

Schleifen



Übersicht

- While
- Do-While
- For
- For in
- For of
- Break und continue
- Übung zu Schleifen



While-Schleife

- Schleifen definieren einen Code-Block, der so lange ausgeführt wird, so lange die gegebene Bedingung erfüllt ist.
- Die einfachste Form ist die while-Schleife, auf das Schlüsselwort while folgt die Bedingung und anschließend der Code-Block.
- Dieser wird so lange wiederholt, so lange die Bedingung erfüllt ist.
- Ist die Bedingung von Anfang an nicht erfüllt, so wird der Code-Block nicht ausgeführt.
- **Vorsicht:** Wird die Bedingung immer erfüllt, habt ihr eine Endlosschleife, das Programm wird abstürzen!

```
let num = 2;

while (num < 1000) {
  console.log(num);
  num = num ** 2;
}
```

Do-while-Schleife

- Die do-while-Schleife wird im Gegensatz zur while Schleife immer mindestens einmal ausgeführt, da die Bedingung erst am Ende geprüft wird. Auf das Schlüsselwort
- do folgt der Code-Block, erst dann folgt das Schlüsselwort while und die Bedingung.
- Nach der Bedingung wird ein Semikolon gestetzt. Diese Schleife wird häufig genutzt,
- um Benutzereingaben zu validieren und bei Falscheingabe zu wiederholen.

```
let count = 0;

do {
  console.log(`Schleifeniteration: ${count}`);
  count++;
} while (count < 5);
```

for-Schleife

- Die for-Schleife (auch Zählschleife) besitzt einen Zähler, der von einem Startwert bis zu einem gegebenen Wert zählt.
- Nach dem Schlüsselwort for folgt eine Klammer () in der drei Teile getrennt durch Semikolon definiert werden.
- Im ersten Teil wird eine Zählvariable mit einem Startwert initialisiert,
- Der zweite Teil ist die Bedingung, bis wann die Schleife auszuführen ist
- Im dritten Teil wird die Zählvariable erhöht (das passiert immer nach der Ausführung des Code-Blocks)
- Danach folgt der Code-Block.
- Die Zählvariable wird meist **i** genannt, wird **i** bereits verwendet, dann **j**, **k** usw.

```
let count = 0;

do {
  console.log(`Schleifeniteration: ${count}`);
  count++;
} while (count < 5);
```

for-Schleife bei Arrays

- Man nutzt for-Schleifen häufig, wenn man mit Arrays arbeitet, und die einzelnen Werte des Arrays durch-iterieren ("durchgehen") möchte.
- Hier ist zu beachten, dass Arrays bei 0 beginnen, und der letzte Wert also an der Stelle Arraylänge minus 1 ist, deshalb wird das < in der Bedingung und nicht <= verwendet.

```
const days = ["Mo", "Tu", "We", "Th", "Fr", "Sa", "Su"];

for (let i = 0; i < days.length; i += 1) {
  console.log(days[i]); // => "Mo", "Tu", "We", "Th", "Fr", "Sa", "Su"
}
```

for-in Schleife

- Über Arrays zu iterieren geht in JS aber auch einfacher! Die for-in Schleife kümmert sich selbstständig um die Zählvariable, und geht die Länge eines Arrays einmal vollständig durch.
- Anstelle der drei Teile in der Klammer wird hier nur eine Zählvariable definiert, gefolgt von dem Schlüsselwort in, gefolgt vom Array über das iteriert werden soll.
- Die for-in Schleife kann auch über Objekte iterieren, dann werden in der "Zählvariable" nacheinander die Schlüssel hinterlegt.

```
const car = { brand: "Suzuki", model: "Jimny", doors: 2 };  
  
for (let i in car) {  
    console.log(i);           // => "brand", "model", "doors"  
    console.log(car[i]);      // => "Suzuki", "Jimny", 2  
}
```

for-of Schleife

- Ist man nur an den Werten im Array oder Objekt interessiert, bietet sich die for-of Schleife an
- dabei wird in die Variable jeweils nur der Wert gespeichert, den Index/Schlüssel erhalten wir nicht.
- Hierbei wird das Schlüsselwort of anstelle von in verwendet.
- **Vorsicht**, man verwechselt die beiden sehr leicht, zum Merken: in für index

```
const car = { brand: "Suzuki", model: "Jimny", doors: 2 };  
  
for (let value of car) {  
    console.log(value); // => "Suzuki", "Jimny", 2  
}
```


break und continue

- Innerhalb von Schleifen können die Schlüsselwörter break und continue genutzt werden um die aktuelle Ausführung des Code-Blocks an dieser Stelle abubrechen.
- Bei break wird die gesamte Schleife gestoppt
- bei continue wird mit der nächsten Iteration fortgefahren.

```
const numbers = [1, 3, -2, 5, -7, 8];  
  
for (let value of numbers) {  
  if (value < 0) {  
    break;  
  }  
  console.log(value); // => 1, 3  
}  
  
for (let value of numbers) {  
  if (value < 0) {  
    continue;  
  }  
  console.log(value); // => 1, 3, 5, 8  
}
```

Übungen zu Schleifen

- FizzBuzz:
Gib alle Zahlen von 1 – 100 aus, jedoch werden Zahlen die durch 3 teilbar sind durch das Wort „Fizz“ ersetzt, Zahlen die durch 5 teilbar sind werden durch „Buzz“ ersetzt und Zahlen die durch 3 und 5 teilbar sind werden durch „FizzBuzz“ ersetzt.
Nutze dafür eine **while loop**!
Beispiel Output: 1,2,Fizz, 4,Buzz,Fizz,7,8,....
- 99 Bottles of Beer on the Wall:
Schreibe ein Programm bei dem der Benutzer eine Zahl über einen Prompt eingibt. Als Ergebnis wird in einem Alert der Songtext “99 bottles of beer” ausgegeben. Die Anzahl der Flaschen zu Beginn des Songtexts wird über die eingegebene Zahl bestimmt
Nutze dafür eine **for loop**!
Beispiel Output: 21 bottles of beer on the wall, 21 bottles
- Fibonacci (optional):
Erstelle eine Funktion die die Fibonacci-Folge bis zu einem bestimmten Wert generiert.
Beispiel: generateFibonacci(10)
Output: 0,1,1,2,3,5,8,13,21,34

Lyrics of the song 99 Bottles of Beer

99 bottles of beer on the wall, 99 bottles of beer.
Take one down and pass it around, 98 bottles of beer on the wall.

98 bottles of beer on the wall, 98 bottles of beer.
Take one down and pass it around, 97 bottles of beer on the wall.

97 bottles of beer on the wall, 97 bottles of beer.
Take one down and pass it around, 96 bottles of beer on the wall.

96 bottles of beer on the wall, 96 bottles of beer.
Take one down and pass it around, 95 bottles of beer on the wall.

<https://99-bottles-of-beer.net/lyrics.html>

Viel Erfolg beim Entwickeln!

