

Loops - Schleifen

Schleifen führen eine Aktion für eine bestimmte Anzahl an Wiederholungen aus (diese Anzahl kann auch 0 sein).

Dazu wird eine Bedingung festgelegt, solange diese Bedingung zutrifft, wird die Aktion durchgeführt.

Dazu wird ein Schleifenzähler (Index) initialisiert, eine Bedingung festgelegt und eine Endzuweisung (immer wiederkehrende Beeinflussung des Schleifenzählers) initialisiert.

Als Iteration wird der Schleifendurchlauf bezeichnet.

Es gibt verschiedene Schleifenarten, die gebräuchlichste ist die For-Schleife.

For-Schleife

Syntax:

```
for ([Schleifenzähler]; [Bedingung]; [Endzuweisung]) {  
    Iteration = Code  
}
```

```
for (var i = 0; i < 3 i++) {  
    console.log(`Das ist der ${i+1}. Ausdruck bei Index i = ${i}`);  
}
```

// gibt auf der Console aus:

Das ist der 1. Ausdruck bei Index i = 0

Das ist der 2. Ausdruck bei Index i = 1

Das ist der 3. Ausdruck bei Index i = 2

Solange die Bedingung zutrifft, wird der Code ausgeführt. Die Endzuweisung erfolgt am Ende eines Schleifendurchgangs.

Sollte die Bedingung gleich am Anfang nicht zutreffen, wird der Code überhaupt nicht ausgeführt.

Dies gilt ebenso für While-Schleifen.

While-Schleife

Syntax:

Initialisierung des Schleifenzählers;

```
while (Bedingung) {  
    Iteration = Code;  
    Enzuweisung;  
}
```

While-Schleifen bergen die Gefahr von Endlosschleifen wenn die Endzuweisung am Ende der Aktion falsch definiert ist.

```
var j = 0;
```

```
while (j < 3 ) {  
  console.log(`Das ist der ${j+1}. Ausdruck bei Index j = ${j}`);  
  j++;  
}
```

// gibt auf der Console aus:

Das ist der 1. Ausdruck bei Index j = 0

Das ist der 2. Ausdruck bei Index j = 1

Das ist der 3. Ausdruck bei Index j = 2

Do-While-Schleifen

Do-While-Schleifen werden im Gegensatz zu For-Schleifen und While-Schleifen mindestens einmal ausgeführt, weil die Bedingung erst am Ende geprüft wird.

Syntax:

Initialisierung des Schleifenzählers;

```
do {  
  Iteration = Code;  
  Endzuweisung;  
} while (Bedingung),
```

```
var k = 0;
```

```
do {  
  console.log(`Das ist der ${k+1}. Ausdruck bei Index k = ${k}`);  
  k++;  
} while (k > 100)
```

// gibt auf der Console aus:

Das ist der 1. Ausdruck bei Index k = 0

Hier wird also der Code einmal ausgeführt **obwohl** die Bedingung nicht zutrifft, da die Bedingung erst nach dem ersten Ausführen des Codes geprüft wird.

Break - Abbrechen einer Schleife

Eine Schleife kann durch den Break-Befehl innerhalb des Durchlaufen in Verbindung mit einer If-Condition beendet werden

```
for (var i = 0; i < 10; i++) {  
  console.log(`Der Schleifenzähler hat den Wert = ` + i);  
  if (i === 3) {  
    break;  
  }  
}
```

Sobald der Schleifenzähler den Wert 3 erreicht hat, wird die Schleife beendet und nicht weiter durchlaufen.

```
// auf der Console wird ausgegeben:  
Der Schleifenzähler hat den Wert = 0  
Der Schleifenzähler hat den Wert = 1  
Der Schleifenzähler hat den Wert = 2  
Der Schleifenzähler hat den Wert = 3
```

Continue - Sofortige Iteration innerhalb einer Schleife

Ein Durchlauf der Schleife kann durch den Continue-Befehl in Verbindung mit einer If-Condition beendet werden, somit wird Code nach dem Continue bei diesem Schleifendurchlauf nicht mehr ausgeführt.

Es tritt sofort die Enzuweisung ein und die Schleife wird erneut durchlaufen.

```
for (j = 0; j < 4; j++) {  
  if (j === 2) {  
    continue;  
  }  
  console.log('Der Schleifenzähler j hat den Wert = ' + j);  
}
```

Sobald der Schleifenzähler j den Wert 2 erreicht, wird die Iteration (der Schleifendurchlauf) abgebrochen und eine Endzuweisung durchgeführt.

```
// auf der Console wird ausgegeben  
Der Schleifenzähler j hat den Wert = 0  
Der Schleifenzähler j hat den Wert = 1  
Der Schleifenzähler j hat den Wert = 3
```

Questions? You're welcome!

holger.zerbe@digitalcareerinstitute.org