Arrays

Een array is een variabele waarin een groot aantal items kan worden opgeborgen en elk item is bereikbaar via zijn nummer, de zogeheten index. De nummering van de items begint altijd bij 0.

De eenvoudigste manier om een array te maken, is met behulp van een array-literal, het letterlijk uitschrijven. Veel notaties die betrekking hebben op arrays maken gebruik van vierkante haken:

```
var hoogtes = [ 156, 192, 178 ];
```

In deze variabele, die `hoogtes` heet, zijn drie waarden opgeborgen. Die waarden (`156`, `192` en `178`) kunnen worden opgevraagd met respectievelijk `hoogtes[0]`, `hoogtes[1]` en `hoogtes[2]`.

Een array is een object en krijgt automatisch een aantal properties. Een van de properties is `length` die het aantal elementen in de array geeft. Dus `hoogtes.length` heeft de waarde 3.

Een andere manier om een array te maken, is door te beginnen met een lege array:

```
var a = [];     // lege array
```

Vervolgens kunt u de array vullen, bijvoorbeeld zo:

```
<p id="res"></p>
<script>
var a = [];
a[0] = 0;
a[1] = 10;
a[2] = 20;
document.getElementById("res").innerHTML = a.toString();
document.getElementById("res").innerHTML += " lengte= ";
document.getElementById("res").innerHTML += a.length;
</script>
```

De uitvoer van dit fragment is: `0,10,20 lengte= 3`.

Wanneer u bij het vullen van een array een of meer indexen overslaat, wordt op de ontbrekende plaatsen de waarde `undefined` ingevuld:

```
<p id="res"></p>
<script>
var a = [];
a[0] = 0;
a[1] = 10;
a[2] = 20;
a[10] = 100;
document.getElementById("res").innerHTML = a.toString();
document.getElementById("res").innerHTML += " lengte= ";
document.getElementById("res").innerHTML += a.length;
</script>
```

De uitvoer is nu: `0,10,20,,,,,,,,100 lengte= 11.`
Let op het rijtje komma's in de uitvoer dat de lege plaatsen aangeeft. De elementen `a[3]` tot en met `a[9]` hebben de waarde `undefined`.

Met arrays is nog veel meer mogelijk en er zijn methoden die belangrijk en nuttig kunnen zijn in het gebruik van arrays:

- De methodes `push()` en `pop()` om een element aan het einde van de array toe te voegen of te verwijderen.
- De methoden `shift()` en `unshift()` om aan het begin te verwijderen of toe te voegen.
- De methode `splice()` om elementen tussen te voegen, en eventueel te verwijderen.
  Let op: er is ook een methode `slice()` (dus niet s**p**lice) om een stukje uit een array te verwijderen.
- De methode `reverse()` om de volgorde van de elementen om te keren.
- De methode `concat()` om arrays samen te voegen.
- Een belangrijke methode is `sort()`, een methode die de elementen in een array (alfabetisch) sorteert.

Zie w3schools.com – arrays en w3schools.com – array methods.

Een array met getallen sorteren

De methode `sort()` sorteert de elementen van een array, maar zet ze daarbij eerst om naar een string (indien nodig). Dat betekent dat getallen niet goed gesorteerd worden. Een voorbeeld:

```
var getallen = [30, 100, 2, 22, 1, 3, 33, 10, 200];
getallen.sort();
alert(getallen);
```

De uitvoer is: `1, 10, 100, 2, 200, 22, 3, 30, 33.`
De getallen die met een 2 beginnen, staan vooraan. Daarna komen de getallen die met een 2 beginnen et cetera. Net zoals in een woordenboek alle woorden die met a beginnen, vooraan staan.

Om getallen te sorteren, moet `sort()` weten hoe hij getallen onderling moet vergelijken. Dat gebeurt door `sort()` aan te roepen met een functie die aangeeft hoe er gesorteerd moet worden. De functie heeft twee parameters: `a` en `b`. De functie moet zo zijn geconstrueerd dat hij op grond van de parameters `a` en `b` een negatieve waarde aflevert, de waarde 0 aflevert of een positieve waarde aflevert:

- Als a < b moet de functie een negatieve waarde afleveren.
- Als a ==b moet de functie de waarde 0 afleveren.
- Als a > b moet de functie een positieve waarde afleveren.

Een functie die voor de array getallen aan deze criteria voldoet, is:

```
function vergelijk( a, b ) {
  return a - b;
}
```

Toegepast levert dat:

```
var getallen = [30, 100, 2, 22, 1, 3, 33, 10, 200];

function vergelijk( a, b ) {
  return a - b;
}

getallen.sort( vergelijk );
alert(getallen);
```

De uitvoer is nu: `1`, `2`, `10`, `22`, `30`, `33`, `100`, `200`.
De methode `sort()` maakt hier gebruik van de functie `vergelijk()` om te bepalen welk getal voor het andere in de sortering komt.

Het kan nog iets korter door een *anonieme functie* te gebruiken:

```
var getallen = [30, 100, 2, 22, 1, 3, 33, 10, 200];
getallen.sort( function( a, b ) {
  return a - b;
});
alert(getallen);
```

Het voordeel van een anonieme functie is dat het erg kort is. Het nadeel is dat u de functie maar één keer kunt gebruiken; er is immers geen naam om aan de functie te refereren zoals in het eerste voorbeeld.

Een belangrijk statement dat vaak in samenhang met een array wordt gebruikt, is het *for-statement*. Hierover leert u later meer.