Programmieren lernen mit BOB3

Lehrerbegleitheft - Sekundarstufe I
Einführung in die textuelle Programmierung mit dem Roboter BOB3

Modul 2

DIGITALE BILDUNG: Programmieren lernen mit BOB3, Lehrerhandreichung

Modul 2

Kontrollstrukturen

Dieses Dokument dient zur Übersicht des Kurses „Programmieren lernen mit BOB3“ für die Sekundarstufe I. Es werden die Lehr- und Lernmaterialien, die benötigten Zeiteinheiten und Vorschläge für konkrete Unterrichtseinheiten gegeben. Im Folgenden ist eine Unterrichtseinheit für 45 min ausgelegt.

Übersicht Modul 2:

Modul 2 umfasst insgesamt 6 Unterrichtseinheiten mit je ca. 45 Minuten

Lehrerbegleitheft mit Konzepten und Ablaufplänen der einzelnen Einheiten

12 Programmier-Tutorial Einheiten:

Die Schüler lernen Verzweigungen anhand der Kontrollstruktur ‚if / else‘ kennen und arbeiten mit Vergleichsoperatoren und Wahrheitswerten. In kleinen Programm-beispielen wird die Verzweigung konkret zur Auswertung des IR-Sensors eingesetzt. In diesem Zusammenhang lernen die Schüler das Konzept von Variablen kennen und anwenden. Sie lernen die einzelnen Hardware-Bestandteile des Roboters und die jeweiligen Funktionen kennen. Anhand der Software-Bibliothek des BOB3 setzen sie sich mit Methoden, Funktionen und Parametern auseinander und erlernen als weitere Kontrollstruktur das Prinzip und die Anwendung von ‚for-Schleifen‘. Zum Abschluss der Lerneinheit erfolgt eine einfache Fehlersuche, bei der die Schüler lernen, Fehler im Programmcode zu entdecken und zu beheben.

Arbeitsblatt 5 - „***if-else-Anweisung***“
+ Lösungen zum AB 5

Arbeitsblatt 6 - „***Bob‘s Hardware***“
+ Lösungen zum AB 6

Arbeitsblatt 7 - „***Bob‘s Software***“
+ Lösungen zum AB 7

Arbeitsblatt 8 - „***for-Schleife***“
+ Lösungen zum AB 8

1. Unterrichtseinheit

In der ersten Unterrichtseinheit lernen die Schülerinnen und Schüler die **if-else-Anweisung** als erste **Kontrollstruktur** kennen. In dem Zusammenhang lernen die SuS **Bedingungen**, **Wahrheitswerte** und **Vergleichsoperatoren** kennen und anwenden.

Zeitbedarf: ca. 45 min

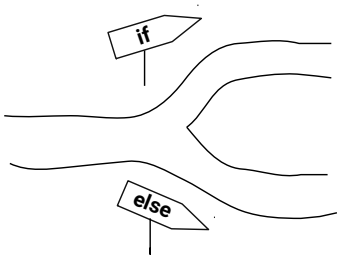
Vorkenntnisse: BOB3 Modul 1

Material: Papier, Stift, Arbeitsblatt 5
PC oder Laptop, BOB3 mit Helm

Ablauf

Die Schüler bearbeiten das **Arbeitsblatt 5** und lernen als erste Programmierstruktur die Kontrollstruktur **„if-else“** kennen:

„Eine if-else-Anweisung ist eine zweiseitige Verzweigung“



Die SuS erlernen die abstrakte Definition einer **zweiseitigen Verzweigung** kennen und verstehen das Konzept anschließend anhand von konkreten Programmier-Beispielen.
Das Arbeitsblatt gibt zusätzlich eine Übersicht über mathematische **Vergleichsoperatoren** und

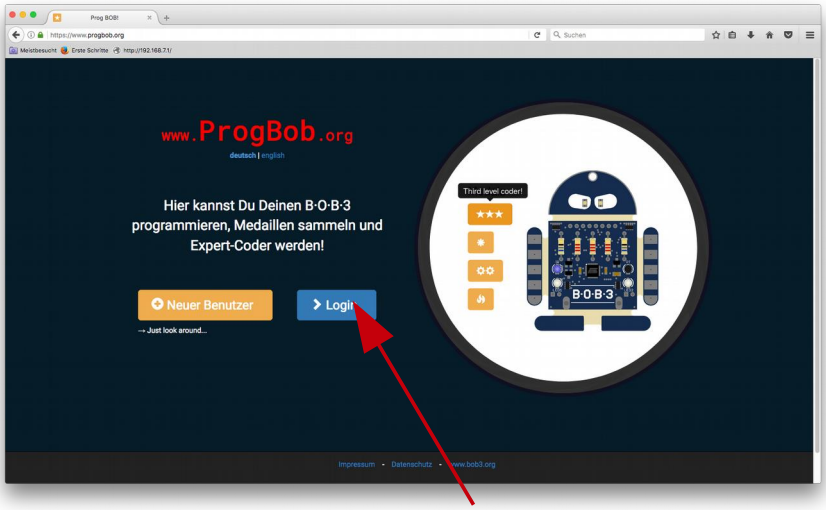
und die entsprechenden Ausdrücke beim Programmieren.

Vergleichsoperatoren:

Mathe	Vergleichsoperator	Beispiel	Erklärung
=	==	a == b	→ ergibt „**wahr**“, falls die Werte a und b **gleich** sind
≠	!=	a != b	→ ergibt „**wahr**“, falls die Werte a und b **ungleich** sind
>	>	a > b	→ ergibt „**wahr**“, falls a **größer** als b ist
<	<	a < b	→ ergibt „**wahr**“, falls a **kleiner** als b ist
≥	>=	a >= b	→ ergibt „**wahr**“, falls a **größer oder gleich** b ist
≤	<=	a <= b	→ ergibt „**wahr**“, falls a **kleiner oder gleich** b ist

Anschließend bearbeiten die SuS die Aufgaben 1 – 8 des Arbeitsblatts und besprechen ihre Lösungen.

Die Schüler loggen sich mit ihren jeweiligen Accounts auf der Seite **http://www.ProgBob.org** ein und starten das **„Intro II“-Tutorial**:



!! Wichtig: erneutes Einloggen mit bereits vorhandenen Daten über „**Login**“ !!

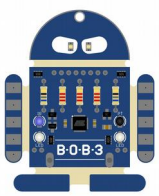
www.bob3.org

Die Schüler bearbeiten die **erste Tutorial-Einheit** und vertiefen mit dem Szenario **„BOB3 als Wahrheitsfinder“** das zuvor erworbene Wissen:

```
#include <BOB3.h>

// B-O-B-3 als Wahrheitsfinder

void setup() {

}

void loop() {

  // Falls die Bedingung wahr ist -> Augen grün
  if (1 == 1) {
    bob3.setEyes(GREEN, GREEN);
  }
  // Sonst -> Augen rot
  else {
    bob3.setEyes(RED, RED);
  }

}

```

In der Lerneinheit wird zunächst eine Bedingung vorgegeben, die von den Schülern ausprobiert wird. Anschließend bekommen sie die Aufgabe, verschiedene andere Bedingungen einzugeben und auf dem BOB3 zu testen. Falls die eingegebene Bedingung **wahr** ist, dann leuchten am Bob die Augen **grün**. Falls die Bedingung **falsch** ist, dann leuchten die Augen **rot**.

In der **Wissensüberprüfungs-Einheit** des ersten Teils werden anschließend noch einzelne Aufgaben zum Thema **Bedingungen** bzw. **wahr** oder **falsch** gestellt. Die Schüler bearbeiten diese Aufgaben und können ihre Lösung direkt auswerten lassen und reflektieren!



www.bob3.org

2. Unterrichtseinheit

In der zweiten Unterrichtseinheit bearbeiten die Schülerinnen und Schüler die **Lerneinheiten zwei** und **drei** des **„Intro II“**-Tutorials, die sich ebenfalls mit der Kontrollstruktur **„if-else“** beschäftigen. Sie experimentieren mit dem bestehenden Programm und vertiefen ihre Kenntnisse über die Programmierung mit **Vergleichsoperatoren „!=“**, **„<“** und **„>“**. Als Anwendung des Gelernten bietet die dritte Lerneinheit das Programm, **„Grundschüler oder nicht?“**, mit dem sich die Schüler anhand eines kleinen Algorithmus einklassifizieren können.

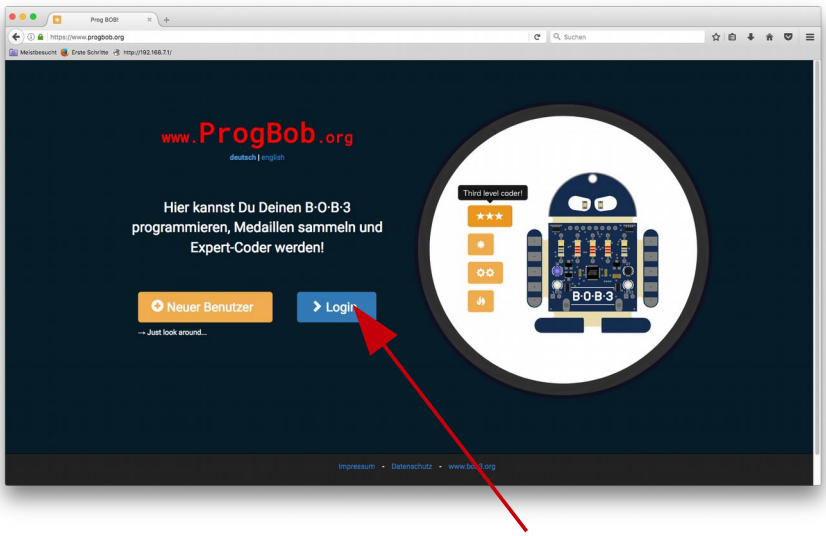
Zeitbedarf: ca. 45 min

Vorkenntnisse: BOB3 Modul 1

Material: PC oder Laptop, BOB3 mit Helm

Ablauf

Die SuS starten den Webbrowser, gehen auf die Seite **http://www.ProgBob.org**, loggen sich mit Ihrem Account ein und gehen zum **„Intro-II“-Tutorial**:



„Login“

Als Beispiel aus der **eigenen Erlebniswelt** der Schüler probieren sie nun ein Programm aus, das anhand des **Geburtsjahres** einer Person ermittelt, ob diese Person noch in die Grundschule geht oder nicht mehr! Die Schüler geben ihr eigenes Geburtsjahr ein und können mit dem Programm experimentieren.

```
1  #include <BOB3.h>
2
3  // BOB3 als Wahrheitsfinder:
4  // Grundschüler oder nicht?
5
6  void setup() {
7
8  }
9
10 void loop() {
11
12   int geburtsjahr;
13   geburtsjahr = 2005;
14
15   // Grundschüler detektiert -> Augen orange
16   if (geburtsjahr > 2007) {
17     bob3.setEyes(ORANGE, ORANGE);
18   }
19   // Sonst -> Augen blau
20   else {
21     bob3.setEyes(ROYALBLUE, ROYALBLUE);
22   }
23
24 }
```

Die Lerneinheit erklärt den Sinn und die Verwendung einer **Variablen**. Die SuS lernen, wie eine Variable **deklariert** und **initialisiert** wird. Anschließend wird der Variable ein konkreter Zahlenwert **zugewiesen**. In der folgenden if-else-Abfrage wird die Variable in der Bedingung verwendet.

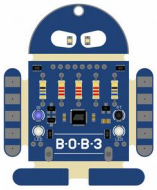
__info__

In Zeile 13 wird unsere neue Variable **initialisiert**, das heißt sie bekommt erstmalig einen Wert zugewiesen:

`geburtsjahr = 2005;`

Für den Compiler ist ab jetzt die Variable `geburtsjahr` gleichbedeutend mit der Zahl `2005`.

Jetzt können wir in unserem eigentlichen Wahrheitsfinder (Zeile 16 - 22) die Variable verwenden!



www.bob3.org

3. Unterrichtseinheit

In der dritten Unterrichtseinheit bearbeiten die Schülerinnen und Schüler die **Lerneinheiten vier** und **fünf** des „**Intro II**“-Tutorials, die sich mit dem **IR-Sensor** des Roboters beschäftigen. Sie lernen, einen **Sensorwert abzufragen**, das Ergebnis in einer **Variablen zu speichern** und im folgenden Programm mit dieser Variablen zu arbeiten. Je nach Variablenwert soll der Roboter unterschiedlich reagieren.

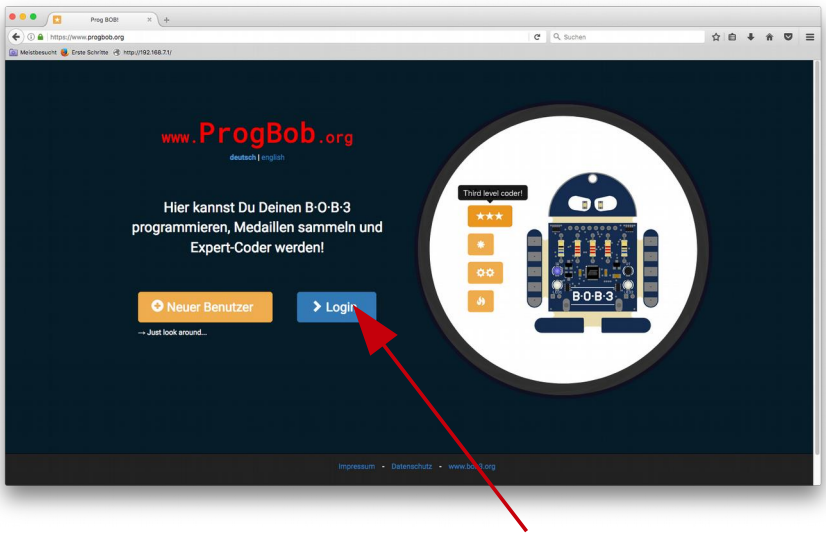
Zeitbedarf: ca. 45 min

Vorkenntnisse: BOB3 Modul 1

Material: Papier, Stift, Arbeitsblatt 6
PC oder Laptop, BOB3 mit Helm

Ablauf

Die SuS starten den Webbrowser, gehen auf die Seite **http://www.ProgBob.org**, loggen sich mit Ihrem Account ein und gehen zum „**Intro-II“-Tutorial**:



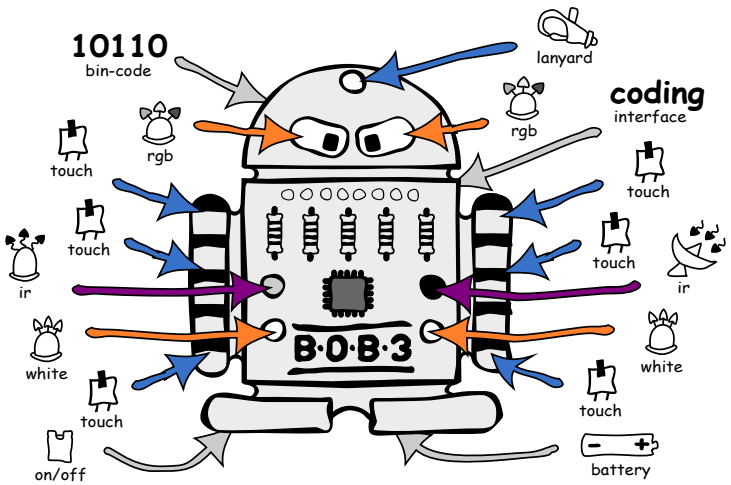
„**Login**“

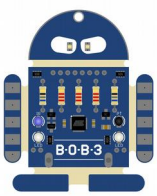
Zunächst probieren die Schüler das bestehende Programm-Beispiel aus: Der Bob soll mit den LEDs anzeigen, dass er mittels seiner **IR-Sensorik** berührungslos auf Objekte **reagieren** kann. Sobald man die Hand oder ein Blatt Papier vor den Roboter hält, wird dies vom IR-Sensor bemerkt und der Bob schaltet die Augen-Leds rot ein. Sobald man die Hand wieder weg nimmt, gehen die Augen wieder aus.

```
#include <BOB3.h>

void setup() {

}

void loop() {

  // Sensorwert abfragen
  int irWert = bob3.getIRSensor();

  // Auf Objekte reagieren
  if (irWert > 8) {
    bob3.setEyes(RED, RED);
  } else {
    bob3.setEyes(OFF, OFF);
  }

  delay(100);
}
```

Die Schüler bearbeiten das **Arbeitsblatt 6**, in dem die **Hardware-Komponenten** des BOB3 aufgeführt und erklärt werden.

„Bob's Hardware"



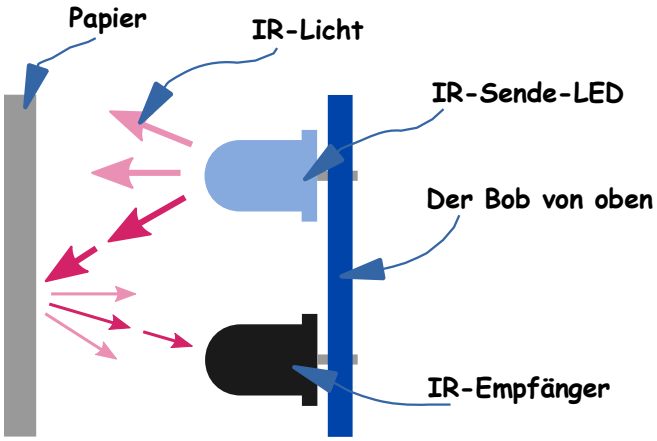


www.bob3.org

Insbesondere lernen die Schüler, wie der **IR-Sensor** des BOB3 funktioniert. Der Sensor besteht aus zwei Teilen: Einer **IR-Sende-LED** und einem **IR-Empfänger**. Auf diese Weise können zwei Roboter-Einheiten drahtlos miteinander kommunizieren oder sich gegenseitig fernsteuern. Beispiele hierzu sind in den späteren Vertiefungs-Tutorials enthalten.

Zusätzlich kann der Roboter mit dem IR-Sensor **berührungslos Objekte „bemerken“** und auf diese reagieren. Er bemerkt auch den Unterschied zwischen **nah** und **fern** und kann darauf verschieden reagieren. Um zu verstehen, wieso der Roboter diese Fähigkeit hat, erlernen die Schüler das Prinzip des **Reflektionsverfahrens**:

Die **IR-Sende-LED** sendet IR-Licht aus, dieses trifft dann auf ein Hindernis (z.B. ein Blatt Papier), wird von dem Hindernis **zurückreflektiert** und kann so von dem IR-Empfänger empfangen werden.
Je näher das Papier vor dem Sensor ist, **desto mehr IR-Licht detektiert** der IR-Empfänger.



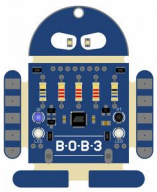
Die Schüler bearbeiten das nächste Programmier-Beispiel und **experimentieren** mit verschiedenen Werten. Sie finden heraus, bei welchen Grenzwerten ihr IR-Sensor nah und fern detektiert und lernen so reale Robotik kennen. Anschließend erweitern die Schüler das Programm um eine „**else-if-Abfrage**“, um noch eine Abstufung einzubauen:

```
12  // Viel Reflektion - Objekt nah
13  if (irWert > 8) {
14    bob3.setEyes(RED, RED);
15
16  // Weniger Reflektion - Objekt weiter weg
17  } else if (irWert > 12) {
18    bob3.setEyes(BLUE, BLUE);
```

__info__

Das Programm soll jetzt so ergänzt werden, dass die Augen bei **wenig** Reflektion **blau** leuchten und bei **viel** Reflektion **rot** leuchten.

Dazu fügen wir in Zeile 17 eine `else if` Abfrage ein.



www.bob3.org

4. Unterrichtseinheit

In der vierten Unterrichtseinheit bearbeiten die Schülerinnen und Schüler die **Lerneinheiten sechs** und **sieben** des **„Intro II“**-Tutorials. Zunächst arbeiten sie mit dem im Mikrocontroller integrierten **Temperatur-Sensor** des Roboters. In diesem Zusammenhang lernen die SuS das Konzept und die Bedeutung einer **globalen Variablen** kennen und anwenden. Im nächsten Teil beschäftigen sie sich insbesondere mit der **Software-Bibliothek** des BOB3. Sie lernen **neue Methoden** kennen und vertiefen ihr Wissen über **Parameter**.

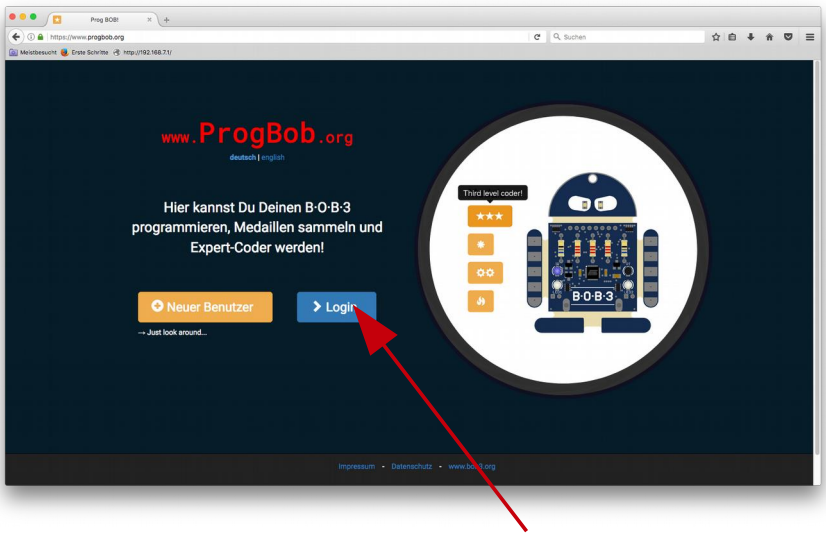
Zeitbedarf: ca. 45 min

Vorkenntnisse: BOB3 Modul 1

Material: Papier, Stift, Arbeitsblatt 7
PC oder Laptop, BOB3 mit Helm

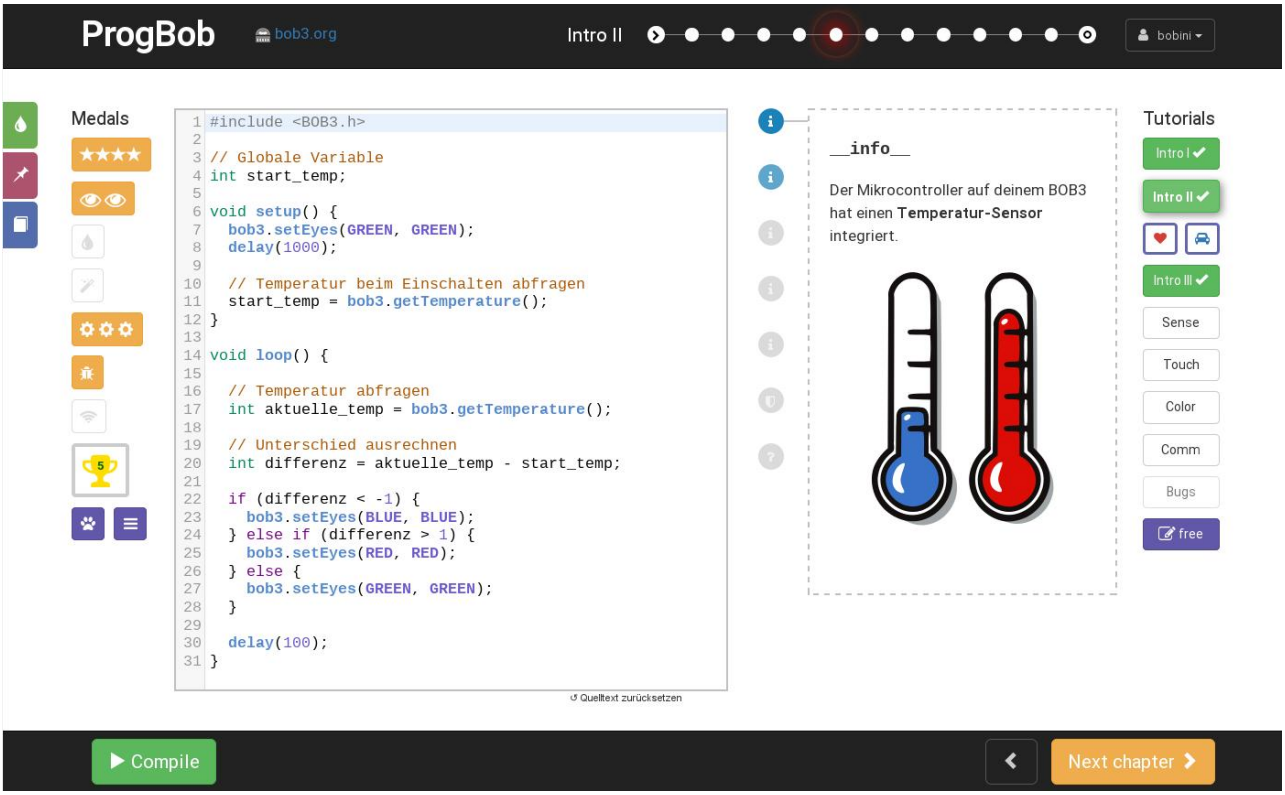
Ablauf

Die SuS starten den Webbrowser, gehen auf die Seite **http://www.ProgBob.org**, loggen sich mit Ihrem Account ein und gehen zum **„Intro-II“-Tutorial**:



„Login“

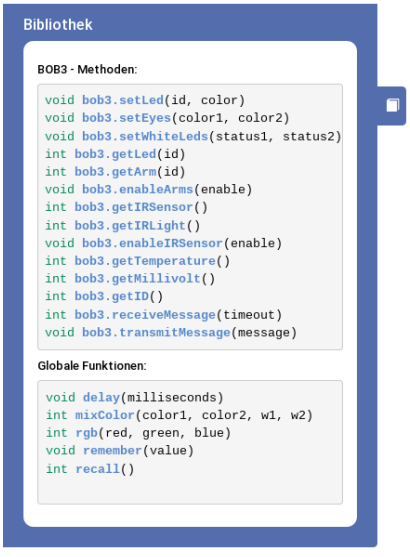
Zunächst probieren die Schüler das bestehende Programm-Beispiel aus: Der Mikrocontroller **misst beim Einschalten die Temperatur**, dann sind die Augen grün. Wenn man den Controller dann z.B. mit einem Finger **erwärmt**, dann steigt die Temperatur und die Augen werden **rot**. Wenn man den BOB3 dann z.B. in den Kühlschrank legt, dann wird ihm **kalt** und die Augen werden **blau**.

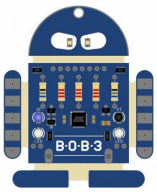


Weiterhin erlernen die Schüler das Konzept einer **globalen Variablen** und experimentieren mit verschiedenen Variablen in dem Temperatur-Programm.

Die Schüler bearbeiten das **Arbeitsblatt 7**, in dem die **Software-Bibliothek** des BOB3 erklärt wird.

„Bob's Software"





www.bob3.org

Das Arbeitsblatt 7 erklärt insbesondere, wie man für einzelne **Methoden** und **Funktionen** gültige **Parameter** herausfinden und anwenden kann:

Die Schüler vertiefen ihr Wissen über Methoden mit **Parametern** und wenden dieses in einem eigenen Programm an.

Lerninhalte des „**Intro II**“-Tutorials:

- SuS lernen Bedingungen wie 'gleich', 'ungleich' oder 'kleiner als' im Kontext einer Programmiersprache kennen
- SuS setzen die Bedingungen mit dem 'if/else' Konstrukt der Programmiersprache ein und experimentieren damit
- SuS lernen das Konzept einer Ganzzahl-Variablen kennen
- SuS speichern den Wert eines Sensors in einer Variablen und werten diese mit dem 'if/else' Konstrukt aus
- SuS lernen den IR-Reflex-Sensor und den Temperatur-Sensor kennen
- SuS erlernen den Sinn und Zweck einer 'for'-Schleife zur wiederholten Durchführung
- SuS lernen wie die Parameter der 'for'-Schleife eingesetzt werden können
- SuS lernen wie man Fehler im Programm-Code lokalisieren kann und wie man diese behebt

5. + 6. Unterrichtseinheit

In der fünften und sechsten Unterrichtseinheit beenden die Schüler das „**Intro II**“ Tutorial. Sie beschäftigen sich in **fünf Tutorial-Einheiten** mit dem Konzept der **for-Schleife** als Zähl-Schleife. Es werden verschiedene Programme als Beispiele angeboten, die von den SuS zunächst ausprobiert und anschließend experimentell verändert und erweitert werden. Zusätzlich bearbeiten die SuS das **Arbeitsblatt 8 „for-Schleife“**. Hierbei vertiefen sie ihr experimentell erworbenes Wissen zur Theorie der **for-Schleife** als Kontrollstruktur.

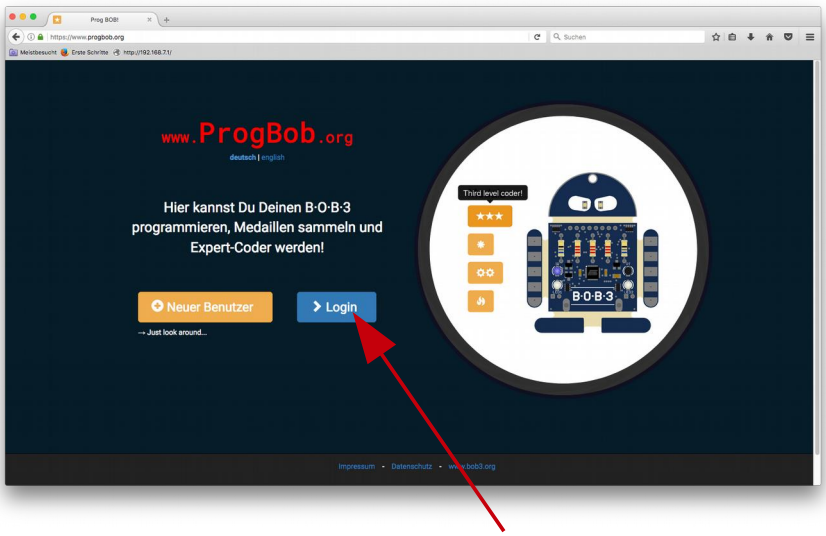
Zeitbedarf: ca. 90 min

Vorkenntnisse: BOB3 Modul 1

Material: Papier, Stift, Arbeitsblatt 8,
PC oder Laptop, BOB3 mit Helm

Ablauf

Die SuS starten den Webbrowser, gehen auf die Seite **http://www.ProgBob.org**, loggen sich mit Ihrem Account ein und gehen zum **„Intro-II“-Tutorial**:

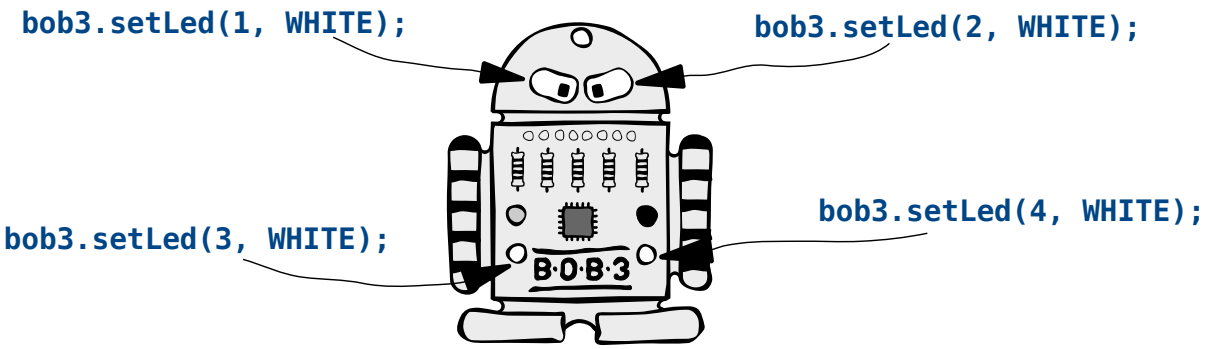


„Login“

Die Schüler lernen das **Konzept einer for-Schleife** zur wiederholten Durchführung an einem einfachen Programmierbeispiel kennen:

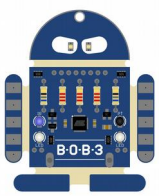
```
#include <BOB3.h>

void setup() {

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

}

void loop() {

}
```

Das Programm schaltet unter Verwendung einer for-Schleife nacheinander alle vier LEDs am BOB3 weiß ein:



Die Schüler lernen, welche Bestandteile zur **Parameterliste** einer for-Schleife gehören und wie diese verwendet werden. Sie lernen, wie eine **Laufvariable** verwendet wird und wenden ihr zuvor erworbenes Wissen über **Bedingungsprüfungen** erneut vertieft an. In diesem Zusammenhang lernen sie Ausdrücke/Operatoren wie z.B. `ledNummer++` kennen und anwenden.

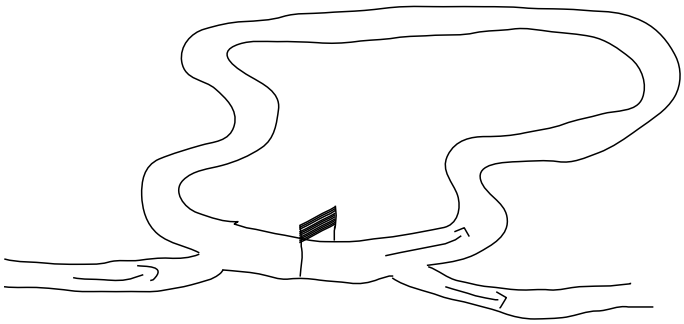
Die Schüler **experimentieren** mit dem Programm und beobachten am BOB3, welche Quellcode-Änderungen bestimmte Aktionen hervorrufen. In der nächsten Tutorial-Einheit wird das Programm-Beispiel weiter verändert, wobei jetzt auch Ausdrücke wie z.B. `ledNummer--` verwendet werden. Die Schüler erhalten vom Tutorial die Aufgabe, die Parameter der for-Schleife so abzuändern, dass nur die Augen weiß eingeschaltet werden. Hierbei zeigt sich, ob das Konzept der for-Schleife verstanden wurde.



www.bob3.org

Die Schüler bearbeiten das **Arbeitsblatt 8**:

„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	→	Ende eines Durchlaufs: die Laufvariable wird aktualisiert

Die SuS setzen sich ausführlich mit der **Theorie** der for-Schleife auseinander. Sie vertiefen ihr experimentell erworbenes Wissen über die **Initialisierung** der Laufvariablen, der **Bedingungsprüfung** und der **Aktualisierung**. Das Arbeitsblatt bespricht die Vorgänge anhand eines konkreten Programmier-Beispiels. Dabei wird der **erste Schleifendurchlauf** Schritt für Schritt analysiert und im Detail erklärt. Anschließend wird erklärt, was in den folgenden Schleifendurchläufen passiert und wann und warum die Schleife schließlich abgebrochen wird.

Die Schüler bearbeiten die **Aufgaben 1 bis 6** des Arbeitsblatts. Dabei setzen sie sich unter anderem noch einmal mit den Laufvariablen auseinander:

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

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

Durchlauf 1: i=_____

Durchlauf 2: i=_____

Durchlauf 3: i=_____

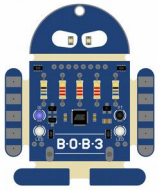
Durchlauf 4: i=_____

Durchlauf 5: i=_____

In der **elften Tutorial-Einheit** programmieren die Schüler ein neues Programm, das ebenfalls eine for-Schleife verwendet. Das Programm lässt die LEDs am BOB3 nacheinander weiß blinken:

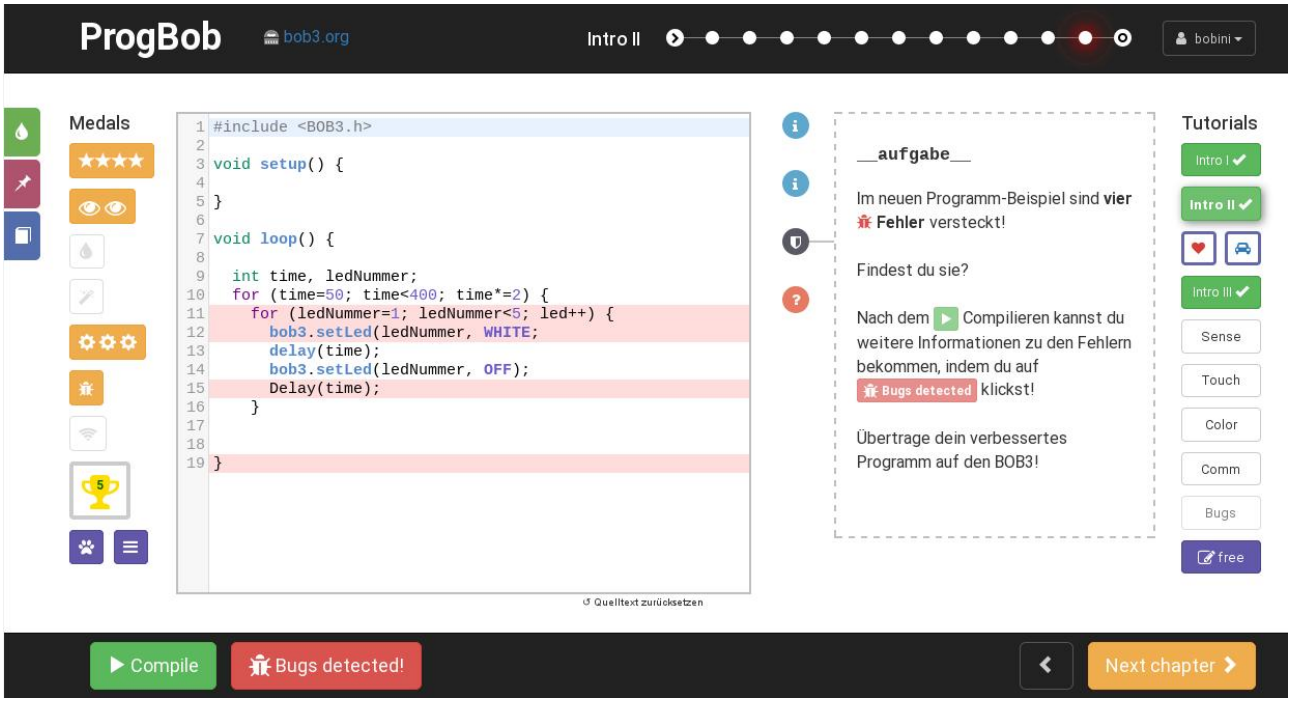
```
#include <BOB3.h>

void setup() {

}

void loop() {

  int ledNummer;
  for (ledNummer=1; ledNummer<5; ledNummer++){
    bob3.setLed(ledNummer, WHITE);
    delay(350);
    bob3.setLed(ledNummer, OFF);
    delay(150);

  }
}

```

Die Schüler testen das Programm am BOB3 und besprechen, was das Programm macht. Anschließend bekommen sie die Aufgabe, den Programmcode so zu verändern, dass keine for-Schleife verwendet wird, am Roboter jedoch dasselbe passiert wie vorher. Die Schüler sollen so möglichst den Sinn und Zweck von for-Schleifen verstehen.



www.bob3.org

Zum Abschluss bearbeiten die Schüler noch die **zwölfte und letzte Tutorial-Einheit** des „**Intro II**“-Tutorials. Es wird ein Programm mit zwei **verschachtelten** for-Schleifen präsentiert, in das Fehler eingebaut wurden. Die Schüler erlernen, **Fehler im Programmcode** zu entdecken und zu beheben:



Geschafft!! Die Schüler haben nun auch das „**Intro II**“-Tutorial erfolgreich beendet!!