

while und for-Schleifen in Java

Schleifen sind Anweisungen, die die wiederholte Ausführung von Aktionen ermöglichen

Eine **for-Schleife** wird dann eingesetzt, wenn die **Anzahl der Wiederholungen** bereits **bekannt** ist.

Die Syntax einer for- Anweisung sieht so aus:

```
for (starting value ; condition ; change value){ // Loop body - if 'true' do this ; }
```

Was passiert, wenn dieser Code ausgeführt wird?

1. Startwert des **Zählers** setzen.
Auswerten des booleschen Wertes der Bedingung, die in () nach dem **ersten ;** steht.
2. Ist der Wert **true**, werden die Anweisungen im Schleifenkörper { } ausgeführt.
3. Danach wird der **Zähler geändert**, wie es in der () nach dem **zweiten ;** steht
4. Ist der Wert **false**, geht es mit der ersten Anweisung nach der } weiter.

Eine **while-Schleife** kann überall dort eingesetzt werden, wo eine Anweisung oder eine Gruppe von Anweisungen mehrmals ausgeführt werden muss, **bis ein bestimmter Status erreicht** ist.

Die Syntax einer while- Anweisung sieht so aus:

```
while (condition) { // Loop body - if 'true' do this ; change condition!! }
```

Was passiert, wenn dieser Code ausgeführt wird?

1. Auswerten des booleschen Wertes der Bedingung, die in () nach while steht.
2. Ist der Wert **true**, werden die Anweisungen im Schleifenkörper { } ausgeführt.

Damit keine Endlosschleife entsteht ist es wichtig im Schleifenkörper die Bedingung zu beeinflussen. Danach, fahren wir mit Schritt 1 fort.

3. Ist der Wert **false**, geht es mit der ersten Anweisung nach der } weiter.

Beispiele - die beiden jeweiligen Codeausschnitte sind identisch.

while-Schleife

```
int i = 0;
while (i < 20){
    System.out.println(i); i++; }
```

```
int a = 3;
while (a >= 0){
    System.out.println(i); i--; }
```

```
int i = 0;
while (i < 3){
    System.out.println(i);i++;}
```

for-Schleife

```
for (int i = 0; i < 20; i++){
    System.out.println(i);}
```

```
for (int a = 3; a >= 0; a--){
    System.out.println(i);}
```

```
for (int i = 0; i < 3; i++){
    System.out.println(i); }
```