

## Dokumentasjon for prosjektgruppe 6

Github-repository: [https://github.com/Eirik00/Gruppe6\\_UIN24\\_WhatToSee.git](https://github.com/Eirik00/Gruppe6_UIN24_WhatToSee.git)

Som utgangspunkt ønsker vi i gruppen å minst oppnå karakter c.

Github-alias for gruppemedlemmer:

MartinInformatikk – Martin Malo Treseng

Eirik00 – Eirik Rasmussen



### Project invitations

+ Invite project members

Invitations that have been sent, but not accepted yet. Changed your mind? Pending invitation may be revoked.

2 of 20  
non-viewer members  
(20 included in plan)

[Learn more about roles](#)

| EMAIL                     | ROLE          | INVITED  |     |
|---------------------------|---------------|--|-----|
| marinus.akerbak@gmail.com | Administrator |  0s | ... |
| ann.c.karlsen@hiof.no     | Administrator |  0s | ... |

Kilder:

<https://stackoverflow.com/questions/6712034/sort-array-by-firstname-alphabetically-in-javascript>

<https://stackoverflow.com/questions/8217419/how-to-determine-if-a-javascript-array-contains-an-object-with-an-attribute-that>

<https://www.w3schools.com/css/default.asp> (brukt flere ulike deler av W3schools ved utvikling av CSS)

in javascript, if i have two arrays of objects and i only want the ones that are the same in both lists, how would i do that?



You can achieve this in JavaScript using various methods. One common approach is to use the `filter()` method along with the `some()` method or `includes()` method to check for common elements between the arrays. Here's a sample implementation:

```
javascript Kopier kode

// Sample arrays of objects
let array1 = [{ id: 1, name: "John" }, { id: 2, name: "Jane" }, { id: 3, name: "Doe" }];
let array2 = [{ id: 2, name: "Jane" }, { id: 3, name: "Doe" }, { id: 4, name: "Alice" }];

// Function to find common objects between arrays
function findCommonObjects(arr1, arr2) {
    return arr1.filter(obj1 => arr2.some(obj2 => obj1.id === obj2.id));
}

// Find common objects
let commonObjects = findCommonObjects(array1, array2);

console.log(commonObjects); // Output: [{ id: 2, name: "Jane" }, { id: 3, name: "Doe" }]
```

In this code:

- The `findCommonObjects` function takes two arrays as input and filters the first array (`arr1`) based on whether any object in the second array (`arr2`) has the same `id`.
- The `filter()` method is used to iterate through the first array and return only those objects for which the condition specified by the `some()` method is true.
- The `some()` method checks if at least one element in the second array satisfies the condition specified by the callback function.
- We're assuming that the objects have an `id` property that can be used for comparison. You can adjust the comparison logic based on your object structure and requirements.

🗨️ 📄 🔄 🌟 📌 🔍