

# User Interfaces 1

## JavaScript

Arrays

# Arrays

---

- Wat zijn Arrays?
- Arrays aanmaken
- Arrays lezen en schrijven
- Lengte van een array
- Itereren over een array
- Array methodes
- Strings als arrays

# Wat zijn arrays?

---

```
let array = ["appel", "peer", 100, "banaan"];  
console.log(array[1]);  
console.log(array.length);
```

- Geïndexeerde lijst van elementen
  - Index start met 0
  - Elementen moeten niet van hetzelfde type te zijn
- Geen vaste grootte: kunnen groeien...
  - Maximaal  $2^{32}-2$  elementen
  - Lengte: via `length` eigenschap

# Arrays maken

---

```
let tabel = ["appel", "peer"];  
let tabelX = new Array("appel", "peer");  
let tabelY = Array("appel", "peer");
```

- Via
    - literal notatie: [ , , ]
    - Array-constructor met of zonder new
- Good-practice: gebruik de literal notatie!

- Lege array:

```
let array1 = [];  
let array2 = new Array();  
let array3 = Array();
```

```
let array = new Array(40);  
=> geeft array met 40 'undefined' elementen
```

# Lezen en schrijven van Arrays

---

```
let tabel = ["appel", "peer"];  
console.log(tabel[1]);  
tabel[0] = "banaan";
```

- Via [] notatie
- Arrays kunnen automatisch groeien:

```
tabel[2] = "citroen";
```

 → dit mag!

- Ook te hoog element uitlezen mag:

```
console.log(tabel[25]);
```

 → geeft 'undefined'

Je hebt dus nooit een "out of bounds"!

# Lengte van een Array

---

```
let tabel = ["appel", "peer"];  
tabel.length = 1;  
console.log(tabel[1]);
```

- `length` eigenschap
  - Geeft een numerieke waarde terug dat het aantal elementen in de array weergeeft
  - Is aanpasbaar:
    - lengte verkleinen: verwijdert effectief de elementen achteraan!
    - lengte vergroten: voegt 'undefined' elementen toe!

# Itereren over een Array

```
for (let item of tabel) {  
    console.log(item);  
}
```

sinds ES6: **for..of**  
opgelet: dit is achterliggend een gewone  
for-lus met een teller

```
for (let i=0; i<tabel.length; i++) {  
    //...  
}
```

Maak gebruik van een **for..of** lus  
tenzij je een teller nodig hebt (gewone for lus)  
Alternatief: **forEach()** function

```
tabel.forEach(function(element) {  
    console.log(element);  
})
```

→ Functie in functie (zie  
later)



Géén **for..in** lus gebruiken! Die loopt ook over eventuele andere  
eigenschappen van het array-object... (zie oefening arrays 2)

# Array methodes

---

```
let tabel = ["appel", "peer"];  
let array = ["banaan", "citroen"];  
console.log(tabel.reverse());  
console.log(tabel.concat(array).sort());
```

## Veel methodes:

push() → achteraan toevoegen  
pop() → laatste element verwijderen  
shift() → eerste element verwijderen  
join()  
reverse()  
sort()  
concat()  
slice()  
...

wijzigt grootte !!!  
→ geen for-lus mogelijk  
vanwege de teller

<https://developer.mozilla.org/en/JavaScript/Reference>



# Strings als arrays beschouwen

```
let woord = "papegaaieeieren";
for (let i = 0; i < woord.length; i++) {
  console.log(woord[i]);
}
```

for-of is veel eenvoudiger:

```
let woord = "papegaaieeieren";
for (let letter of woord) {
  console.log(letter);
}
```

- Je kan karakters van strings via de [ ] notatie aanspreken, in plaats van via de charAt methode...
- Je kan arrays naar strings omzetten via de toString of join methoden.

Opgelet:

`woord[0]="P"` kan niet in JS !  
strings zijn immers immutable...  
(geeft geen fout tenzij "use strict",  
moet vooraan, vóór eerste statement)

```
let numbers = ["One", "Two", "Three"];
console.log(numbers.toString());
console.log(numbers.join(' + '));
```

Te kiezen separator string

# Oefeningen arrays 1



## Deel 1:

Maak een tabel met de strings "een", "twee", "drie", "vier" en "vijf" en druk daarna de inhoud ervan met behulp van `console.log` onder elkaar af (gebruik een `for` lus)

Druk ze ook eens alle 5 op één regel af.

## Deel 2:

Maak via een `Array()` declaratie een tabel met de inhoud:  
"Meir 16", 2000, 'Antwerpen'

Gebruik de gegevens uit de tabel om via `console.log` de volgende uitvoer te bekomen:

```
Meir 16  
2000 Antwerpen
```

# Oefeningen arrays 2



Gegeven de volgende declaraties:

```
let steden = ["Brussel", "Antwerpen", "Gent"];  
let postcodes = new Array(3);  
postcodes["Brussel"] = 1000;  
postcodes["Antwerpen"] = 2000;  
postcodes["Gent"] = 9000;
```

← associatieve array!

Zorg ervoor dat je de volgende uitvoer bekomt (verplicht gebruik van een for-lus):

```
Postcodes  
Brussel - 1000  
Antwerpen - 2000  
Gent - 9000
```

doe dit ook eens met een for-in lus...

# Oefeningen arrays 3



(Queue) Vertrek vanuit de volgende JavaScript code

```
let tabel = ["appel", "peer", "banaan"];  
let extra = ["citroen", "sinaasappel"];
```

1. Voeg de elementen van de array **extra** achteraan aan de array **tabel** toe. Maak gebruik van de **push** en **shift** functies van Array.

```
let tabel = ["appel", "peer", "banaan", "citroen", "sinaasappel"];  
let extra = [];
```

2. Haal de elementen een voor een (**begin met het eerste**) uit de nieuw gevormde array **tabel** en plaats ze achter elkaar in de string **result** (laat een spatie na elk element). Maak gebruik van de **push** en **shift** functies van Array.

Gewenste afdruk:

shift wijzigt length!

```
appel peer banaan citroen sinaasappel
```

Controleer ook de inhoud van de arrays 'tabel' en 'extra'...  
...ze moeten nu 'leeg' zijn...

# Oefeningen arrays 4



(Stack) Vertrek vanuit de volgende JavaScript code

```
let tabel = ["appel", "peer", "banaan"];  
let extra = ["citroen", "sinaasappel"];
```

Voeg de elementen van de array **extra** in volgorde achteraan aan de array **tabel** toe. Haal de elementen een voor een (**begin met het laatste**) uit de nieuw gevormde array **tabel** en plaats ze achter elkaar in de string **result** (laat een spatie na elk element). Maak gebruik van de **pop** functie.

Gewenste afdruk:

pop wijzigt length!

```
sinaasappel citroen banaan peer appel
```

Controleer ook de inhoud van de arrays 'tabel' en 'extra'...  
...ze moeten weer 'leeg' zijn...

# Oefeningen arrays 5



Tik het volgende script in en voer het uit in de console:

```
var getallen = [ 12, 19, 23, 123, 100, 121, 45, 44 ];  
getallen.sort();  
console.log(getallen);
```

Wat merk je op in de uitvoer?

Zoek (google) een oplossing voor het probleem, de uitvoer moet als volgt worden:

```
[12, 19, 23, 44, 45, 100, 121, 123]
```

# Oefeningen strings 1

---



Start in de console met de volgende declaratie:

```
let zin = prompt("Geef een zin in:");
```

Maak gebruik van de nodige String-functies om het volgende te bekomen:

Invoer: **Dit is een korte zin**

Uitvoer: **zin korte een is Dit**

Probeer ook deze uitvoervariante eens:

```
niz etrok nee si tiD
```

# Oefeningen strings 2



Start in de console met de volgende declaratie:

```
let zin = "De aardappeleters van Van Gogh";
```

Maak gebruik van de nodige String-functies om via console.log de volgende uitvoer te bekomen:

DGVaaaaadeeeeeghInnopprstv

4 spaties



# Oefeningen strings 3

---



Start in de console met de volgende declaratie:

```
let zin = "the quick brown fox jumps over the lazy dog";
```

Tracht nu via console.log  
de volgende uitvoer te bekomen:

```
The  
Quick  
Brown  
Fox  
Jumps  
Over  
The  
Lazy  
Dog
```