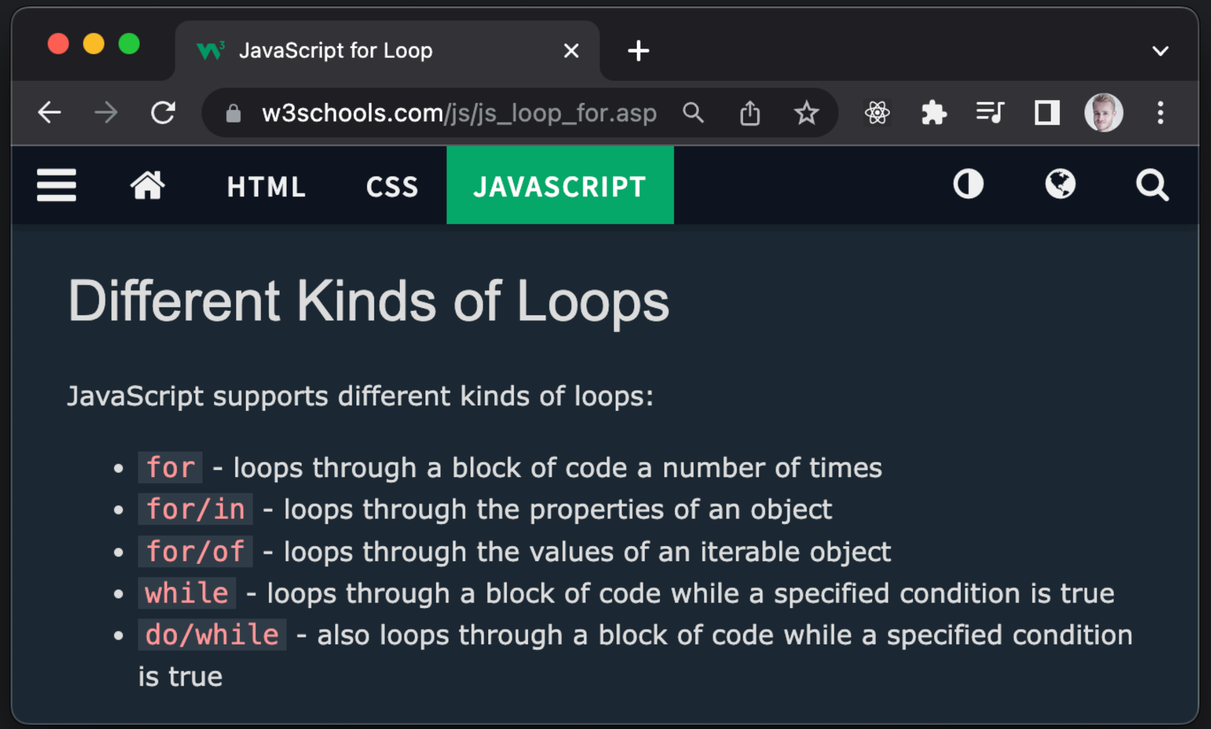
Ex loop through an array of objects.

```
// .forEach
characterList.forEach(showCharacter);

// for of loop
for (const character of characterList) {
    showCharacter(character);
}

// for loop
for (let index = 0; index < characterList.length; index++) {
    const character = characterList[index];
    showCharacter(character);
}
```

Loops

Loops can execute a block of code a number of times.

There are a few ways to do that in JavaScript 🤷🏼



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

.forEach

an array method

```
const names = ["Peter", "Oskar", "Lenka", "Rasmus"];

names.forEach(showName);

function showName(name) {
    console.log(name);
}
```

You can use it on an array to do *something* for each element in the array.

`forEach` will call a function for each element in an array.

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

forEach only iterates over an array

You can only have one kind of "loop" with forEach, as the name implies:

one that does something for each item in an array

You can't make it stop half-way – or make it run a certain number of times – or even run without an array.

But other kinds of loops can!

For

Loop

```
const names = ["Peter", "Lenka", "Oskar", "Rasmus"];

for (let index = 0; index < names.length; index++) {
    const name = names[index];
    console.log(name);
}
```

You can use it to loop over an array.

A `for loop` repeats until a specified condition is false.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration#for_statement
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for
https://www.w3schools.com/js/js_loop_for.asp

for to the rescue

“all in one”

sets up a counter

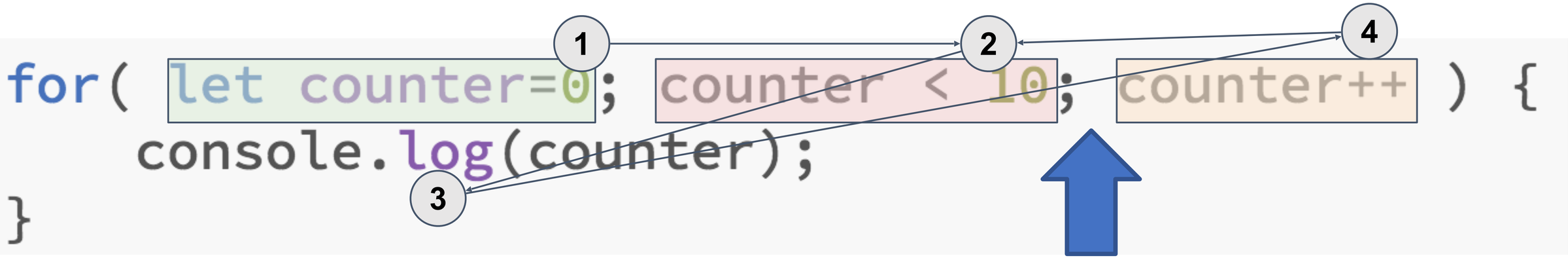
checks the counter

increments the counter

```
for( let counter=0; counter < 10; counter++ ) {
    console.log(counter);
}
```

for loop flow

It is a bit complicated



The importantest thing to remember is:

The code is executed *after* the check, but *before* the increment!

The scope is also a bit weird

What happens?

```
for( let counter=0; counter < 10; counter++ ) {
    console.log(counter);
}

console.log(`after th                nter is ${counter}`);
```

This is the block-scope of the counter

Even though it was created outside the block ...

Try it!

For of

Loop

```
const names = ["Peter", "Lenka", "Oskar", "Rasmus"];

for (const name of names) {
    console.log(name);
}
```

You can use it to loop over an array to do *something* for every element.

`for of` loops through the values of an iterable object (an array is an iterable object).

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

For of loop

Iterates over arrays or other iterable objects

https://scrimba.com/learn/introductiontojavascript/for-loops-cMMM8U9

https://scrimba.com/learn/introductiontojavascript/challenge-for-loops-cPkpJrcv

For in

Loop

```
const person = {
    name: "Peter",
    mail: "petl@kea.dk",
    age: 45
};

for (const key in person) {
    console.log(key);
    const value = person[key];
    console.log(value);
}
```

`for in` loops through all properties of an Object.

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

While

Loop

```
const names = ["Peter", "Oskar", "Lenka", "Rasmus"];

let index = 0;

while (index < names.length) {
    const name = names[index];
    console.log(name);
    index++;
}
```

Loops executes a block of code as long as a specified condition is true.

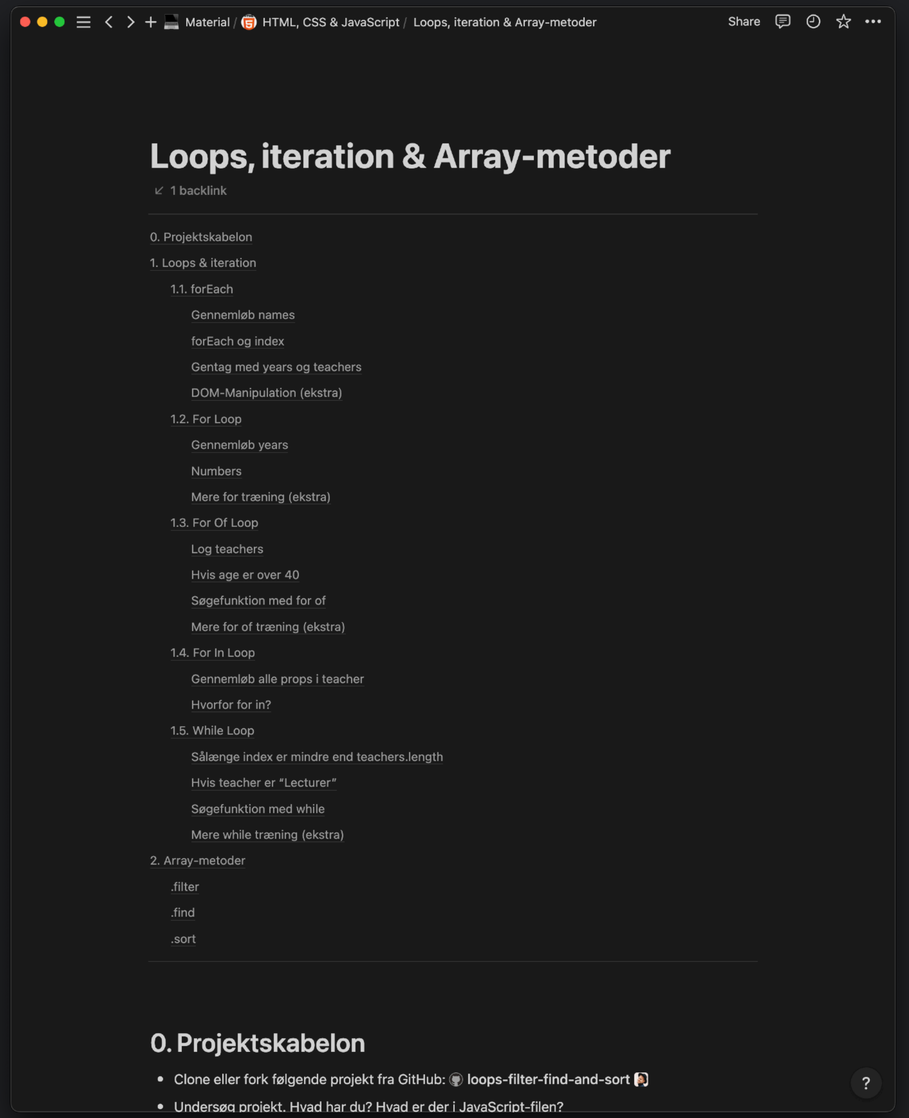
The while loop loops through a block of code as long as a specified condition is true.

... while the index is less than the length of the array, execute...

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

Loops, iteration & Array-metoder

Jeg har samlet lidt lækkert til jer 🎉

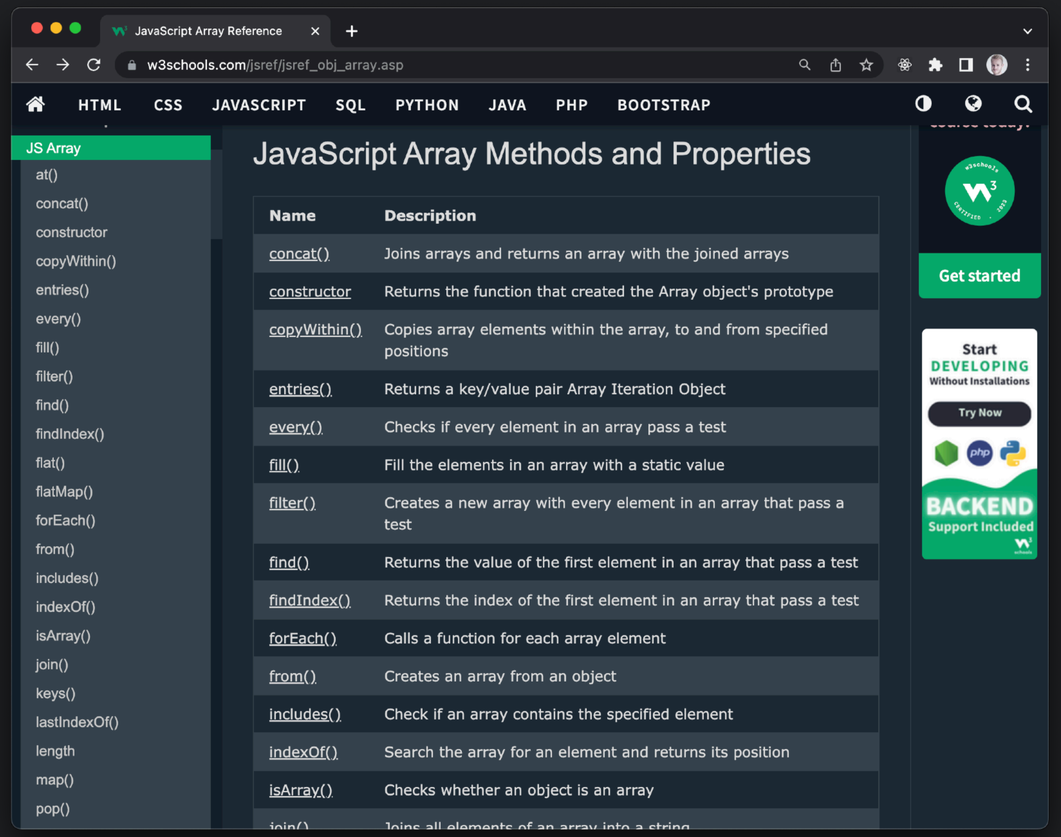


Loops, iteration & Array-metoder

Array Reference

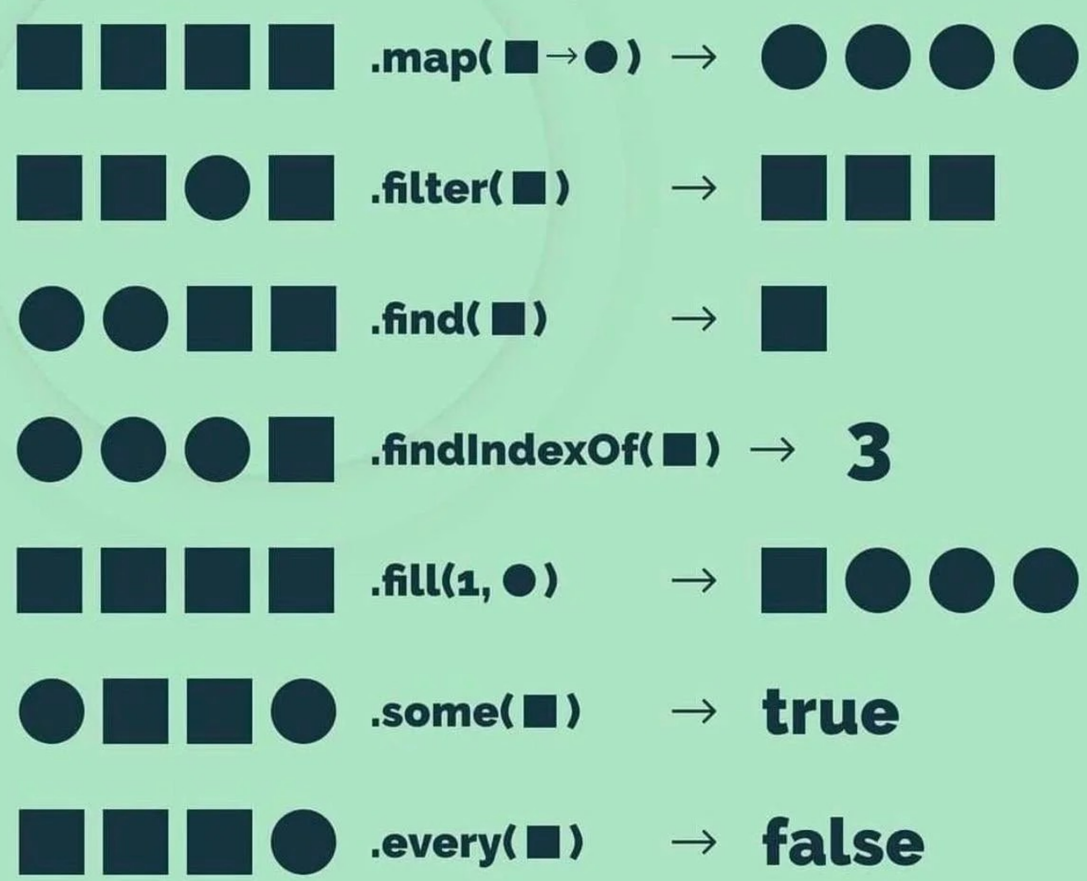
Array Methods

Der er mange, og vi skal nok vende tilbage til nogle af dem.



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

Array Methods



https://javascript.info/array-methods
https://medium.com/@mandeepkaur1/a-list-of-javascript-array-methods-145d09dd19a0

.filter()

Array Method

```
const users = [
    { name: "John", age: 35 },
    { name: "Pete", age: 40 },
    { name: "Mary", age: 44 }
];

// returns an array with users over the age 39
const results = users.filter(checkAge);

function checkAge(user) {
    return user.age > 39;
}

console.log(results);
```

.filter() creates a new array with elements that pass a test provided by a function.

The filter() method does not execute the function for empty elements.

The filter() method does not change the original array.

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

.find()

Array Method

```
const users = [
    { name: "John", age: 35 },
    { name: "Pete", age: 40 },
    { name: "Mary", age: 44 }
];

// returns an object with the
// first matcing user of the age 40
const result = users.find(matchAge);

function matchAge(user) {
    return user.age === 40;
}

console.log(result);
```

find() returns the value of the first element that passes a test.

find() executes a function for each array element.

find() returns undefined if no elements are found.

find() does not execute the function for empty elements.

find() does not change the original array.

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

.sort()

Array Method

```
const names = ["Peter", "Magdalena", "Frederikke", "Anders"];
names.sort(); // sorts the original array with strings
console.log(names);
```

```
const years = [2003, 2032, 1990, 1989, 1975, 2023, 1933];
years.sort(); // sorts the original array of numbers
console.log(years);
```

.sort() sorts the elements of an array.

.sort() overwrites the original array.

.sort() sorts the elements as strings in alphabetical and ascending order.

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

.sort()

Array Method

```
const users = [
    { name: "John", age: 35 },
    { name: "Pete", age: 40 },
    { name: "Mary", age: 44 }
];

// compare and sort by the age prop
users.sort(compareAge);

function compareAge(teacher1, teacher2) {
    return teacher1.age - teacher2.age;
}

// compare and sort by the name prop
users.sort(compareName);

function compareName(teacher1, teacher2) {
    return teacher1.name.localeCompare(teacher2.name);
}
```

If you need to sort an array of objects, you must provide a compare function.

The compare function defines how to compare the objects.

.sort() uses the compare function to sort the original array.

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

Code Every Day

```
while (alive) {
        eat();
        sleep();
        code();
        repeat();
}
```