

## ALGORITMISCH DENKEN

---

# ARRAYS

# Wat Zijn Arrays?

## Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

## Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

## Looping

for-of

for-in

## JavaScript vs Java

# Arrays

## Booleans

true

false

## Getallen

0

1

2

-1

2.45

...

# Arrays

## Booleans

true

false

## Getallen

0

1

2

-1

2.45

...

## Arrays

[]

[1, 2, 3]

[false, 8, 1, 4, true]

# Arrays

Arrays zijn lijsten

## Voorbeeld

[1, 2, 3, 4, 5]

Deze array bevat vijf elementen, nl. de waarden van 1 t.e.m. 5.

# Arrays

Arrays kunnen een willekeurig aantal elementen bevatten.

## Voorbeeld

Leeg []

Eén element [1]

Twee elementen [1, 2]

Drie elementen [1, 2, 3]

# Arrays

Arrays kunnen alle types waarden bevatten.

## Voorbeeld

Booleans            `[true, false, false]`

Getallen            `[4, 1, 2, 9, 3]`

Gemengd            `[1, true, 9, 2, false]`

Andere arrays      `[ [1, 2, 3], [4, 5, 6] ]`

# Visualisatie

[8, 1, true]

stellen we visueel door als





## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

### JavaScript vs Java

# Opvragen Elementen

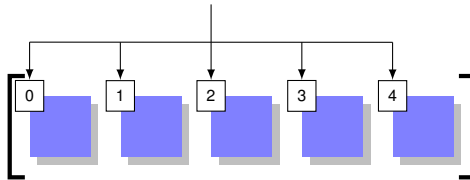
Gegeven

```
let xs = [7, 2, 9, false, 1]
```

Hoe geraken we aan de elementen in de array `xs`?

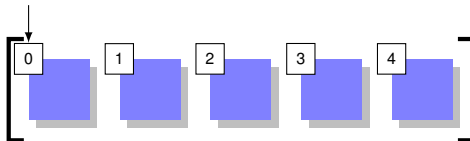
# Indices

Elk element heeft een eigen index



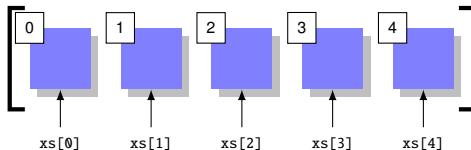
# Indices

Eerste element heeft index 0



# Indices

## Syntax



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

### JavaScript vs Java

# Lengte Array

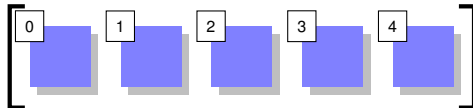
Aantal elementen in een array opvragen

```
xs.length
```

# Geldige Indices

Geldige indices gaan van 0 t.e.m. `xs.length-1`

## Voorbeeld



- Array heeft lengte 5
- Index van laatste element is  $5 - 1 = 4$



# Voorbeeld

```
let xs = [7, 2, 9, false, [3, 5]]
```

Expressie	Resultaat
<code>xs.length</code>	5
<code>xs[0]</code>	7
<code>xs[1]</code>	2
<code>xs[2]</code>	9
<code>xs[3]</code>	false
<code>xs[4]</code>	[3, 5]

## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

**Overschrijven Elementen**

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

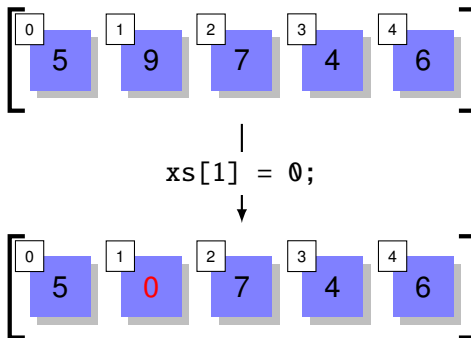
### JavaScript vs Java

# Overschrijven Elementen

Men kan elementen overschrijven met

`xs[index] = waarde;`

## Voorbeeld



## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking



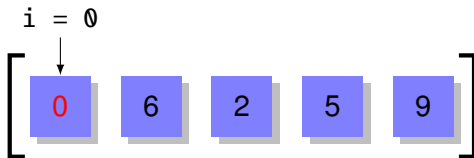
## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking



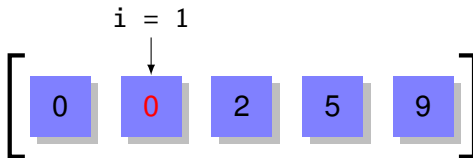
## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking



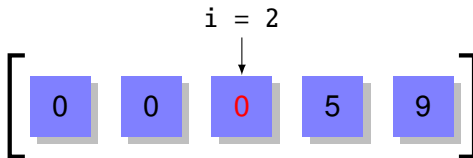
## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking



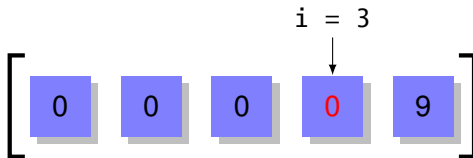
## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking





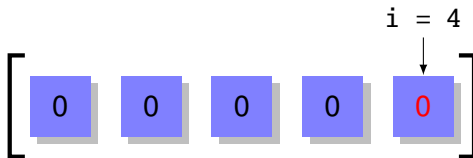
## Voorbeeld

---

```
function fillWithZeros(xs)
{
  for ( let i = 0; i !== xs.length; ++i )
  {
    xs[i] = 0;
  }
}
```

---

## Visualisatie Werking



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

**Aanmaken Array**

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

### JavaScript vs Java

# Aanmaken Array

Via array literals: [ elt1, elt2, ... ]

Voorbeeld

```
let xs = []
```

Lege array

# Aanmaken Array

Via array literals: [ elt1, elt2, ... ]

Voorbeeld

```
let xs = [4]
```



# Aanmaken Array

Via array literals: [ elt1, elt2, ... ]

## Voorbeeld

```
let xs = [5,4,3,2,1]
```

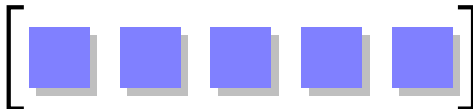


# Aanmaken Array

Via Array constructor: **new** Array(lengte)  
Array is dan gevuld met undefined

## Voorbeeld

```
let xs = new Array(5)
```



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

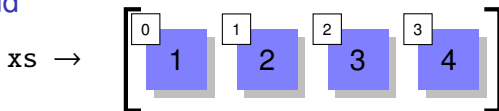
for-in

### JavaScript vs Java

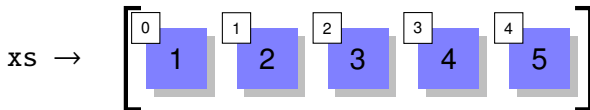
# Push

`xs.push(elt)` voegt `elt` toe aan einde van `xs`

## Voorbeeld



`xs.push(5);`

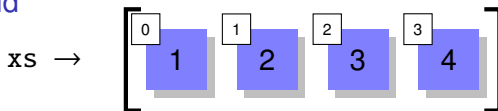




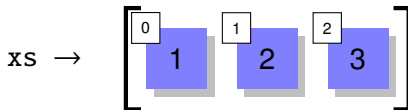
# Pop

`xs.pop()` verwijdt laatste element en retournt het

## Voorbeeld



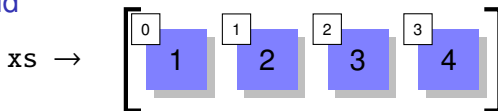
```
let last = xs.pop();
```



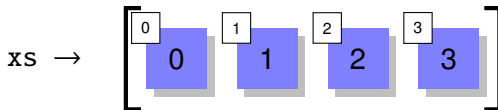
# Unshift

`xs.unshift(elt)` voegt `elt` toe aan begin van `xs`

## Voorbeeld



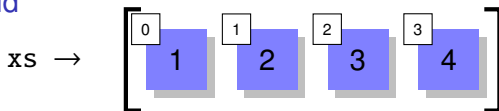
`xs.unshift(0);`



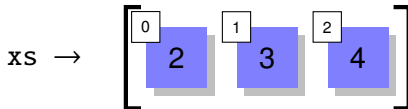
# Shift

`xs.shift()` verwijdt eerste element en returnt het

## Voorbeeld



```
let first = xs.shift();
```



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

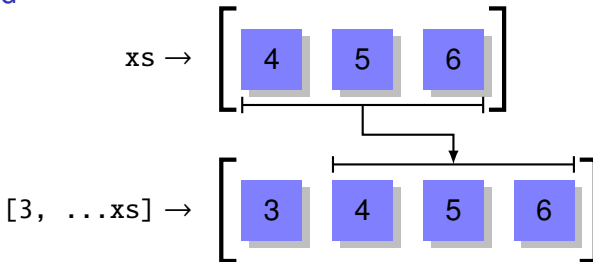
### JavaScript vs Java

# Spread Operator

Spread operator laat toe om snel nieuwe arrays op te bouwen

`...xs` betekent "plaats hier de elementen van `xs`"

## Voorbeeld

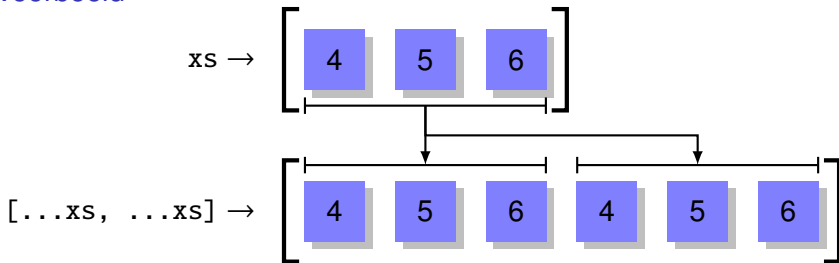


# Spread Operator

Spread operator laat toe om snel nieuwe arrays op te bouwen

`...xs` betekent "plaats hier de elementen van `xs`"

## Voorbeeld



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

### JavaScript vs Java

## Andere Methodes

Signatuur	Beschrijving
<code>xs.includes(x)</code>	Komt x voor in xs?
<code>xs.indexOf(x)</code>	Index van eerste voorkomen van x
<code>xs.lastIndexOf(x)</code>	Index van laatste voorkomen van x
<code>xs.reverse()</code>	Keert volgorde elementen om
<code>xs.slice(i, j)</code>	Deelarray van i tot j



## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

**for-of**

**for-in**

### JavaScript vs Java

# Vaak Voorkomend Patroon

Vaak moeten we alle elementen van een array afgaan

---

```
for ( let i = 0; i !== xs.length; ++i )  
{  
    const x = xs[i];  
  
    // ...  
}
```

---

## for-of

---

```
for ( const x of xs )  
{  
    // ...  
}
```

---

- De lusbody wordt voor elk element om beurt uitgevoerd
- `x` is gelijk aan het "huidig" element

# Voorbeeld

---

```
const xs = [8, 5, 7];
```

```
for ( const x of xs )  
{  
    foo(x);  
}
```

---

is equivalent met

---

```
foo(8);  
foo(5);  
foo(7);
```

---

## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

**for-of**

**for-in**

### JavaScript vs Java

## for-in

---

```
for ( const i in xs )  
{  
    // ...  
}
```

---

Gaat niet de *waarden* van een array af, maar de *indices*

## Voorbeeld

---

```
const xs = [4, 7, 2];
```

```
for ( const i in xs )  
{  
    foo(i);  
}
```

---

is equivalent met

---

```
foo(0);  
foo(1);  
foo(2);
```

---

## Wat Zijn Arrays?

### Basisoperaties

Opvragen Elementen

Opvragen Lengte

Overschrijven Elementen

Aanmaken Array

### Extra Operaties

Toevoegen & Verwijderen

Spread Operator

Andere Methodes

### Looping

for-of

for-in

## JavaScript vs Java



# JavaScript vs Java

- Terminologie is vaak programmeertaal-specifiek
- JavaScript arrays  $\neq$  Java arrays
- JavaScript arrays  $\approx$  Java ArrayLists

# JavaScript vs Java

## JavaScript Arrays

- Arrays kunnen groeien (push, unshift)
- Arrays kunnen krimpen (pop, shift)

## Java Arrays

- Arrays hebben een vaste lengte

## Java ArrayLists

- ArrayLists kunnen groeien en krimpen

# JavaScript vs Java

## JavaScript Arrays

- Heterogeen: kunnen elementen van verschillende types bevatten

## Java Arrays

- Homogeen: alle types moeten eenzelfde type hebben

## Java ArrayLists

- Homogeen: alle types moeten eenzelfde type hebben