



JavaScript

Verzweigungen, Schleifen und Arrays



```
if (0) {  
    console.log('A');  
} else if (false) {  
    console.log('B');  
} else if (null) {  
    console.log('C');  
} else if (undefined) {  
    console.log('D');  
} else if (NaN) {  
    console.log('E');  
} else if ('' || '') {  
    console.log('F');  
} else {  
    console.log('G');  
}
```

G

```
for (let i = 0; i < 100; i++) {  
    console.log(i);  
}  
// 0 1 2 3 ...
```

```
let l = 0;  
for (;;) {  
    if (l === 100) break;  
    console.log(l);  
    l++;  
}  
// 0 1 2 3 ...
```



- Mittels dieser Schleife kann man über alle aufzählbaren Eigenschaften eines Objektes iterieren.
 - Die Variable `i` stellt den Eigenschaftsnamen als String dar.
- Dies funktioniert auch mit Arrays, da Arrays Objekte und die Indexes Eigenschaftsnamen sind.
- Die Reihenfolge ist aber nicht garantiert.

```
const objekt = {  
  eigenschaft1: '...',  
  eigenschaft2: '...' ;  
};  
  
for (let i in objekt) {  
  console.log(i);  
}  
  
// eigenschaft1 eigenschaft2
```



- Die Schleife iteriert über alle Elemente eines iterierbaren Objektes.
 - Iterierbare Objekte sind Array, Typedarrays, Map, ...
- Es werden nur eigene Eigenschaften iteriert.
- `i` enthält den Wert der Eigenschaft und nicht den Namen.

```
const array1 = [1, 2, 3];
```

```
for (let i of array1) {  
    console.log(i);  
}
```

```
// 1 2 3
```



```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 100);
// 0 1 2 3 ...
```

```
i = 0;
while(i < 100) {
  console.log(i);
  i++;
}
// 0 1 2 3 ...
```



```
var a1 = new Array();  
a1[0] = "Hallo";  
a1[1] = 42;  
document.write(a1.length+"<br/>");  
a1[22] = 23;  
document.write(a1+"<br/>");  
document.write(a1.length+"<br/>");  
document.write(a1[5]+"<br/>");
```

2
Hallo,42,,,,,,,,,,,,,,,,,,,,,23
23
undefined



```
var a2 = new Array(3);  
a2[0] = "Hallo";  
a2[1] = 42;  
document.write(a2.length+"<br />");  
a2[12] = 23;  
document.write(a2+"<br />");  
document.write(a2.length+"<br />");
```

```
3  
Hallo,42,,,,,,,,,23  
13
```




```
// Array direkt initialisieren
var a3 = new Array("Hallo", 24, 23);
// Array Literale
var leer = [];
var zahlen = [ "eins", "zwei", "drei" ];

document.write(leer[0]+"<br/>");
document.write(zahlen[1]+"<br/>");
```

undefined
zwei



```
var zahlen = [ "eins", "zwei", "drei" ];  
delete zahlen[1];  
document.write(zahlen+"<br/>");  
document.write(zahlen.length+"<br/>");
```

eins,,drei

3



JavaScript Array splice() Method

Definition and Usage

The splice() method adds/removes items to/from an array, and returns the removed item(s).

Note: This method changes the original array.

https://www.w3schools.com/Jsref/jsref_splice.asp



JavaScript Array splice() Method

Syntax

```
array.splice(index, howmany, item1, ....., itemX)
```

Parameter Values

| Parameter | Description |
|--------------------------|---|
| <i>index</i> | Required. An integer that specifies at what position to add/remove items, Use negative values to specify the position from the end of the array |
| <i>howmany</i> | Optional. The number of items to be removed. If set to 0, no items will be removed |
| <i>item1, ..., itemX</i> | Optional. The new item(s) to be added to the array |

https://www.w3schools.com/Jsref/jsref_splice.asp



JavaScript Array splice() Method

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.splice(2, 0, "Lemon", "Kiwi");
```

Banana,Orange,Apple,Mango

Banana,Orange,Lemon,Kiwi,Apple,Mango

https://www.w3schools.com/Jsref/jsref_splice.asp



JavaScript Array splice() Method

```
var fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi"];  
fruits.splice(2, 2);
```

Banana,Orange,Apple,Mango,Kiwi

Banana,Orange,Kiwi

https://www.w3schools.com/Jsref/jsref_splice.asp



```
var a5 = [ "a", "b", "c" ];  
a5[a5.length] = "d";  
document.write(a5+"<br/>");  
a5.push("e");  
document.write(a5+"<br/>");
```

a,b,c,d

a,b,c,d,e



- besonderes Array mit String als Index
 - unterschiedliche Typen als Inhalt möglich
 - ähnlich einer Map in Java
- Verarbeitung
 - Erzeugung mit `new Array()`
 - Auslesen eines Elements über `["Index"]`
 - Iterieren über alle Elemente mit for-in Schleife
`for (var e in array) { ... }`



Assoziative Arrays: Beispiel



```
var a6 = new Array();
a6["go"] = "gehen";
a6["walk"] = "gehen";
a6["sleep"] = "schlafen";
document.write(a6["go"]+"<br/>");
for (var e in a6 ) {
    document.write(e + ": " + a6[e] + "<br/>");
}
```

gehen
go: gehen
walk: gehen
sleep: schlafen



Assoziative Arrays: Beispiel als Array-Literal



```
var a7 = {  
    "go" : "gehen",  
    "walk" : "gehen",  
    "sleep" : "schlafen"  
};  
  
for (var e in a7 ) {  
    document.write(e + " : " + a7[e] + "<br/>");  
}
```

go: gehen
walk: gehen
sleep: schlafen