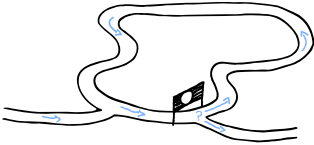


W2

Schleifen

Thema:	Schleifen
Bereich:	Know-How
Voraussetzung:	Station W1
Lernziele:	Bedeutung und Anwendung von Schleifen kennenlernen, «Wiederhole x-mal»-Schleife, «Wiederhole bis»-Schleife, Endlosschleife, Abbruch von Schleifen
Anspruch:	★★☆☆☆
Aufgaben:	A1 – A9
Zeitbedarf:	30 min



Beim Programmieren ist es manchmal nützlich, wenn man einen bestimmten Programmteil **mehrfach wiederholen** kann. Für diesen Zweck gibt es die Kontrollstruktur **Schleife**. Wir verwenden drei verschiedenen Schleifen:



«**wiederhole x mal**»-Schleife

die Zahl suchst du selber aus 💡



«**wiederhole bis**»-Schleife



«**wiederhole fortlaufend**»-Schleife

Wir schauen uns zunächst die «**wiederhole x mal**»-Schleife an:

Wissensbox

«**wiederhole x mal**»-Schleife

Die «*x-mal*»-Schleife ermöglicht, dass alle Anweisungen, die innerhalb der Schleife stehen genau *x-mal* ausgeführt werden.

Wir schauen uns dazu ein Beispiel an, Bob soll uns genau **3-mal** zuzwinkern:



- Aufgabe 1:** Probiere das Beispiel aus, verwende den «**wiederhole 3 mal**»-Block aus dem Bereich **Kontrolle** und zähle mit! Wie oft zwinkert Bob dir zu?
- Aufgabe 2:** Ändere dein Programm: Bob soll dir jetzt **acht mal** zuzwinkern. Hast du eine Idee, wie das geht? Probiere mal!
- Aufgabe 3:** Erweitere dein Programm: Bob soll dir zuerst **acht mal** mit **Auge 1** zuzwinkern. Danach soll er mit beiden **Bauch-Leds** schnell **10 mal** aufblitzen!
- Aufgabe 4:** Jetzt mal mehr Tempo! Ändere die **Geschwindigkeit**: Bob soll dir wieder zuerst **acht mal** mit **Auge 1** freundlich zuzwinkern. Dann soll er mit den beiden **Bauch-Leds** sehr sehr schnell **20 mal** aufblitzen. Und zwar so schnell, dass man kaum noch mitzählen kann!

Tip: Ein schönes **Aufblitzen** mit den Leds entsteht, wenn man zuerst ganz **kurz** abwartet (z.B. 20 Millisekunden) und dann **länger** wartet (z.B. 300 Millisekunden):

```

    Mache einmal am Anfang
    wiederhole 8 mal
        setze Augenfarbe auf gelb
        warte 200 Millisekunden
        setze Augenfarbe auf schwarz
        warte 200 Millisekunden
    wiederhole 20 mal
        schalte Bauchled ein
        warte 20 Millisekunden
        schalte Bauchled aus
        warte 300 Millisekunden
    
```

Jetzt schauen wir uns die «**wiederhole bis**»-Schleife an:

Wissensbox

«**wiederhole bis**»-Schleife

Mit der «**wiederhole bis**»-Schleife werden alle Anweisungen innerhalb der Schleife solange wiederholt, bis die Bedingung wahr ist.



Wir probieren mal ein Beispiel aus:

Bob soll wie eine **Auto-Diebstahlsicherung** funktionieren. Solange kein Dieb da ist, der IR-Sensor also keinen Dieb bemerkt, wartet Bob ab. Während er wartet, soll er mit den Augen in dunkelrot kurz aufblitzen. Sobald der IR-Sensor einen Wert größer als 15 bemerkt, (wenn also z.B. deine Hand vor dem Sensor ist) dann soll er mit den Bauch-Leds ein weisses Warnblinklicht machen!



- **Aufgabe 5:** Probiere das Beispiel aus, verwende den «**wiederhole bis**» Block und einen «**wiederhole 50 mal**» Block aus dem Bereich **Kontrolle**. Dann halte mal deine Hand oder ein Blatt Papier vor den IR-Sensor. Was macht der Bob?

- **Aufgabe 6:** Ändere dein Programm:
 Die Bauch-Leds sollen jetzt mal **abwechselnd blinken**.
 Wenn die linke Bauch-Led an ist, dann soll die rechte
 aus sein und umgekehrt!

Im Moment funktioniert unsere **Auto-Diebstahlsicherung** so, dass genau 50 mal das Alarmblinken stattfindet. Eigentlich müsste Bob aber solange Alarm anzeigen, bis der Besitzer des Autos den Alarm abschaltet!

Für diesen Zweck lernen wir jetzt die «**wiederhole fortlaufend**»-Schleife kennen:

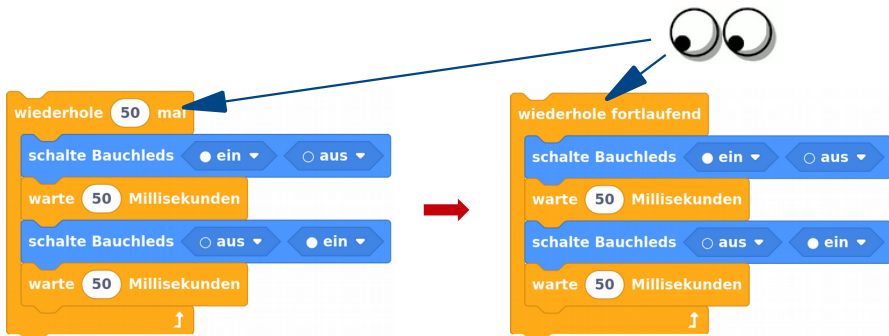
Wissensbox

«wiederhole fortlaufend»-Schleife

Die «wiederhole fortlaufend»-Schleife ist eine sogenannte **Endlosschleife**. Alle Anweisungen innerhalb der Schleife werden immer wieder wiederholt.

wiederhole fortlaufend

- **Aufgabe 7:** Ersetze den «wiederhole 50 mal»-Block durch einen «wiederhole fortlaufend»-Block und teste dein Programm! Welchen Unterschied stellst du fest?



Unsere Auto-Diebstahlsicherung wäre jetzt also **für immer** im Alarm Modus. Oder zumindest solange, bis die Batterie leer ist ;-)

Jetzt programmieren wir noch eine **Alarm-Reset** Funktion für den Besitzer des Autos, damit dieser das Alarmblinken auch wieder abstellen und den Sensor-Detektion Modus wieder starten kann. Hierfür benötigen wir den **«die Schleife abbrechen»**-Block:

die Schleife abbrechen

Wissensbox

Abbruch einer Schleife

Mit dem **«die Schleife abbrechen»**-Block kann man jede Schleife direkt **beenden**.

- Aufgabe 8:** **Erweitere** dein Programm: Mit Berührung der Armsensoren soll der **Alarm neu gestartet** werden. **Falls** einer der beiden Arme (Arm 1 oder Arm 2) berührt wird, **dann** soll das Alarmblinken ausgeschaltet und die Schleife abgebrochen werden!

Wahrheitswerte

- falsch = falsch
- falsch und falsch
- falsch oder falsch
- falsch exklusiv oder falsch
- nicht falsch
- wahr ▾



Tip:
 Verwende den **„oder“** Block aus der Rubrik Operatoren

Dein Programm sollte jetzt in etwa so aussehen:

```

    C Mache immer wieder
    wiederhole bis hole IR-Sensor-Wert > 15
    setze Augenfarben [rot] [rot]
    warte 50 Millisekunden
    setze Augenfarben [schwarz] [schwarz]
    warte 1000 Millisekunden
    wiederhole fortlaufend
    schalte Bauchleds [ein] [aus]
    warte 50 Millisekunden
    schalte Bauchleds [aus] [ein]
    warte 50 Millisekunden
    falls wird Arm1 irgendwo berührt? oder wird Arm2 irgendwo berührt? dann
    setze Augenfarben [schwarz] [schwarz]
    schalte Bauchleds [aus] [aus]
    warte 1000 Millisekunden
    die Schleife abbrechen
    
```



1 x Klick genügt!



aber wo?

Aufgabe 9: Jetzt programmieren wir eine **Profi-Reset-Funktion!** Der Alarm soll nur neu gestartet werden können, wenn man **beide Arme gleichzeitig** berührt. Ändere dein Programm an einer einzigen Stelle! Weisst du wo?