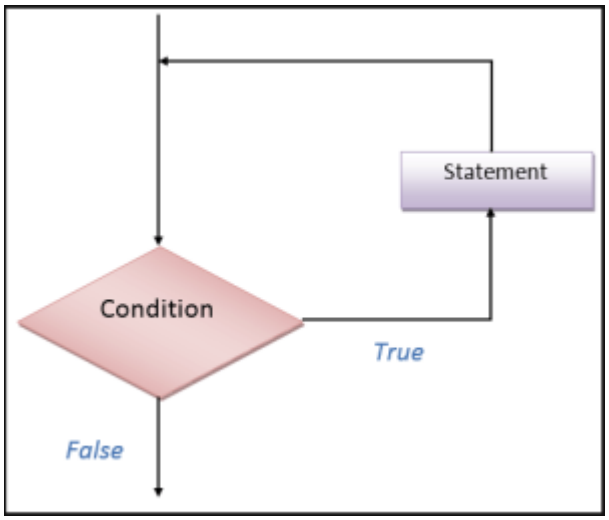
LU04a - Loops/Schleifen Übersicht

Einleitung bzw. Problemstellung

Stellen Sie sich vor, Sie sollen 100 Zahlen auf die Konsole schreiben. Das ginge mit:

```
console.log(1);
console.log(2);
console.log(3);
// ... bis 100
```

Das eben gezeigte Beispiel umfasst nur 100 Zahlen-Elemente, aber stellen Sie sich vor es seien tausende oder Millionen (Ergebnis einer Google-Suche). Sehr unpraktisch und in der Realität so nicht handhabbar. Wir brauchen also eine Verarbeitungsstruktur, die dynamisch die Lösungsmenge verarbeitet. Wir brauchen also stattdessen **Schleifen** – sie wiederholen Anweisungen, solange eine bestimmte Bedingung erfüllt ist.

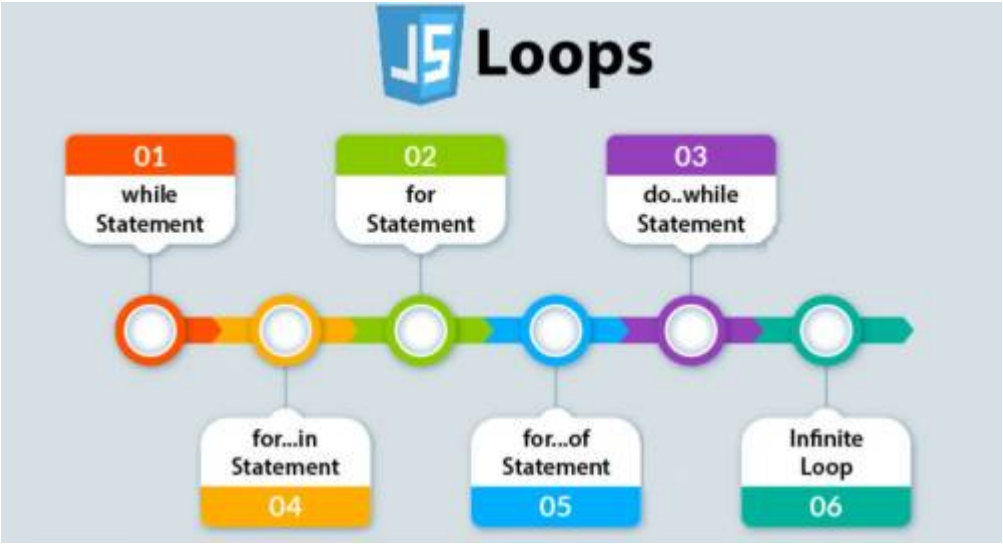


Lösung: Schleifen

Um eine grosse Anzahl von Elementen, die beispielsweise in Arrays gepeichert sind verarbeiten zu können, verwenden wir am besten Schleifen. Man könnte sogar sagen, dass *Schleifen* und *Arrays* ein perfektes Paar abgeben, weil

- **Arrays** können eine beliebige Anzahl von Elementen speichern/aufnehmen.
- **Schleifen** können diese beliebige Anzahl von Elemente verarbeiten.

In fast jeder Programmiersprache, so auch in JavaScript, gibt es die nachfolgende Typen:



Wir haben nicht Zeit für alle Loop-Arten. Innerhalb des Moduls 288 werden wir uns daher auf die nachfolgenden drei Loops konzentrieren.

1. **for-Schleife:** Zählerschleife, kopfgesteuert
2. **while-Schleife:** Bedingung, kopfgesteuert
3. **for … in:** Bedingung, kopfgesteuert

Wann ist welche Schleife passend?

Typ	Wann nutzen?
for	Wenn die Anzahl der Wiederholungen bekannt sind.
while	Wenn nur eine Bedingung überprüfen werden soll.
for…in	Wenn Objekte durchgegangen erden sollen.

Achtung: Endlosschleifen sind der Klassiker für eingefrorene Programme.

Lernvideos

Schleifen im JavaScript

Zusatzmaterial

- W3School- JavaScript Loops
- SelfHTML - Schleifen

Volkan Demir

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m288/learningunits/lu04/01?rev=1758609213

Last update: **2025/09/23 08:33**