

“ *Every programmer is an author.*  
—Sercan Leylek

## Objecten & arrays

1 Object

2 Array

Een object is een verzameling van eigenschappen, die elk een naam en een waarde hebben. Een array is een geordende lijst van waarden.

### 1 Object



Meer info over objecten vind je op <https://javascript.info/object> en volgende pagina's.

JavaScript heeft 8 datatypes. Er zijn zeven primitieve types (Number, BigInt, String, Boolean, null, undefined, symbol). We noemen ze ‘primitief’ omdat een variabele van dit type maar één ding kan bevatten.

In een object kan je een verzameling van verschillende soorten complexe data stoppen. Het is zonder twijfel het meest gebruikte datatype in JS. Zonder het te beseffen gebruik je al sinds het eerste hoofdstuk objecten.

Zo is het startpunt voor het DOM altijd ‘document’. Probeer volgende code in de console:

```
typeof(document) // resultaat is "object"  
typeof(document.querySelector('body')) // ook "object"
```

Je maakt een object m.b.v. accolades (`{...}`) die een optionele lijst van eigenschappen bevatten. Een eigenschap is een ‘sleutel:waarde’ paar (‘key:value’)

met als sleutel een string. De waarde kan alles zijn. Even een klein voorbeeld dat je zelf in een console kan intypen:

```
let gebruiker = {  
  naam: 'Jan',  
  leeftijd: 56,  
  isLector: true  
};  
typeof(gebruiker) // geeft "object"
```

Het ‘gebruiker’ object heeft drie eigenschappen. Elk van deze eigenschappen heeft een sleutel (i.p.v. het woord ‘sleutel’ wordt ook vaak ‘naam’ gebruikt) en een waarde. De eerste eigenschap heeft als sleutel de string ‘naam’ en als waarde de string ‘Jan’.

Je kan de waarde van een eigenschap opvragen met de punt-notatie:

```
gebruiker.naam // geeft "Jan"  
gebruiker.isLector // geeft de boolean true
```

Je kan ook de waarde van een eigenschap opvragen met de haakjes-notatie:

```
gebruiker['naam'] // geeft "Jan"
```

De haakjes-notatie is handig als je de naam van de eigenschap niet kent maar in een variabele hebt staan. Bijvoorbeeld:

```
let eigenschap = 'naam'; gebruiker[eigenschap] // geeft "Jan"
```

Je kan een eigenschap toevoegen aan een object door een nieuwe sleutel:waarde paar toe te voegen:

```
gebruiker.lengte = 187
```

Als je nu de inhoud van de variabele ‘gebruiker’ opvraagt, krijg je als antwoord:

```
Object { naam: "Jan", leeftijd: 56, isLector: true, lengte: 187 }
```

Je kan een eigenschap verwijderen door het ‘delete’ keyword te gebruiken:

```
delete gebruiker.isLector
```

Je bekomt als resultaat voor de variabele ‘gebruiker’:

```
Object { naam: "Jan", leeftijd: 56, lengte: 187 }
```

Je kan een object in een andere object stoppen. Dit noemen we een ‘geneste’ objecten. Bijvoorbeeld:

```
let gebruiker = { naam: 'Jan', adres: { straat: 'Kerkstraat', nummer: 12 } }
```

Als je nu de inhoud van de variabele ‘gebruiker’ opvraagt, krijg je als antwoord:

```
Object { naam: "Jan", adres: Object { straat: "Kerkstraat", nummer: 12 } }
```



We voeren nu objecten in voor onze ciatenapp. De video toont:

- hoe je de objectnotatie kan tonen in de console;
- het aanmaken van een nieuw citaat-object;
- het aanpassen van de functie `voegCitaatObjectToe`.

Invalid KS "EXPIRED", Error "-1,INVALID\_STR"



Nu we beschikken over een functie `voegCitaatObjectToe`, herschrijven we een stukje in onze app om hiervan gebruik te maken. We bekommen zo vier leesbaarder citaten en vier functieoproepen om de HTML article elementen te maken voor deze vier citaten.

Invalid KS "EXPIRED", Error "-1,INVALID\_STR"

## 2 Array



Meer info over arrays vind je op <https://javascript.info/array>.

Een array is een *geordende* lijst van waarden. Je kunt een array maken door een lijst van waarden tussen vierkante haken te zetten, gescheiden door komma's.

Erg efficiënt was het eindpunt van het vorige filmpje niet echt: we hadden voor elk citaat een aparte variable en moesten vier keer de functie `voegCitaatObjectToe` toepassen. Dat kan dus meer DRY!

Met een array kunnen we een lijst van citaten maken. We kunnen dan in een lus over deze array itereren en voor elk citaat een HTML article element maken.

We ontdekken later nog wel een aantal arraymethodes. Voor nu volstaat het dat we een array kunnen maken en via een lus alle elementen van de array kunnen bereiken.



Het eerste van de twee volgende filmpjes introduceert een array voor alle citaatobjecten. We bekijken volgende topics:

- een array aanmaken met [];
- alle elementen van de array overlopen via een klassieke `for`-lus;
- herschrijven naar een kortere `for ... of ...` lus.

Invalid KS "EXPIRED", Error "-1,INVALID\_STR"



Meer info over arraymethoden: zie <https://javascript.info/array-methods>.

Arrays worden zo vaak gebruikt dat het niet hoeft te verbazen dat opeenvolgende JS versies steeds krachtiger arraymethodes voor developers beschikbaar gemaakt hebben. Bij wijze van voorbeeld refactoren we de net geschreven lus via een kortere `forEach` arraymethode.

## Error

---

**No source video was found**

