

“ Users do not care what technologies you use. They only care that a website is fast and accessible. That's the core of what responsible usage of JavaScript is all about: prioritizing user experience over developer experience. —Kyle Simpson, You Don't Know JS: Async & Performance

Arraymethoden

- 1 Lengte van een array
- 2 Langste citaat
- 3 Refactor met forEach
- 4 Citaat toevoegen met push of unshift
- 5 Refactor nieuwe functies
- 6 RWD

1 Lengte van een array

Met de `length` eigenschap kan je het aantal elementen van een array opvragen (correcer: het volgnummer van het laatste element + 1). Bovendien kan je deze eigenschap zelf een waarde geven. We tonen even hoe dit werkt:

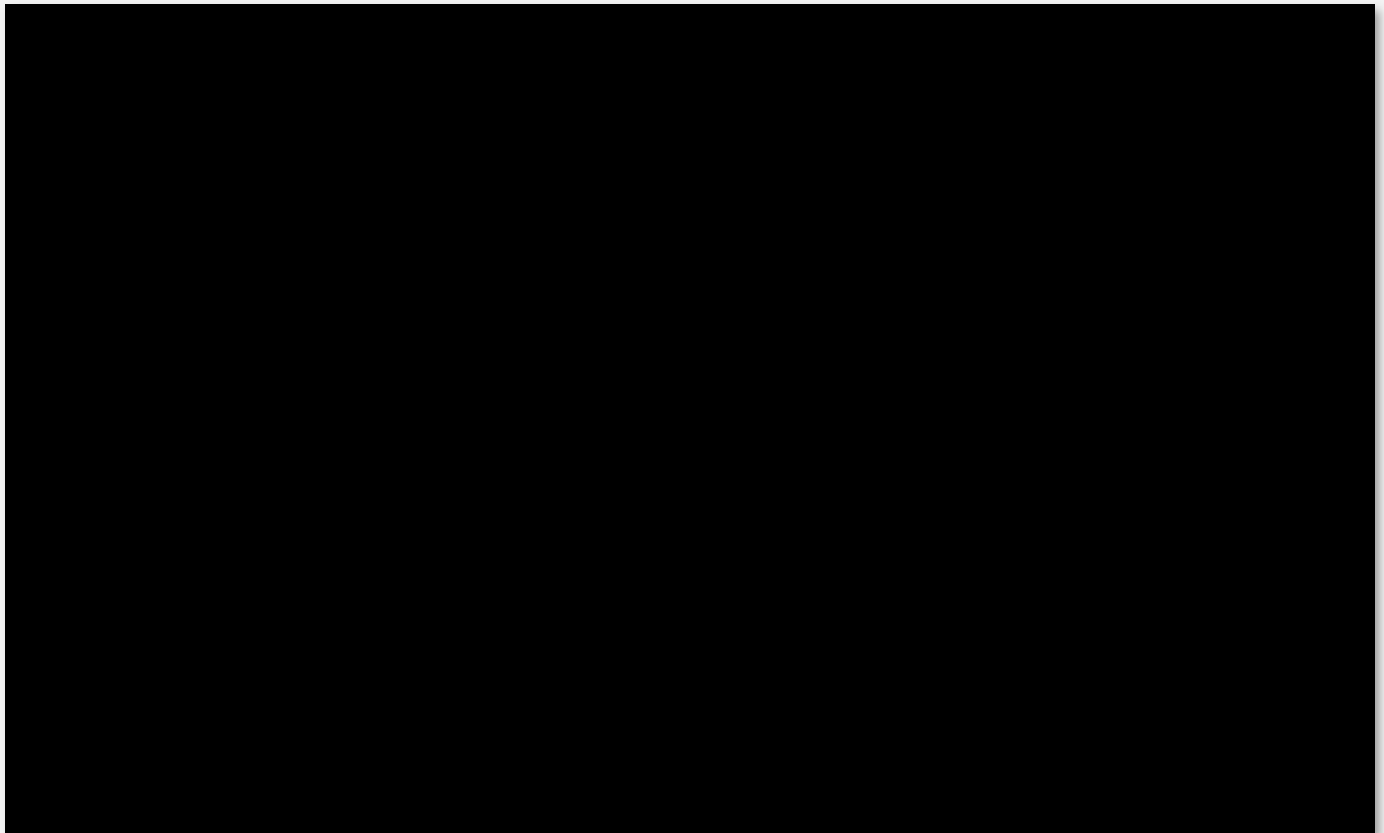
```
// definieer nieuwe array
const arr = [1, 'test', true, 3.5]
console.log(arr.length) // geeft als resultaat 4

// voeg een element op een bepaalde plaats toe
arr[6] = 'laatste'
console.log(arr) // geeft: Array(7) [ 1, "test", true, 3.5, <2 empty slots>, "laatste" ]

// verander de lengte van de array
arr.length = 0
console.log(arr) // geeft lege array: Array []
```



Het eerste filmpje toont hoe we de array-eigenschap `length` gebruiken om het aantal basiscitaten te tonen. We passen ook de CSS lichtjes aan.



2 Langste citaat



Het volgende filmpje toont:

- het verschil tussen `let` en `const`;
- de functie `vindLangsteCitaat` met een returnwaarde;
- een `for` lus om het grootste element in een array te vinden;
- de functieoproep in een nieuwe functie `toonLangsteCitaat`.

3 Refactor met forEach



Meer info over deze krachtige arrayfunctie vind je op

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

De arrayfunctie `forEach` is een iteratieve functie. Ze krijgt als parameter een callbackFunctie mee die wordt uitgevoerd op elk element van de array. Deze callbackFunctie heeft meestal één verplicht argument (nl. het huidige element waarop ze wordt uitgevoerd, maar kan er ook drie hebben, nl. (optioneel) de index van het huidige element en de array waarop ze wordt uitgevoerd.)

We werken een eenvoudig voorbeeld uit: overloop alle elementen van een array en toon die in de console. Het vertrekpunt is deze array:

```
const kleuren = ['rood', 'groen', 'blauw']
```

We maken eerst een callbackFunctie:

```
function toonkleur(el) {  
  console.log(el)  
}
```

Vervolgens wordt de `forEach` methode uitgevoerd op de array. Deze methode gaat de callbackFunctie op elk (niet leeg) element toepassen:

```
kleuren.forEach(toonkleur)  
  
// het resultaat in de console is:  
// "rood"  
// "groen"  
// "blauw"
```

Als de callbackFunctie ook de plaats van dit element moet weergeven, geven we ze een tweede parameter (waarvan je de naam zelf mag kiezen), bvb `index`:

```
function toonkleur2(el,index) {  
  console.log(`${el} staat op plaats ${index}`)  
}
```

We kunnen dan deze nieuwe callbackFunctie met de twee parameters oproepen met de `forEach`:

```
kleuren.forEach(toonkleur2)  
  
// het resultaat in de console is:  
// "rood staat op plaats 0"  
// "groen staat op plaats 1"  
// "blauw staat op plaats 2"
```

De callbackFunctie zal meestal een *anonieme* functie zijn. We noemen deze functie 'anoniem' omdat de functie die in de haakjes van de `forEach` beschreven wordt, effectie geen naam heeft (bvb. omdat we die verder nergens nog nodig hebben). We geven eerst een versie met de klassieke functienotatie:

```
kleuren.forEach(function(el,index) {  
  console.log(`${el} staat op plaats ${index}`)  
}))
```

Het is trouwens zo dat de meerderheid van JS-developers voor een anonieme callbackFunctie eerder zal kiezen voor de *vette pijl notatie*. We zullen dat dus meestal ook doen en raden je dat ook aan. Een alternatieve versie van bovenstaande code zou er dan zo kunnen uitzien:

```
kleuren.forEach((el,index) => {  
  console.log(`${el} staat op plaats ${index}`)  
})
```



In onderstaand filmpje herschrijven we de functie vindLangesteCitaat met de arraymethode `forEach`.

)

4 Citaat toevoegen met push of unshift

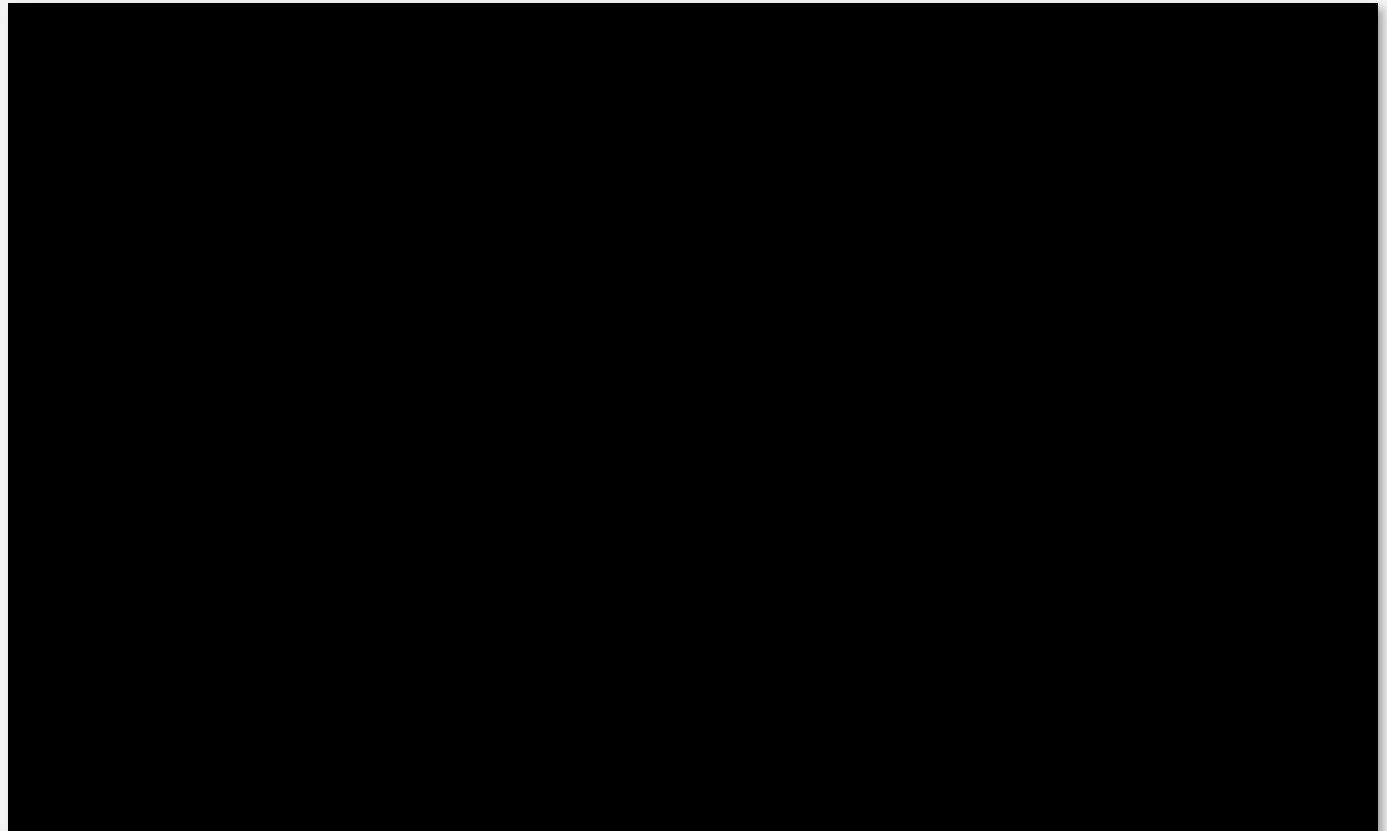
Je kan een element aan een array toevoegen met `push` (achteraan) of `unshift` (vooraan):

```
kleuren.push('oranje')  
// kleuren is nu: Array(4) [ "rood", "groen", "blauw", "oranje" ]
```

```
kleuren.unshift('grijs')  
// kleuren is nu: Array(5) [ "grijs", "rood", "groen", "blauw", "oranje" ]
```



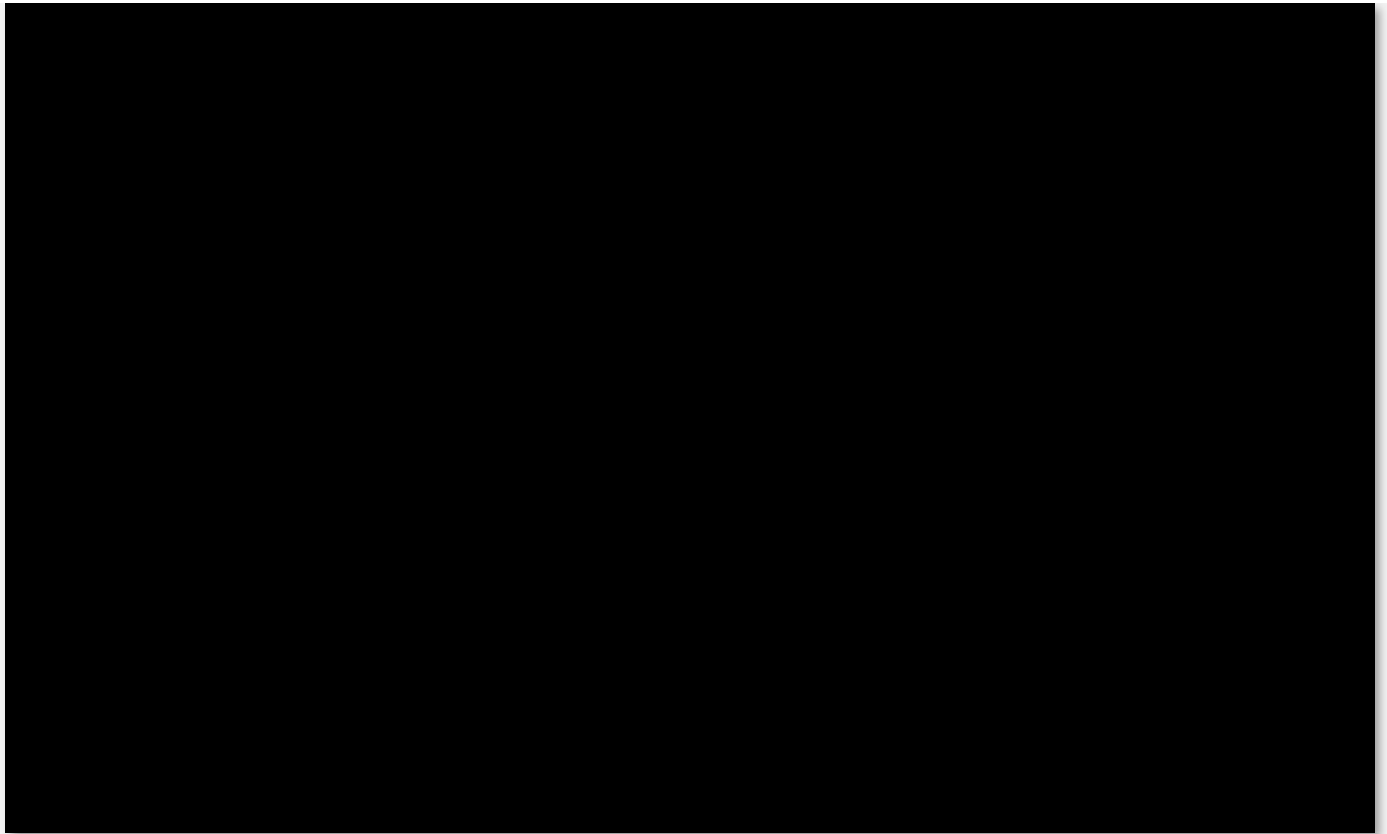
Het filmpje toont hoe je een citaat toevoegt aan de citatenarray.



5 Refactor nieuwe functies



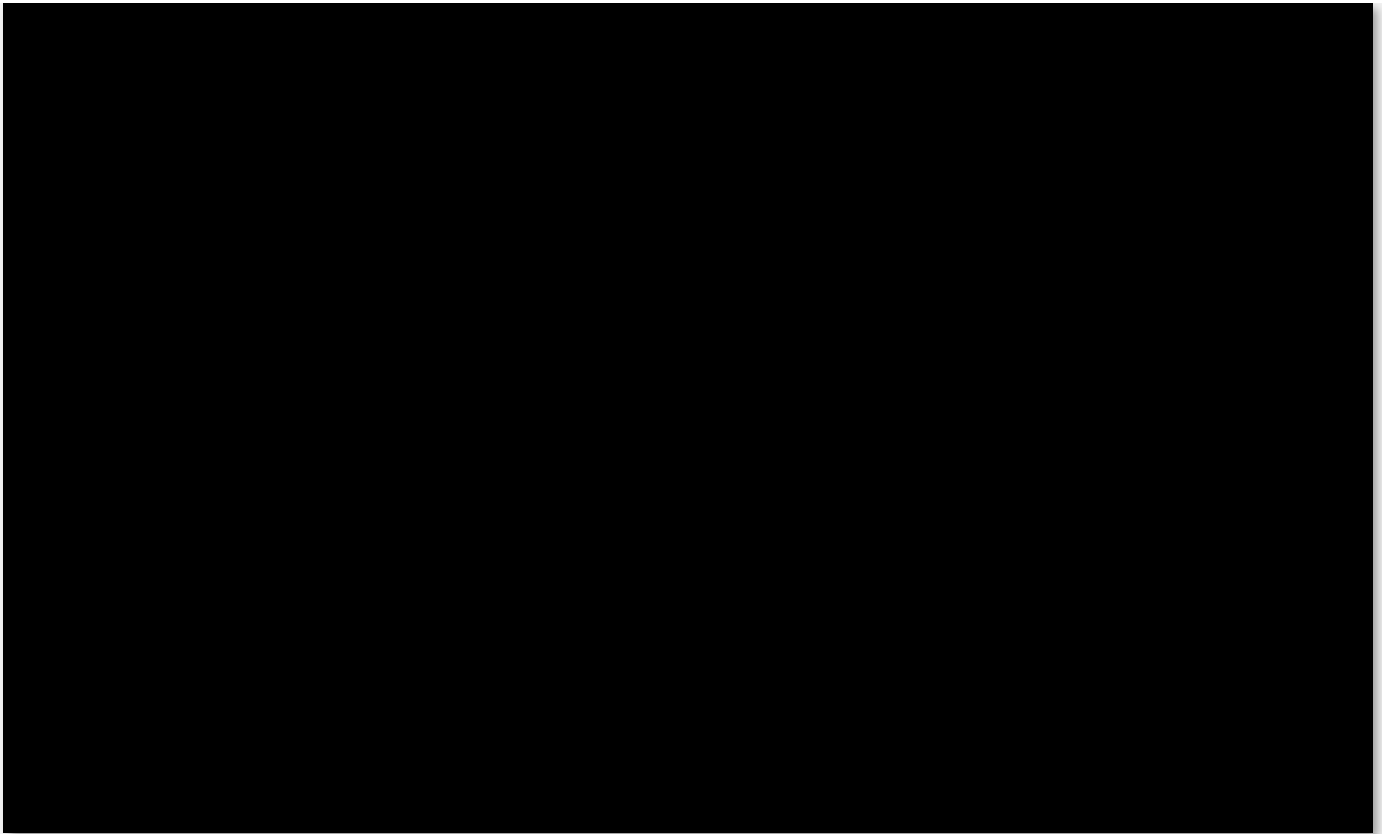
Stilaan tijd om de code wat op te kuisen en de flow van de applicatie wat logischer te maken.



6 RWD



In onderstaand filmpje schrijven we een beetje CSS-code om een basisresponsiviteit voor onze app te voorzien.



[↑ Naar top](#)

© 2024 — Jan Van Hee