

ARRAYS

Arrays sind Behälter / Container für mehrere Werte. Dabei können die gespeicherten Datentypen gemischt (mixed) werden, d.h. es können Numbers, Strings und Booleans in einem Array gespeichert werden. Die einzelnen Teile (Items) eines Arrays sind durch Komma getrennt.

Als Items können auch andere Arrays (nested Arrays) und Objekte gespeichert werden.

Arrays werden durch das Schlüsselwort **var** und durch eckige Klammern `[]` initialisiert.

```
var testArray1 = [];
```

// Initialisiert einen leeren Array namens testArray1

```
var testArray2 = [ `Köln`, 17, true, [1, `Berlin`] ];
```

// Initialisiert einen Array namens testArray2 mit vier Items, das vierte Item ist wieder ein Array.

Jedes Item des Arrays hat eine **Index**-Nummer, welche es ermöglicht einzelne Teile des Arrays anzusprechen.

Die Index-Nummer beginnt bei 0, d.h. das erste Item hat den Index 0, das zweite den Index 1 usw.

```
testArray2[2]
```

// hat den Wert true

```
testArray2[4][1]
```

// hat den Wert 1

Die Länge eines Arrays kann mit `.length` bestimmt werden, dabei wird die Anzahl der Items gezählt.

Das letzte Item eines Array wird mit `.length-1` angesprochen

```
var testArray3 = [ `Köln`, `Berlin`, `Hannover`, `Dresden` ];
```

```
var letztesItemVonTestarray3=testArray3[testArray3.length-1];
```

```
console.log(letztesItemVonTestarray3);
```

// gibt auf der Console `Dresden` aus

Beeinflussung von Arrays

Es gibt folgende Möglichkeiten Arrays zu beeinflussen, dabei wird der ursprüngliche Array überschrieben:

.push(neues_item)

// fügt dem Array ein neues Item als letztes Item zu

```
var testArray4 = [`Deutschland`, `Frankreich`, `Österreich`];  
testArray4.push(`Schweiz`);
```

// fügt dem Array testArray4 den String `Schweiz` als neues letztes Item zu

```
console.log(testArray4);
```

// gibt auf der Console ["Deutschland", "Frankreich", "Österreich", "Schweiz"] aus

.pop()

// löscht das letzte Item des Arrays

```
var testArray5 = [`Milch`, `Joghurt`, `Quark`];  
testArray5.pop();
```

// löscht bei dem Array testArray5 das letzte Item

```
console.log(testArray5);
```

// gibt auf der Console ["Milch", "Joghurt"] aus

.unshift(neues_item)

// analog zu dem Befehl .push(neues_item) fügt es einem Array ein Item als erstes Item zu und ändert dadurch für alle Items des Arrays den Index

.shift()

// analog zu dem Befehl .pop() entfernt es das erste Item eines Arrays und ändert dadurch für alle Items des Arrays den index

.splice()

// verwendet man, um einen Array Index-basiert zu verändern

```
var testArray6 = [1,2,3,4,5,6,7,8,9,10];  
testArray6.splice(3,2);
```

// schneidet ab Index 3 zwei Items heraus.

```
console.log(testArray6);
```

gibt auf der Console [1, 2, 3, 6, 7, 8, 9, 10] aus

```
var testArray7 = [1,2,3,4,5,6,7,8,9,10];  
testArray7.splice(3, 1, `a`);
```

// schneidet ab Index 3 ein Item heraus und setzt an dieser Stelle eine neues Item `a` ein

```
console.log(testArray7);
```

// gibt auf der Console [1, 2, 3, "a", 5, 6, 7, 8, 9, 10] aus

.split()

// verwendet man um aus einem String in einen Array zu verwandeln, in der Klammer wird in Anführungszeichen der Seperator genannt

```
var testString1 = `Donald Duck wohnt in Entenhausen`;
var testArray8 = testString1.split(" ");
// separiert aus den String testString1 alle Zeichenfolgen die durch ein Leerzeichen
getrennt sind und speichert die separierten Zeichenfolgen als Strings in dem Array
testArray8
console.log(testArray8);
// gibt auf der Console ["Donald", "Duck", "wohnt", "in", "Entenhausen"] aus
```

.split kann noch mit einem Limit ergänzt werden, dann wird nur die als Limit angegebene Anzahl

```
var testString2 = `Dagobert Duck wohnt ebenfalls n Entenhausen`;
var testArray9 = testString2.split(" ", 2);
// separiert aus den String testString2 alle Zeichenfolgen die durch ein Leerzeichen
getrennt sind und speichert die ersten beiden separierten Zeichenfolgen als Strings in dem
Array testArray9
console.log(testArray9);
// gibt auf der Console ["Dagobert", "Duck"] aus
```

.join()

// erzeugt einen String aus den einzelnen Items eines Arrays, als Trennzeichen im String wird das in Anführungszeichen genannte Zeichen in der Klammer genommen

```
var testArray10 = [`Gustav`, `Gans`, `ist`, `auch`, `ein`, `Bewohner`, `von`,
`Entenhausen`];
var testString3 = testArray10.join(" ");
// Macht aus den einzelnen Items des Arrays testArray10 einen String namens testString3
mit dem Leerzeichen als Trennzeichen
console.log(testString3)
// gibt auf der Console den String „Gustav Gans ist auch ein Bewohner von Entenhausen“
aus
```

.concat()

// vereint zwei Arrays zu einem einzelnen Array, die Items und ihre Reihenfolge werden nicht verändert

```
var testArray11 = [`Donald Duck`, `Daisy Duck`];
var testArray12 = [`Tick`, `Trick`, `Track`];
var testArray13 = testArray11.concat(testArray12);
// Kreiert einen Array namens testarray13 welches aus dem Arrays testArray1 und
testArray12 besteht
console.log(testArray13);
// gib auf der Console den Array ["Donald Duck", "Daisy Duck", "Tick", "Trick", "Track"] aus
```

delete

// löscht Elemente in einem Array ohne den Index zu beeinflussen, das heißt es wird nur der Wert des Items gelöscht.

```
var testArray14 = [`Bitte`, `nicht`, `löschen!`];
delete testArray14[1];
// löscht den Wert des Index 1 von dem Array testArray14
console.log(testArray14);
// gibt auf der Console folgenden Array aus: ["Bitte", empty, "löschen!"]
console.log(testArray14[1]);
// gibt auf der Console undefined aus
```

.slice

// kopiert den ganzen Array oder Teile davon

```
var testArray15 = ["mein", "Name", "ist", "John", "Doe", "und", "mein",
"Gehirn", "ist", "leer"];
var testArray16 = testArray15.slice();
// erstellt einen neuen Array namens testArray16, welcher eine Kopie von testArray15 ist
console.log(testArray16);
// gib auf der Console ["mein", "Name", "ist", "John", "Doe", "und", "mein", "Gehirn", "ist",
"leer"] aus
```

```
var testArray17 = testArray15.slice(3);
// erstellt eine Kopie von testArray15 ab Index 3
console.log(testArray17);
/// gib auf der Console ["John", "Doe", "und", "mein", "Gehirn", "ist", "leer"] aus
```

```
var testArray18 = testArray15.slice(3,5);
// erstellt eine Kopie von testArray15 ab Index 3, vor Index 5 endet die Kopie
console.log(testArray18);
/// gib auf der Console ["John", "Doe"] aus
```

Questions? You're welcome!

holger.zerbe@web.de