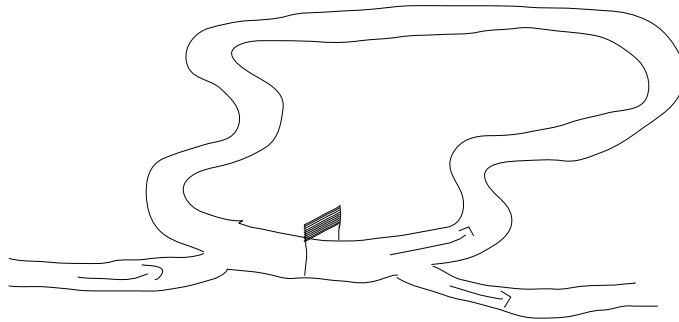


# Arbeitsblatt 8 - for-Schleife

Eine for-Schleife dient zur wiederholten Durchführung:



Die Kontrollstruktur **for-Schleife** ist eine **Zähl-Schleife**. Sie ermöglicht, dass in Abhängigkeit von einer **Bedingung** bestimmte Anweisungen **solange immer wieder** ausgeführt werden, bis die Bedingung **nicht mehr** erfüllt ist.

```
for (Initialisierung; Bedingung; Aktualisierung) {
    Anweisungen;
}
```

Die Parameterliste einer for-Schleife besteht aus **drei Teilen**:

## Initialisierung

→ Die **Laufvariable** wird auf einen **Anfangswert** gesetzt (Dies erfolgt nur bei der ersten Ausführung der Schleife!)

## Bedingungsprüfung

→ Die Bedingung legt das **Abbruchkriterium** der Schleife fest

## Aktualisierung

→ Am Ende eines Durchlaufs wird die **Laufvariable** **aktualisiert**

Wir schauen uns mal ein konkretes Beispiel an:

```

5  int ledNummer;
6  for (ledNummer=1; ledNummer<5; ledNummer++){
7      bob3.setLed(ledNummer, WHITE);
8  }
    
```

In **Zeile 5** wird unsere **Laufvariable ledNummer** deklariert, also neu eingeführt. Sie ist vom Typ **int**, kann also ganze Zahlen speichern.

In **Zeile 6** beginnt unsere for-Schleife, wir starten unseren **ersten Durchlauf**:

- Schritt 1 – Initialisierung:** Die Variable **ledNummer** wird mit 1 initialisiert, also auf den Wert 1 gesetzt.
- Schritt 2 – Bedingungsprüfung:** Wir überprüfen unsere Bedingung: Ist der Wert von **ledNummer** kleiner als 5?
- Schritt 3 – Ausführung:** **Falls** die Bedingung **wahr** ist, werden nun die Anweisungen ausgeführt. In unserem Beispiel ist dies die Anweisung **bob3.setLed(ledNummer, WHITE);**
- Schritt 4 – Aktualisierung:** **Aktualisierung** der Variablen **ledNummer**. Der Ausdruck **ledNummer++** bedeutet, dass **ledNummer** um 1 erhöht wird.
- Schritt 5 – Wiederholung:** **Wiederholung ab Schritt 2.** Wir starten also einen erneuten Durchlauf durch unsere Schleife, wobei der Wert der Variablen **ledNummer** jetzt 2 ist.

!!! Es werden dann **solange** die Schritte 2 bis 5 abgearbeitet, bis die Bedingungsprüfung **falsch** ergibt !!!



**Aufgabe 1:** Betrachte das folgende Programm. Wie oft wird die Anweisung `bob3.setLed(ledNummer, WHITE);` ausgeführt?

```

5   int ledNummer;
6   for (ledNummer=1; ledNummer<5; ledNummer++){
7       bob3.setLed(ledNummer, WHITE);
8   }

```

- einmal
- zweimal
- dreimal
- viermal
- fünfmal

**Aufgabe 2:** Betrachte das folgende Programm. Wie oft wird die Anweisung `bob3.setLed(ledNummer, WHITE);` ausgeführt?

```

5   int ledNummer;
6   for (ledNummer=1; ledNummer<4; ledNummer++){
7       bob3.setLed(ledNummer, WHITE);
8   }

```

- einmal
- zweimal
- dreimal
- viermal
- fünfmal

**Aufgabe 3:** Betrachte das folgende Programm. Wie oft wird die Anweisung `bob3.setLed(ledNummer, WHITE);` ausgeführt?

```

5   int ledNummer;
6   for (ledNummer=3; ledNummer<4; ledNummer++){
7       bob3.setLed(ledNummer, WHITE);
8   }
    
```

- einmal
- zweimal
- dreimal
- viermal
- fünfmal

**Aufgabe 4** Betrachte das folgende Programm. Wie oft wird die Anweisung `bob3.setLed(ledNummer, WHITE);` ausgeführt?

```

5   int ledNummer;
6   for (ledNummer=4; ledNummer>0; ledNummer--){
7       bob3.setLed(ledNummer, WHITE);
8   }
    
```

- einmal
- zweimal
- dreimal
- viermal
- fünfmal

**Aufgabe 5** Betrachte die folgende for-Schleife. Welche Werte nimmt die Variable **i** im jeweiligen Durchlauf an?

```

6   for (i=0; i<10; i=i+2){
7       ...
8   }
```

Durchlauf 1: **i**=\_\_\_\_\_

Durchlauf 2: **i**=\_\_\_\_\_

Durchlauf 3: **i**=\_\_\_\_\_

Durchlauf 4: **i**=\_\_\_\_\_

Durchlauf 5: **i**=\_\_\_\_\_

**Aufgabe 6** Betrachte die folgende for-Schleife. Welche Werte nimmt die Variable **j** im jeweiligen Durchlauf an?

```

6   for (j=1; j<64; j=j*2){
7       ...
8   }
```

Durchlauf 1: **j**=\_\_\_\_\_

Durchlauf 2: **j**=\_\_\_\_\_

Durchlauf 3: **j**=\_\_\_\_\_

Durchlauf 4: **j**=\_\_\_\_\_

Durchlauf 5: **j**=\_\_\_\_\_

Durchlauf 6: **j**=\_\_\_\_\_