

Arrays lassen sich recht einfach sortieren. Dafür gibt es die `.sort()` Methode bzw. die `.reverse()` Methode um die Inhalte umgekehrt aufzulisten.

**JS** `arrayname.sort();`



Die Arrayelemente werden alphabetisch aufwärts sortiert. Bei dieser Methode werden die Elemente wie ein String behandelt.

```
var studium = ["Philosophie", "Chemie", "Medizin", "Jus"];
studium.sort();
window.alert(studium);
```

**JS** `arrayname.reverse();`



Dreht die Reihenfolge um, sodaß das letzte Element zum Ersten wird und das Erste zum letzten.

```
var studium = ["Philosophie", "Chemie", "Medizin", "Jus"];
studium.reverse();
window.alert(studium);
```

**JS** `arrayname.sort(function(a, b) {return a - b});`



Da die `.sort()` Methode nur Strings in eine alphabetische Reihenfolge sortiert und keine Zahlen aufsteigend (z. B. 1, 2, 3, 33, 4, 456, 5 ...) gibt es einen Trick um das Problem zu umgehen. Die Funktion(a, b) übernimmt zwei Werte und subtrahiert sie. Je nachdem wie das Ergebnis ausfällt (Minus, Null, Plus) wird die Zahl auch sortiert.

```
var punkte = [50, 101, 1, 5, 25, 12];
punkte.sort(function(a, b) {return a - b});
window.alert(punkte);
```

**JS** `arrayname.forEach(eineFunktion);`



Startet eine Funktion für jeweils jedes Element im Array. Übergeben wird der Wert und der Index → deshalb sollte die Funktion beide Variablen vereinbaren.

```
var zahlen = [65, 44, 12, 4, „Kein Wert“];

zahlen.forEach(myFunction);

function myFunction(item, index) {
  window.alert("Auf Position: " + index + " ist der Wert " + item);}
```



Selbiges erzielt man auch mit einer `for...in` Schleife. Diese durchwandert jedes Element im Array. `for...in` kann auch auf einen String angewandt werden, dann wandert die Schleife jeden Buchstaben ab.

```
for (var index in zahlen) {
  window.alert("Bei: " + index + " ist der Wert " + zahlen[index]);}
```