

“ Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.

—Martin Golding

Citaten verwijderen

- 1 Element vooraan of achteraan uit een array verwijderen
- 2 Element op een bepaalde plaats verwijderen.
- 3 Element zoeken in een array en verwijderen
- 4 `findIndex()` arraymethode
- 5 Element verwijderen met `findIndex`
- 6 Uitbreiding: PNG vervangen door SVG

1 Element vooraan of achteraan uit een array verwijderen

Soms wil je gewoon het eerste of laatste element van een array verwijderen.

Hernemen we als voorbeeld een kleurenarray zoals in [het hoofdstuk 'arrays 2'](#). We gebruiken de methodes `shift` en `pop`:

```
const kleuren = ['rood', 'geel', 'groen', 'blauw', 'wit', 'bruin']

// eerste element verwijderen
kleuren.shift()
console.log(kleuren) // geeft als resultaat ['geel', 'groen', 'blauw', 'wit', 'bruin']

// laatste element verwijderen
kleuren.pop()
console.log(kleuren) // resultaat is nu ['geel', 'groen', 'blauw', 'wit']
```

Dat is in dit geval eenvoudig. Maar wat als je een element *in het midden van de array* wil verwijderen? Dan moet je de index van het element kennen. Het element op die plaats verwijderen onderzoeken we in volgende sectie.

2 Element op een bepaalde plaats verwijderen.

Laten we terug van onze originele kleurenarray vertrekken. We verwijderen nu het element met index 3.

```
const kleuren = ['rood', 'geel', 'groen', 'blauw', 'wit', 'bruin']

// element met index 3 verwijderen
kleuren.splice(3, 1)
console.log(kleuren) // resultaat is ['rood', 'geel', 'groen', 'wit', 'bruin']
```

De methode `splice` heeft twee argumenten: de index van het te verwijderen element en het aantal elementen dat je wil verwijderen. In dit geval is dat 1. Denk eraan dat het getal met index 3 het *vierde* getal is!

3 Element zoeken in een array en verwijderen

Wat we in onze applicatie willen is nog iets moeilijker: we weten immers niet op welke index het te verwijderen element staat. In een eerste stap moeten we dus op zoek gaan naar de index van dit element. Dat zoeken zullen we twee keer implementeren:

- met een eigen `for`-lus;
- met de arraymethode `findIndex`.



Het eerste ‘verwijderfilmpje’ covert volgende topics:

- op zoek naar een gepast PNG vuilbak-icoontje;
- deze PNG toevoegen aan DOM bij elk citaat;
- CSS: absolute positioning binnen een citaat-article;
- de titel van een citaat is de ‘primary key’;
- functie `verwijderCitaat`;
- `splice` arraymethode.

)

4 `findIndex()` arraymethode

De arraymethode `findIndex` is een elegante manier om de index van een element te vinden. Het argument van `findIndex` is een callbackfunctie die `true` of `false` teruggeeft. De functie wordt opgeroepen voor elk element van de array. De eerste keer dat de functie `true` teruggeeft, stopt `findIndex` en geeft het de index van het element terug. Als de functie nooit `true` teruggeeft, geeft `findIndex` -1 terug.

We leggen de methode kort uit aan de hand van het kleurenvoorbeeld.

```
const kleuren = ['rood', 'geel', 'groen', 'blauw', 'wit', 'bruin']
```

We willen graag de eerste kleur vinden die meer dan vier letters heeft. Hiervoor definiëren we eerst een functie die als antwoord altijd true of false zal geven:

```
function meerDanVierLetters(kleur) {  
  return kleur.length > 4  
}
```

Deze functie wordt nu als callbackfunctie als argument meegegeven met `findIndex`:

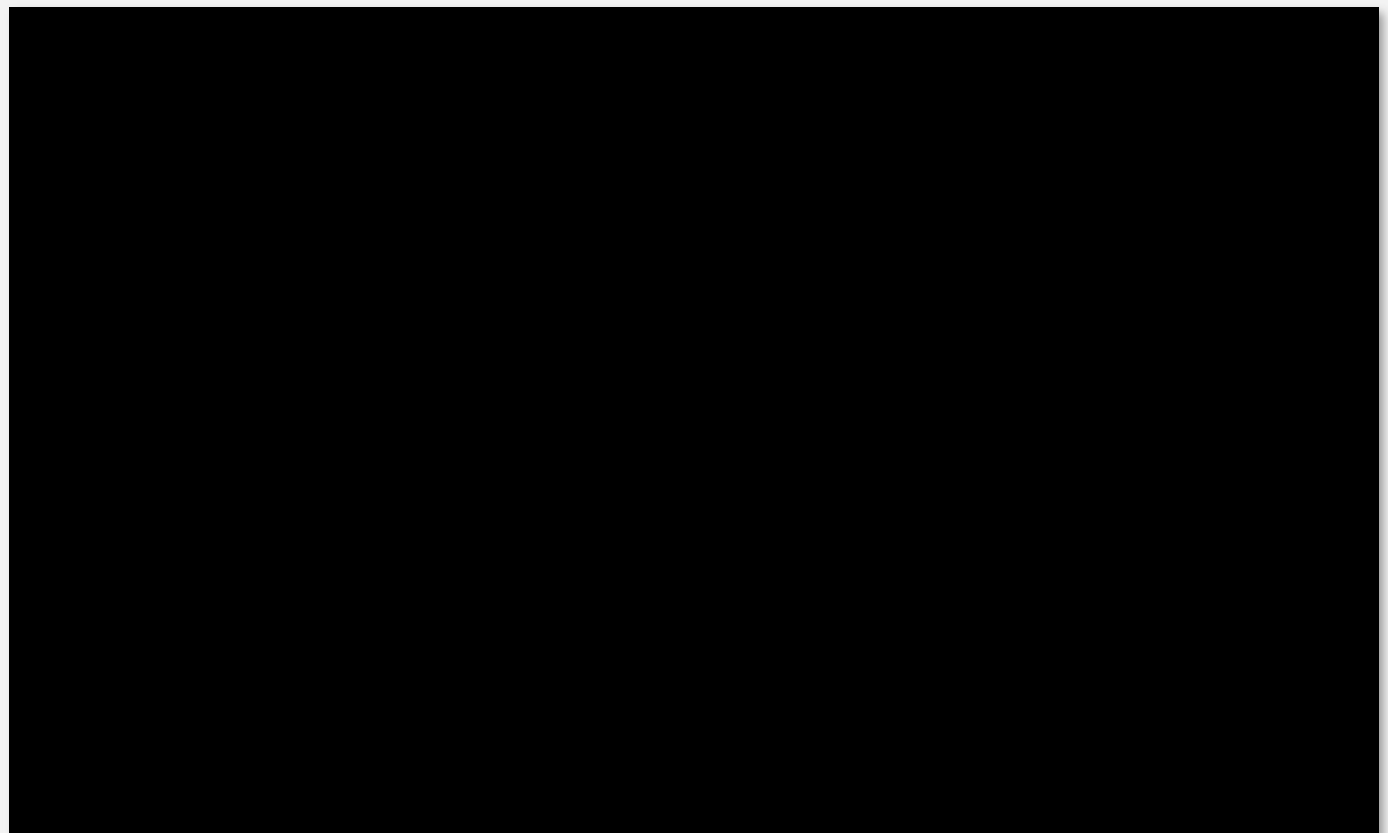
```
const index = kleuren.findIndex(meerDanVierLetters)  
console.log(index) // geeft 2 terug want 'groen' heeft meer dan 4 letters
```

```
// ipv van voor naar achter kan je ook achteraan beginnen zoeken:  
const laatsteIndex = kleuren.findLastIndex(meerDanVierLetters)  
console.log(laatsteIndex) // geeft 5 want 'bruin' heeft meer dan 4 letters
```

CallBackfuncties worden meestal met vette pijl notatie geschreven. Dan kan er dan zo uitzien:

```
const meerDanVierLetters = (kleur) => {  
  return kleur.length > 4  
}  
  
// Omdat er maar één regel is wordt deze functie meestal korter geschreven  
// zonder accolades. Het return statement wordt in dat geval weggelaten:  
const meerDanVierLetters = (kleur) => kleur.length > 4
```

5 Element verwijderen met findIndex



6 Uitbreiding: PNG vervangen door SVG

In het laatste filmpje gaan we net wat verder een ontdekken we een extra mogelijkheid van het SVG vectorformaat. Omdat dit beeldformaat tekstgebaseerd is, kan je er ook dingen in aanpassen via JS en CSS. Dat laat ons in het laatste filmpje toe om de vuilbakkleur aan te passen aan de taalkleur van het citaat.



© 2024 — Jan Van Hee

[↑ Naar top](#)