# Your EdVenture into Robotics

## *10-Lektionen-Plan*

### *Arbeitsblätter*

Diese Unterrichtspläne wurden in Zusammenarbeit mit RoboticsWPS erstellt.

Übersetzt wurde er durch das Institut für Weiterbildung und Medienbildung, PHBern

# Inhalt

Name: _____

# EINFÜHRUNG

Edison ist dein neuer Roboter-Kollege und er bringt dir auf eine lustige und spielerische Art die Elektronik und das Programmieren bei.

Er ist mit den notwendigen Sensoren, Lampen und Motoren ausgerüstet, um dich in die faszinierende Welt der Robotik einzuführen.

*Das tönt ja gut, aber was ist Robotik?* Nun, diese Frage hat keine einfache Antwort. Der Erfinder von Edison, Brenton O'Brien sagt

**"*Ein Roboter ist eine Maschine, die selbständig etwas tun kann.*"**

Edison, der LEGO-kompatible Roboter

Das heisst, ein Roboter kann überlegen, selber Entscheidungen fällen und diese dann ausführen. Andere Leute würden das anders erklären, aber wir finden diese Beschreibung gut, weil sie einfach ist und zu dem passt, was du hier gleich lernen wirst.

Robotik wäre nicht möglich ohne Elektronik. Beim Edison kann man die durch die transparente Oberseite sehen was es da alles gibt: Widerstände, Kondensatoren, Transistoren, Motoren und andere Bauteile. Das wichtigste ist aber der sogenannte "**Mikrocontroller**".

Der Mikrocontroller des Edison

Der Mikrocontroller ist so etwas wie das Gehirn des Edison. Dort läuft sein *"Denken"* ab. Der Mikrocontroller des Edison gleicht dem Prozessor in einem Computer, er ist nur viel kleiner. Und wie beim Prozessor in einem Computer gibt es beim Mikrocontroller des Edison auch Programme. Die Programme lassen Edison Entscheide fällen und selber *"denken"*.

Edison enthält einige fest installierte Programme, die aufgerufen werden können, indem man ihn über einen speziellen Strichcode fahren lässt. Hier ist ein Beispiel:

Dieser Strichcode aktiviert das "An der Grenze abprallen"-Programm des Edison

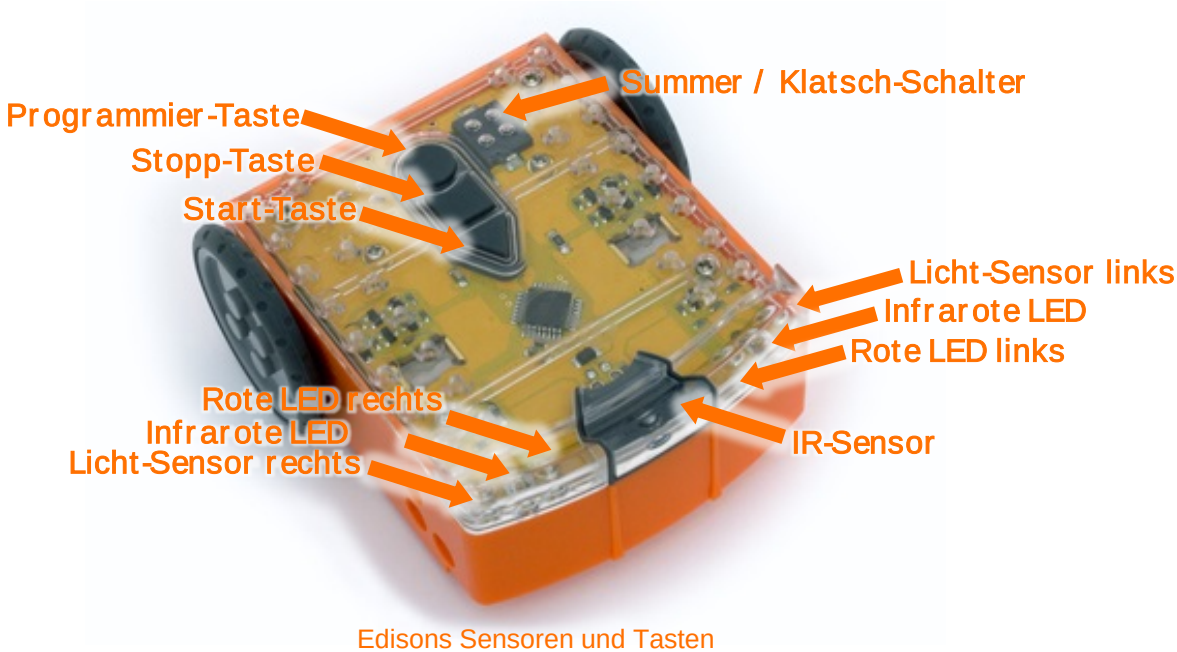
Das faszinierendste bei Edison ist, dass man ihm eigene Programme beibringen kann, und **DU** wirst gleich lernen, wie man das macht! Du kannst Edison vorgeben, wir er denken, sich verhalten und auf seine Umgebung reagieren soll. Programmieren ist überraschend einfach. Das obige Beispiel lässt Edison eine Linie verfolgen.

Name: _____

# Arbeitsblatt 1.1: Edison kennen lernen

Edison ist ein kleiner, LEGO-kompatibler und programmierbarer Roboter.

Um Edison zu benutzen musst du wissen wo alle seine Sensoren sind und was die drei Tasten bewirken. Auf den untenstehenden Bildern sind sie beschriftet. Vielleicht musst du sie später nochmals anschauen, wenn wir die einzelnen Abenteuer durchgehen.




Edisons Sensoren und Tasten

Start-Taste: Das Programm starten
Stopp-Taste: Das Programm beenden
Programmier-Taste: 1x drücken = Programm vom Computer laden
3x drücken = Strichcode lesen


Hauptschalter und Linien-Sensor

Der Linien-Sensor des Edison besteht aus zwei Teilen, nämlich einem roten LED-Licht und einem Helligkeits-Sensor (Fototransistor). Das rote Licht scheint auf die Unterlage. Falls diese weiss ist und das Licht reflektiert, dann bekommt der Helligkeits-Sensor ein starkes Signal. Wenn die Unterlage schwarz ist und kein Licht reflektiert, dann misst der Sensor kein Signal.

Über das Kabel kannst du Programme auf den Edison laden. Das Kabel wird beim Kopfhörer-Ausgang des Computers oder Pads eingesteckt.

Programmierkabel

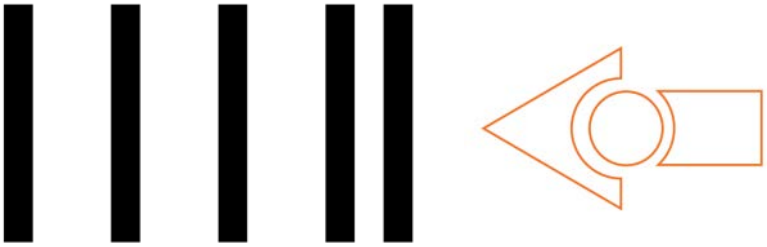
Name: _____

# Arbeitsblatt 1.2: Strichcode-Programmierung

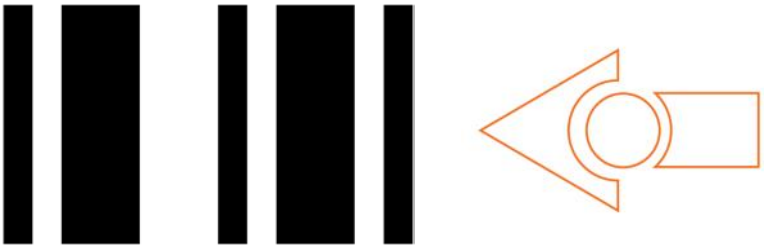
So liest du den Strichcode:

1. Stelle Edison rechts vom Strichcode hin, so dass er nach links schaut
2. Drücke die runde Programmier-Taste 3x
3. Edison fährt ein Stück vorwärts und liest den Strichcode ein

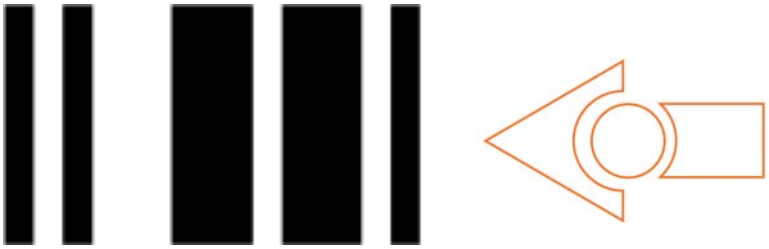
**Beschreibe die Bewegungen, die der Roboter macht.**



Strichcode – Fahrt mit Händeklatschen steuern



Strichcode – Hindernisse umfahren



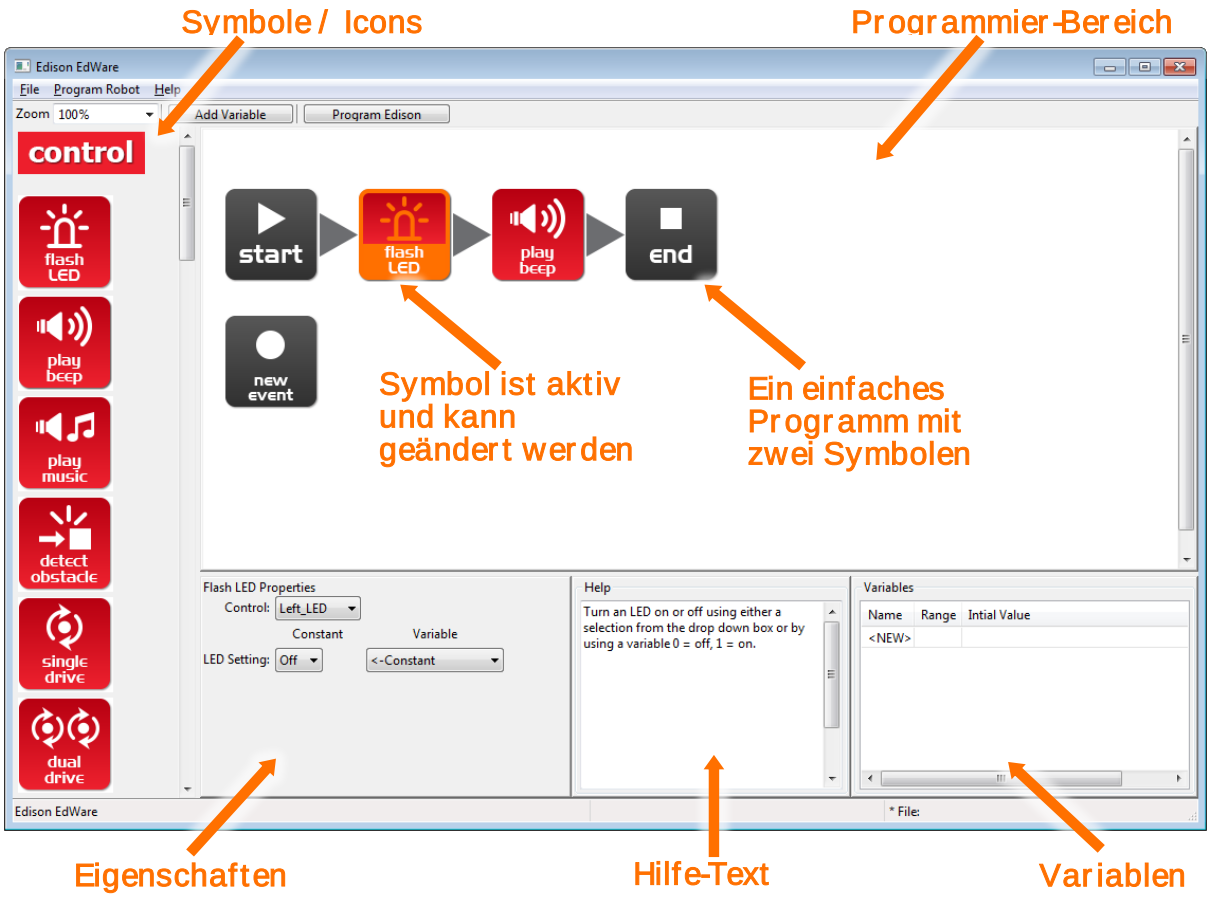
Strichcode – An der Grenze abprallen

Strichcode – Einer Lampe folgen

Name: _____

# Arbeitsblatt 1.3: EdWare kennen lernen

So sieht die Software “EdWare” auf einem Windows-Computer aus. Auf anderen Computern schaut es ganz ähnlich aus.




Um ein Programm zu entwickeln, kannst du die Symbole (Icons) von der linken Seite in den Programm-Bereich nach rechts ziehen und zwar auf die Pfeile zwischen den Symbolen "Beginn" (Start) und "Ende".

Wähle ein Icon aus und passe die Einstellungen bei "Eigenschaften" an, um festzulegen, wie sich Edison bei diesem Icon verhalten soll.

Der Hilfe-Text gibt Tipps zum Programmieren. Alles, was du über ein Icon wissen musst, findest du dort.

Im Bereich "Variablen" kann man kleine Bereiche im Speicher vom Edison verwalten und anschauen. Mehr dazu später!

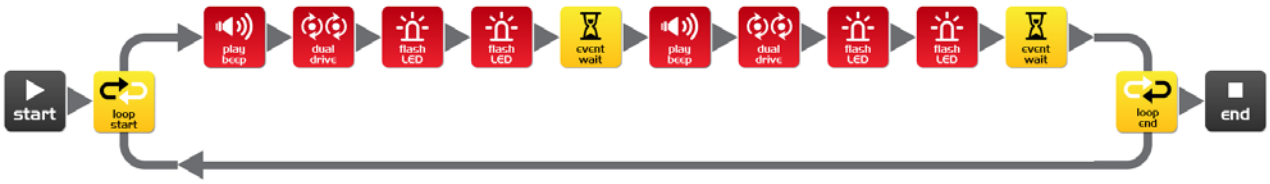
Es gibt vier Arten von Icons. Wie heissen sie?

| | |
|---|---|
| | |

Name: _____

# Arbeitsblatt 1.4: Test-Programm

Wenn du EdWare auf deinem Computer oder Tablet installiert hast, öffne die Datei TestProgram.edw (Datei > öffnen: EdWare/Meine Programme). Das Programm sieht etwa so aus:

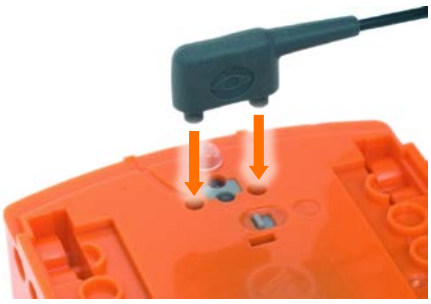



Test program

Alle Programme werden vom Edison immer genau gleich „gelesen“:

- Die Pfeile zeigen die Richtung, in der Edison die Icons liest.
- Edison schaut der Reihe nach auf jedes Icon und tut, was das Icon sagt.
- In diesem Programm gibt es eine Endlos-Schleife, so dass das Programm immer weiterläuft, bis du den Edison abschaltest oder die Batterien leer sind.

Um das Programm auf den Edison übertragen zu können, stecke das Programmierkabel in den Kopfhörer-Anschluss deines Computers und stelle die Lautstärke auf den Maximalwert. Das andere Ende des Programmierkabels steckst du wie abgebildet in den Edison.



Um nun das Testprogramm auf den Edison zu übertragen, führe die folgenden Schritte aus:

1. Drücke bei Edison die runde Programmier-Taste 1x (Edison vorbereiten)
2. Wähle in EdWare "zu Edison übertragen" und danach "Übertrage Programm..."
3. Drücke bei Edison die dreieckige Start-Taste um das Programm zu starten

Was tut dein Edison, wenn du „Play“ drückst?

_____

_____

Kannst du diese Aktionen auf die Icons im Programm beziehen? Beschreibe.

_____

_____

Wie gelangt das Programm vom Computer auf deinen Edison?

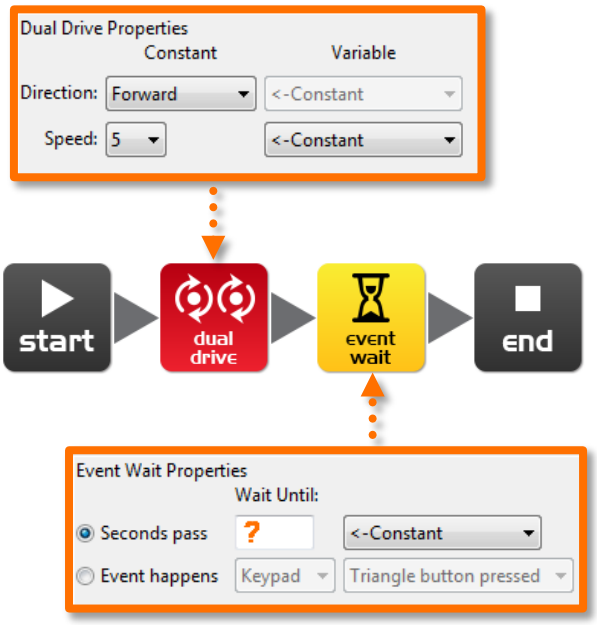
_____

_____

Name: _____

# Arbeitsblatt 2.1: Vorwärts fahren

Schreibe das folgende Programm, um den Edison-Roboter vorwärts fahren zu lassen. Benutze das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass Edison vor der Start-Linie anfährt und vor der Ziel-Linie gleich wieder stoppt (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis du deinen Edison genau auf den Punkt zum Stoppen bringst.

Welches ist die richtige Zeit, um deinen Edison vom Start bis zum Ziel zu bekommen?

_____

Beschreibe, was dein Edison tut und warum.

_____

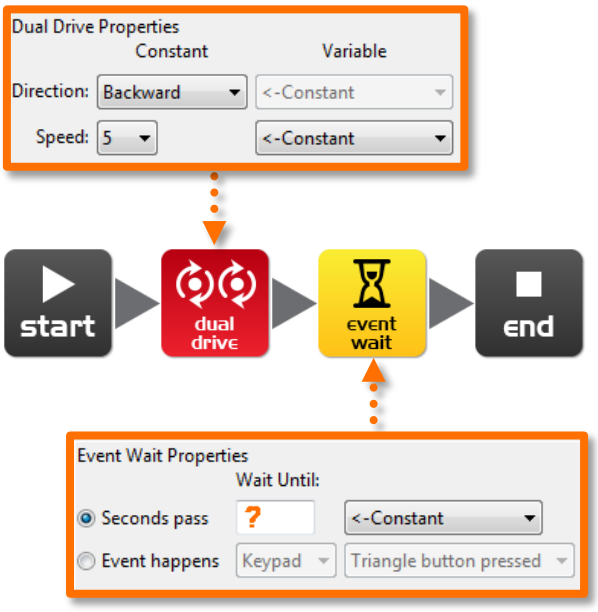
_____

_____

Name: _____

# Arbeitsblatt 2.2: Rückwärts fahren

Schreibe das folgende Programm, um den Edison-Roboter rückwärtsfahren zu lassen. Benutze das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass Edison vor der Start-Linie anfährt und vor der Ziel-Linie gleich wieder stoppt (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis du deinen Edison genau auf den Punkt zum Stoppen bringst.

Welches ist die richtige Zeit, um deinen Edison vom Start bis zum Ziel zu bekommen?

_____

Beschreibe, was dein Edison tut und warum.

_____

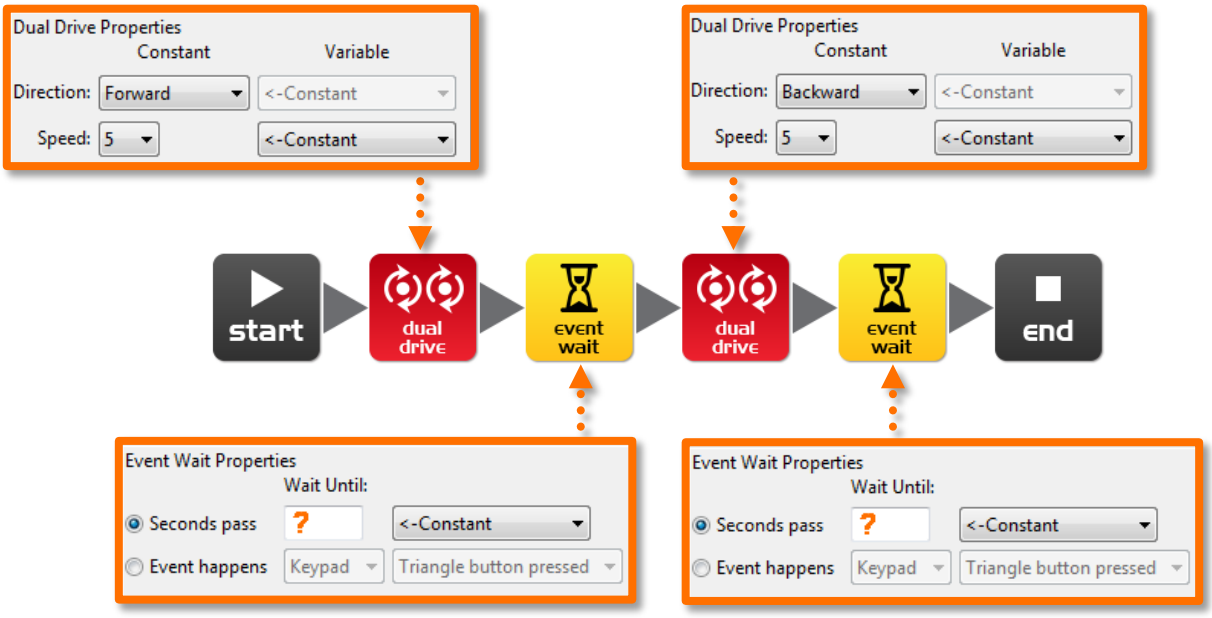
_____

_____

Name: _____

# Arbeitsblatt 2.3: Vorwärts- & Rückwärtsfahren

Schreibe das folgende Programm, um den Edison-Roboter vorwärts und rückwärts fahren zu lassen. Benutze das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass Edison vor der Start-Linie anfährt und vor der Ziel-Linie gleich wieder stoppt um dann rückwärts zu fahren und nach der Start-Linie erneut zu stoppen (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis du deinen Edison genau auf den Punkt zum Stoppen bringst.

Welches sind die richtigen Zeitangaben, um deinen Edison vorwärts und rückwärts fahren zu lassen?

Zeit vorwärts _____ Zeit rückwärts _____

Beschreibe, was dein Edison tut und warum.

_____

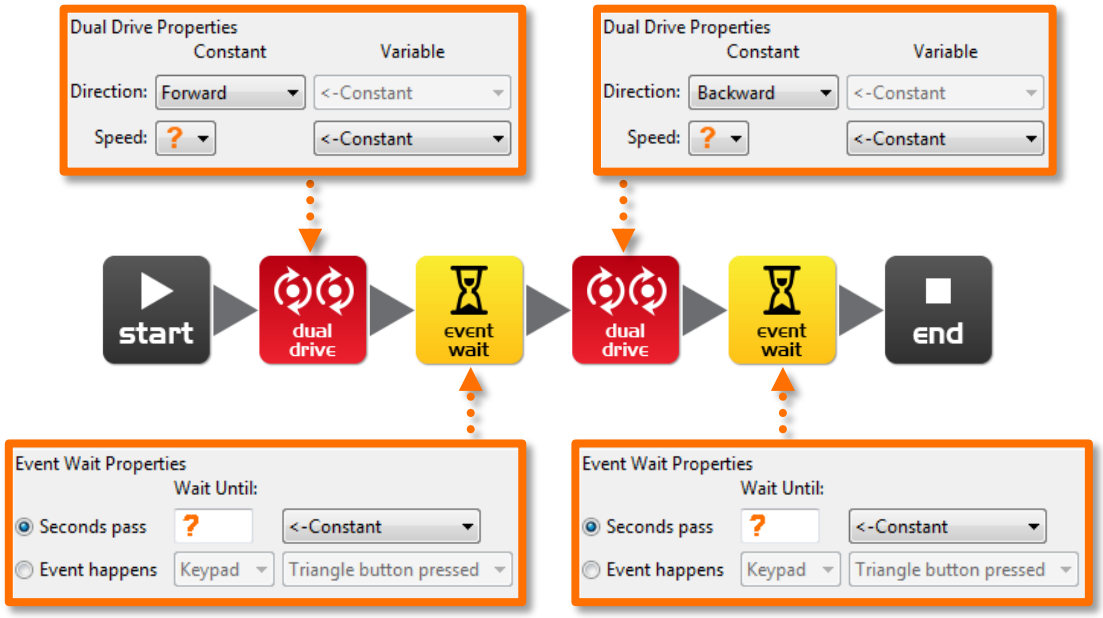
_____

_____

Name: _____

# Arbeitsblatt 2.4: Geschwindigkeit anpassen

Schreibe das folgende Programm, um den Edison-Roboter vorwärts und rückwärts fahren zu lassen. Benutze das Arbeitsblatt „Activity 2.1“ oder ein farbiges Band auf einem Schreibtisch.




Dieses Mal passt du die Zeit und die Geschwindigkeit an.

1. Starte hinter der Start-Linie, fahre vorwärts und stoppe vor der Ziel-Linie um dann so schnell wie möglich rückwärts zu fahren und vor der Start-Linie zu stoppen.

Was sind deine ***schnellsten*** Einstellungen?

Geschwindigkeit vorwärts _____ Zeit vorwärts _____

Geschwindigkeit rückwärts _____ Zeit rückwärts _____

2. Starte hinter der Start-Linie, fahre vorwärts und stoppe vor der Ziel-Linie um dann so langsam wie möglich rückwärts zu fahren und vor der Start-Linie zu stoppen.

Was sind deine ***langsamsten*** Einstellungen?

Geschwindigkeit vorwärts _____ Zeit vorwärts _____

Geschwindigkeit rückwärts _____ Zeit rückwärts _____

Du kannst ebenfalls weitere Icons hinzufügen (LED, Piepton).

Name: _____

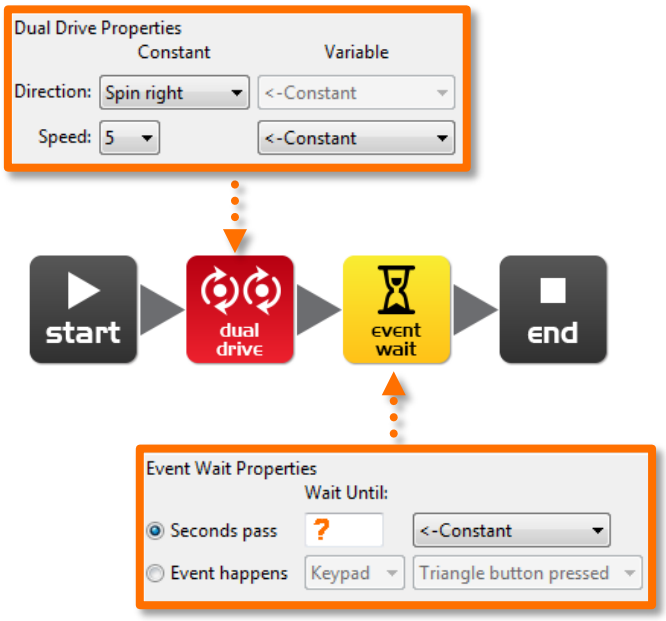
# Arbeitsblatt „Activity 2.1“


START-LINIE

Name: _____

# Arbeitsblatt 3.1: Rechtsdrehung 90°

Schreibe das folgende Programm, um den Edison um 90 Grad (90°) zu drehen. Verwende dazu das Arbeitsblatt „Activity 3.1“.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass sich der Edison genau 90° nach rechts dreht (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis dein Edison sich genau um 90° nach rechts dreht.

Welches ist die richtige Zeit, damit sich dein Edison um 90° nach rechts dreht?

90° nach rechts _____ 180° nach rechts _____

Beschreibe, was dein Edison tut und warum.

_____

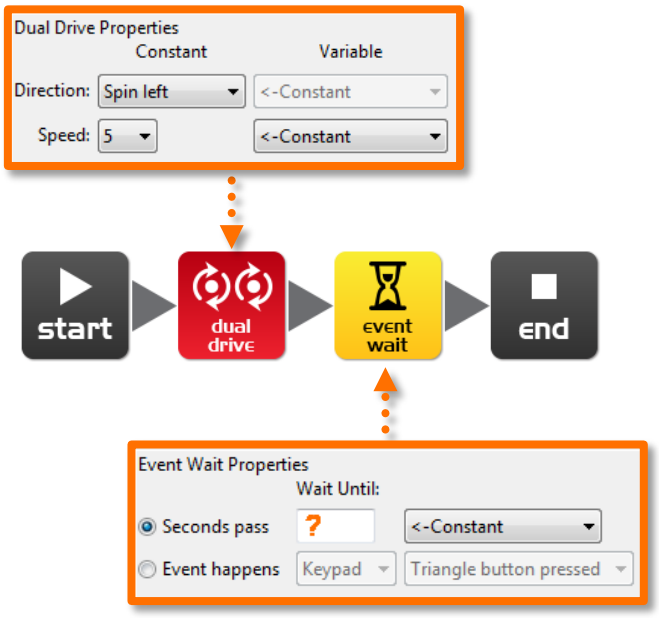
_____

_____

Name: _____

# Arbeitsblatt 3.2: Linksdrehung 90°

Schreibe das folgende Programm, um den Edison um 90 Grad (90°) zu drehen. Verwende dazu das Arbeitsblatt „Activity 3.1“.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass sich der Edison genau 180° nach links dreht (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis dein Edison sich genau um 90° nach links dreht.

Welches ist die richtige Zeit, damit sich dein Edison um 180° nach links dreht?

90° nach links _____ 180° nach links _____

Beschreibe, was dein Edison tut und warum.

_____

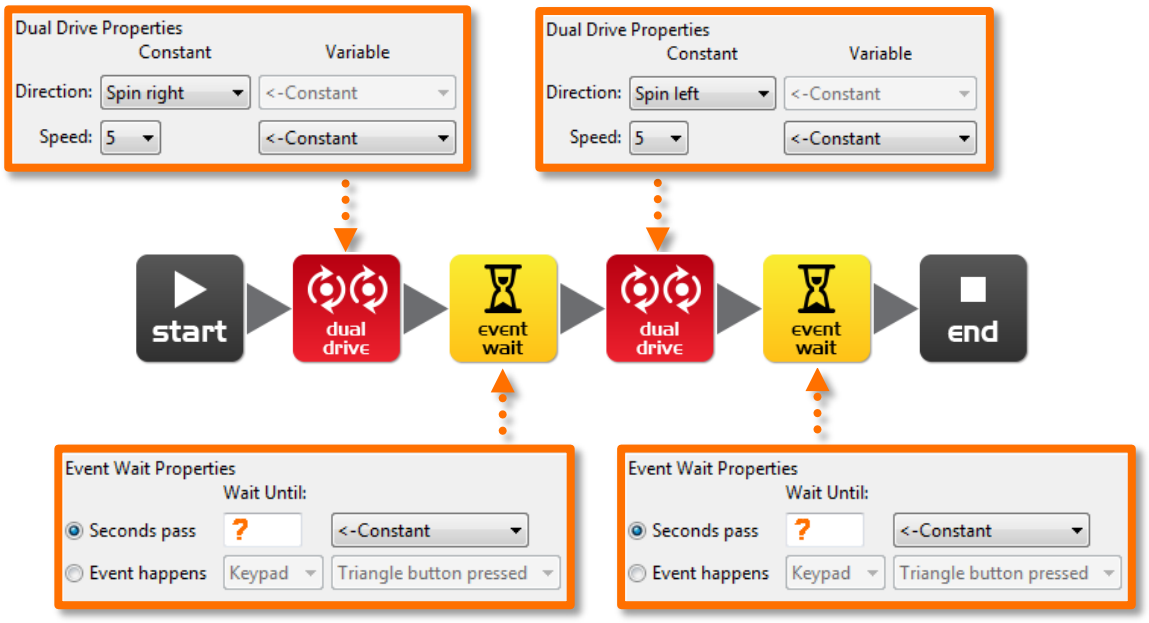
_____

_____

Name: _____

# Arbeitsblatt 3.3: Rechts- & Linksdrehung

Schreibe das folgende Programm, damit sich dein Edison 09° nach rechts dreht und anschliessend 270° nach links. Verwende dazu das Arbeitsblatt „Activity 3.1“.




Setze bei den Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Zeit so, dass sich der Edison erst genau 90° nach rechts und dann 270° nach links dreht (**?** ersetzen).

Die Mindestzeit beträgt 0.01 Sekunden

Die Maximalzeit beträgt 327.67 Sekunden

Versuche unterschiedliche Zeitangaben, bis dein Edison sich genau 90° nach rechts und dann 270° nach links dreht.

Welches ist die richtige Zeit, damit sich dein Edison um 90°nach rechts und dann 270° nach links dreht?

90° nach rechts _____ 270° nach rechts _____

Beschreibe, was dein Edison tut und warum.

_____

_____

_____

Name: _____

# Arbeitsblatt 3.4: Mini-Labyrinth

Benutze dein erworbenes Roboter-Programmier-Wissen, um deinen Edison durch das Mini-Labyrinth auf dem Arbeitsblatt „Activity 3.2“ zu fahren.

Dein Edison muss von der Start-Linie aus starten und nach der Ziel-Linie stehen bleiben. Er darf nicht über die Linie, aus dem Labyrinth fahren

Dazu musst du mehrere Ablauf-Icons kombinieren: Vorwärtsfahren und Drehungen um die gewünschten Wendungen zu machen.

Beschreibe, was dein Edison tut und warum.

_____

_____

_____

Auf welche Herausforderungen bist du gestossen, um deinen Edison durch das Labyrinth zu fahren/steuern?

_____

_____

## Weiterführende Aufgabe

Wer kann seinen am schnellsten durch das Labyrinth fahren?

Es besteht keine Notwendigkeit, eine Stoppuhr zu benutzen. Zähle einfach bei allen Icons-Eigenschaften für „Ereignis warten / Event wait“ bei „ Seconds pass / Zeitablauf“ die Sekunden zusammen.

Denke daran: Dein Edison muss von hinter der Start-Linie starten und nach der Ziel-Linie stoppen. Er darf auch nicht auf den Linien fahren, um zu gewinnen. Er muss innerhalb der Grenze fahren.

Wer hat die schnellste Zeit durch das Labyrinth?

Wie schnell ist dein Edison?

_____

Name: _____

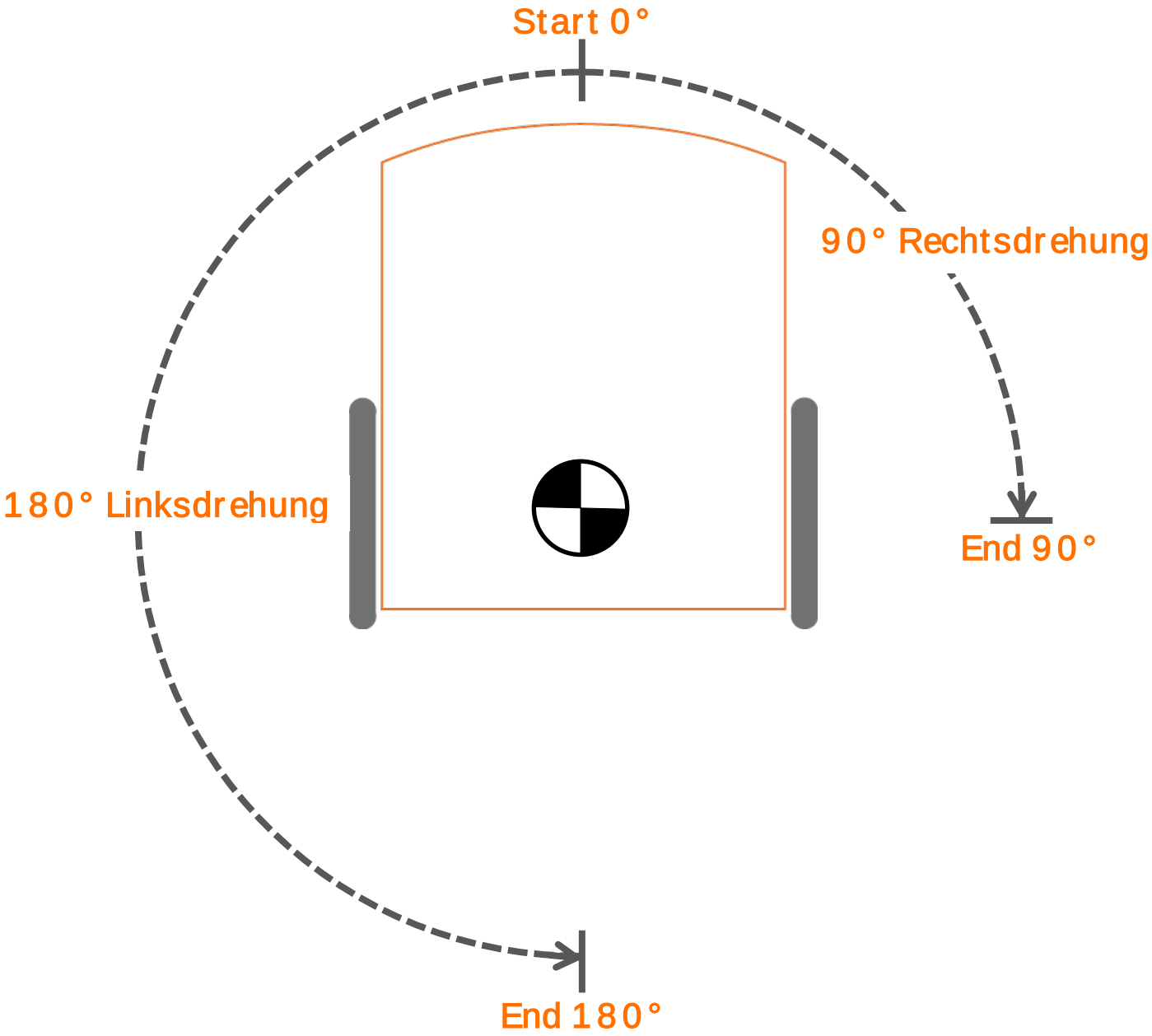
# Arbeitsblatt „Activity 3.1“

Platziere deinen Edison auf dem Umriss. So kannst du kontrollieren, ob er sich richtig dreht.

Aufgabe 1 – Drehung nach rechts von Start 0° bis End 90° (Drehung um 90°)

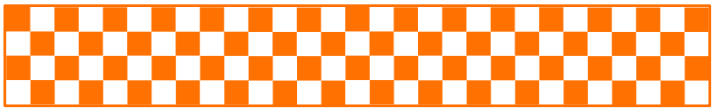
Aufgabe 2 – Drehung nach links von Start 0° bis End 180° (Drehung um 180°)

Aufgabe 3 – Drehung nach rechts von Start 0° bis End 90° (Drehung um 90°) dann Drehung nach links von End 90° bis 180° (Drehung um 270°)



Name: _____

# Arbeitsblatt „Activity 3.2“



FINISH-LINE

START-LINE

Name: _____

# Arbeitsblatt 4.1: Fahr-Challenge

Wähle eine eigene Challenge für dich und deinen Edison. Programmiere ihn so, dass er die Aufgabe lösen kann.

Hier sind einige Beispiele. Du kannst aber deine eigene Challenge kreieren:

- Dein Edison soll um ein Hindernis herum fahren.
- Dein Edison soll am Rand eines Schreibtisches entlang fahren.
- Dein Edison fährt durch ein von dir gezeichnetes Labyrinth (Flipchart, Packpapier, etc.).
- Dein Edison fährt durch ein von dir gebautes Labyrinth
- Eigene Challenge: _____
  _____
  _____

Denke daran, dass du auch weitere Icons (LED, Piepton, etc.) verwenden sollst.

## Welche Challenge/Problem hast du gewählt?

_____

_____

_____

## Welche Herausforderung hattest du beim Schreiben des Programms?

_____

_____

_____

## Welche zusätzlichen Icons hast du in deinem Programm aufgenommen und was haben sie gemacht?

_____

_____

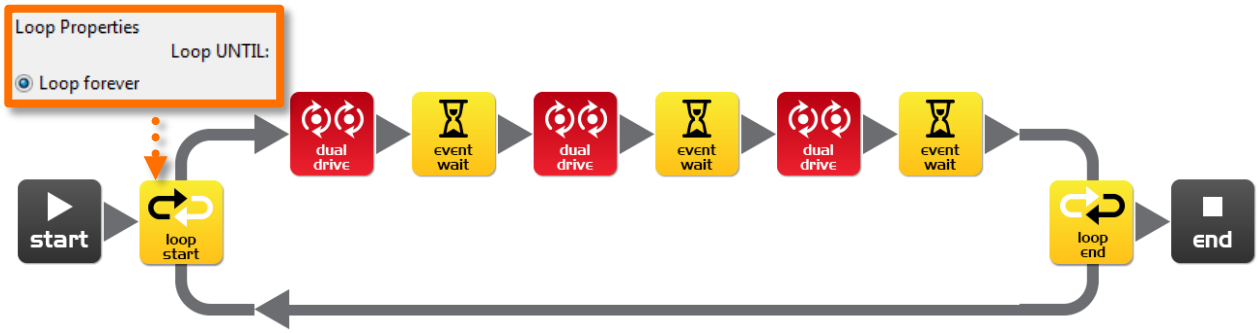
_____

Name: _____

# Arbeitsblatt 4.2: La-Ola-Welle

Dies ist eine lustige Aufgabe, welche du mit der ganzen Klasse machen kannst: Jeder Edison macht das gleiche Programm, nur zeitverschoben. Das Ergebnis ähnelt einer La-Ola-Welle oder einer Choreographie. Loops

Nutze das Ablauf-Icon „Schleife“, so wiederholt sich die Bewegung fortlaufend.




## Aufgabe

Schreibe ein kurzes, cooles Programm mit einer Abfolge von Bewegungen. Vergiss nicht, auch „LED“ und „Piepten“ einzubauen.

Wenn ihr eure Programme geschrieben habt, stellt diese einander vor.

Nun wählt ihr euer „Lieblings-Klassenprogramm“ aus.

Zeigt euch, wie das Programm aussieht. Alle schreiben nun das gleiche Programm für ihren Edison, ABER am Anfang fügt ihr ein „Ereignis warten“ hinzu. Die Zeit, die ihr hier eintragen müsst, erhält ihr von eurer Lehrperson.

Zeit „Ereignis warten“: _____ Startplatz Edison _____

Stellt die Edison gemäss der Startreihenfolge auf einer Linie nebeneinander auf.

Auf Kommando drücken alle GLEICHZEITIG die Starttaste.

Schaut und beobachtet, wie sich nun die Roboter bewegen.

Beschreibe die Bewegung von deinem Edison:

_____

_____

_____

Was siehst du? Wie wirkt die Choreographie?

_____

_____

_____

Name: _____

# Projekt I: Mein erstes Programm

In Zukunft werden Roboter unsere Helfer sein und uns unterstützen können. Wir haben bereits Roboter Staubsauger und daher ist die Zukunft gar nicht mehr so weit weg. Welche hilfreiche oder unterhaltsame Programme könnt ihr selber so programmieren, dass dein Edison diese Aufgaben ausführt?

Beispiele sind:

- Tanz zur Musik – Dein Edison tanzt zu deiner Lieblingsmusik
- Staubsauger – Du zeichnest oder baust den Grundriss deiner Wohnung auf ein grosses Papier und programmierst deinen Edison so, dass er die ganze Fläche abfährt
- Sicherheitsroboter – Definiere einen sicheren Bereich auf Papier oder ein wertvolles Objekt. Dein Edison hat nun die Aufgabe, diesen sicheren Bereich zu schützen, indem er patrouilliert und um das Objekt herumfährt und immer wieder Pausen einlegt.

## 1. Ein Problem auswählen

Entscheide dich für eine Aufgabe, für eine Problemstellung. Diskutier oder Besprich es mit einer Kameradin, einem Kameraden oder deiner Lehrperson.

### Welche Ideen hast du?

_____

_____

_____

_____

_____

### Warum sind deine Ideen möglich, bzw. nicht möglich?

_____

_____

_____

_____

_____

Name: _____

## 2. Das Problem oder die Bewegungen beschreiben

Bevor du beginnst, die Problemstellung zu lösen, beschreibe das Problem, welches dein Edison lösen soll, in deinen eigenen Worten.

Das Problem ist …_____

_____

_____

_____

Als nächster Schritte versuche, eine mögliche Lösung zu beschreiben, wie dein Programm aussehen könnte, welches deinem Edison sagt, wie er das Problem lösen soll.

Mein Edison wird dies lösen, indem er …_____

_____

_____

_____

## 3. Das Programm schreiben und testen

Bevor du dein Programm in der Software schreibst, mache dir Gedanken und skizziere/beschreibe diese.

Nun schreibe dein Programm, indem du die Icons nutzt, die du bereits kennen gelernt hast.

Teste dein Programm mit deinem Edison.

Name: _____

# 4. Fehler?

Nicht immer ist der erste Versuch erfolgreich. Ein zentraler Punkt bei der Programmierung ist es, dass Fehler machen dazu gehört! Fehler sind ein normaler Teil der Programmierung (oder einer Ingenieurdisziplin). Thomas Edison ist 10.000 Mal gescheitert, bevor es ihm gelang, die Glühbirne zu erfinden.

Beschreibe deine gemachten Fehler und versuche sie zu korrigieren.

_____

_____

_____

_____

_____

_____

Name: _____

# 5. Beschreiben der verwendeten Programmiersymbole und was sie bewirken

Zeichne die Icons, welche du in deinem Programm verwendet hast.

Beschreibe, was das Icon bewirkt, bzw. was dein Edison macht, wenn er es liest.

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

_____

_____

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

_____

_____

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

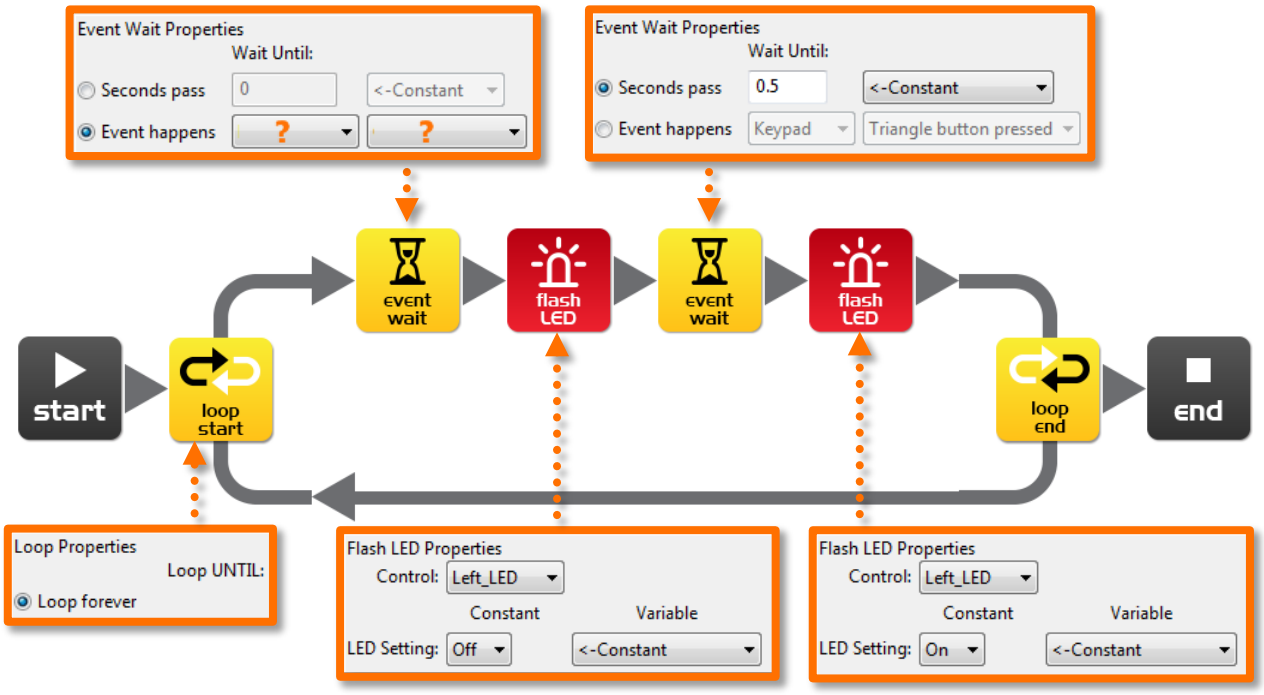
_____

_____

Name: _____

# Arbeitsblatt 6.1: LED reagiert auf Händeklatschen

Schreibe das folgende Programm, damit die linke LED von deinem Edison auf ein lautes Geräusch (Händeklatschen) reagiert.




In diesem Programm wird das erste „Ereignis warten / Event weit“ nicht als Zeit/Dauer von einer Bewegung gebraucht. Stattdessen wird das Programm an diesem Punkt nicht fortgesetzt, bis ein bestimmtes „Ereignis auftritt / Event Happens“.

Dazu musst du innerhalb des erwähnten ersten Icons „Ereignis warten / Event weit“ im Eigenschaftsfeld „Ereignis auftritt / Event Happens“ aktiviert werden. Nun kann das zu erwartende Ereignis, hier auf ein Klatschen zu reagieren, ausgewählt werden.

Bis zu welchem Abstand kann dein Edison dein Klatschen erfassen?

_____

Was bewirkt das Icon „Schleife / Loop“ in diesem Programm?
Und was würde passieren, wenn die Schleife / Loop nicht wäre?

_____

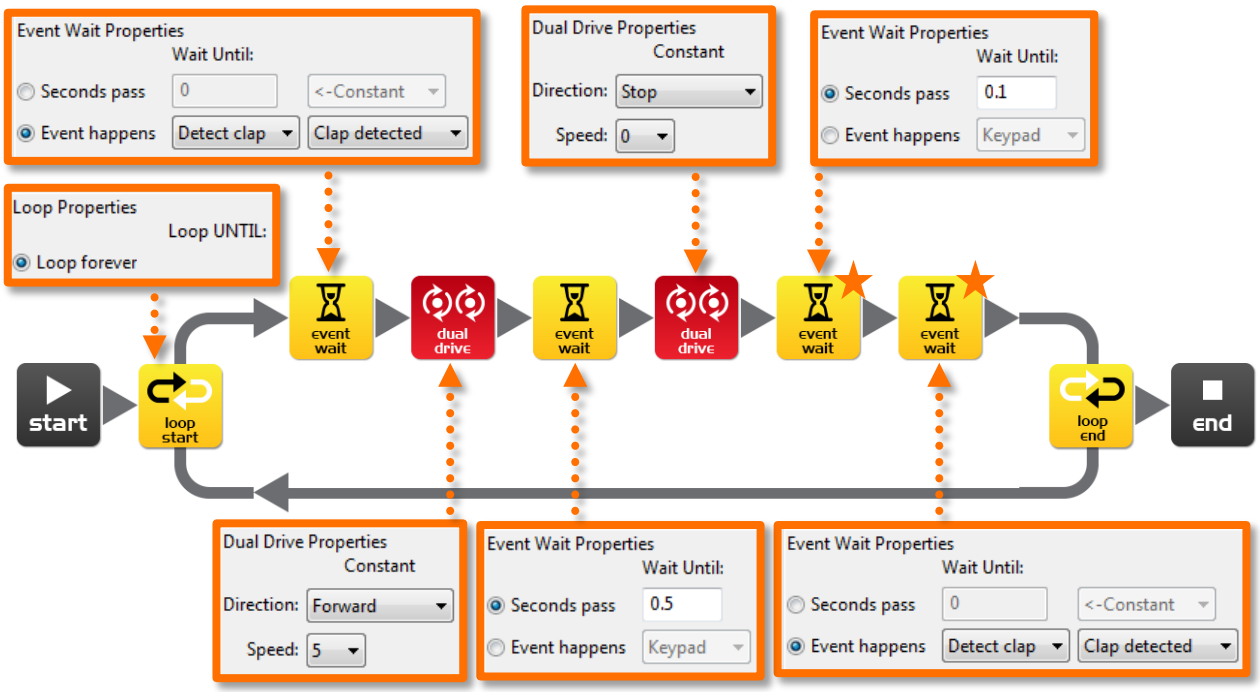
_____

_____

Name: _____

# Arbeitsblatt 6.2 – Mit Händeklatschen fahren

Schreibe das folgende Programm, damit dein Edison auf dein Händeklatschen reagiert und fährt.




Der Sound-Sensor (Summer / Klatsch-Schalter) von deinem Edison ist nicht nur empfindlich auf Händeklatschen, sondern er reagiert auch auf jedes laute Geräusch. Dazu gehören die Motoren, Zahnräder und Räder vom Edison selber, die auch beim Drehen klingen, Geräusche machen. Um zu verhindern, dass dein Edison den Sound-Sensor durch das eigene Fahren auslöst, gibt es zwei zusätzliche Symbole, die in diesem Programm verwendet werden. Sie sind oben mit orangen Sternen markiert. Das erste „Ereignis warten / Event weit“ ist auf 0,1 Sekunden eingestellt und gibt deinem Edison die nötige Zeit, die Motoren zu stoppen. Das zweite „Ereignis warten / Event weit“ ist so eingestellt, dass es auf ein Händeklatschen wartet. So wird es dem Programm ermöglicht fortzufahren, ohne dass es das eigene Motorengeräusch erkennt und darauf reagiert.

Du musst die beiden Icons „Ereignis warten / Event weit“ nach dem Stoppen der Motoren einfügen, wenn du den Sound-Sensor verwendest. Ansonsten funktioniert dein Programm nicht wie gewünscht.

Experimentiere mit verschiedenen Richtungen im ersten Icon „Antrieb / Dual Drive“ sowie dem nachfolgenden Ereignis warten / Event weit“, um verschiedene Bewegungen zu erstellen.

Welche anderen Richtungen und Zeitenhast du ausprobiert?

Experiment 1: Richtung _____ Ereignis warten / Zeit_____

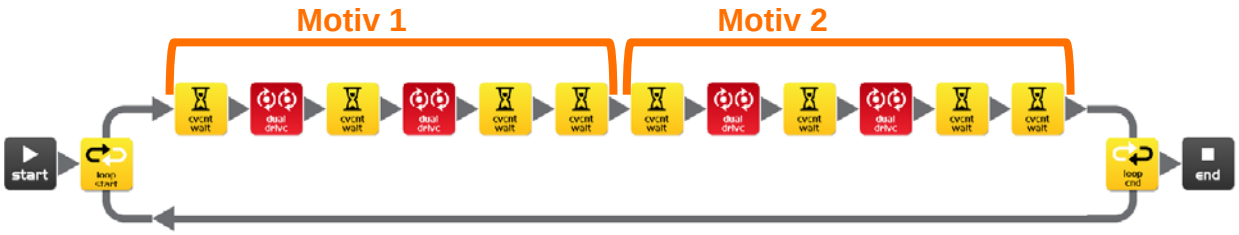
Experiment 2: Richtung _____ Ereignis warten / Zeit_____

Experiment 3: Richtung _____ Ereignis warten / Zeit_____

Name: _____

# Arbeitsblatt 6.3: Tanzen nach Händeklatschen

Mit dem, was du in den beiden vorangegangenen Übungen gelernt hast, kannst du eine Choreographie programmieren, bei der dein Edison auf dein Klatschen reagiert.




Du brauchst mindestens zwei Motive. Du kannst später so viele weitere Motive hinzufügen, wie du willst.

Das oben genannte Programm hat zwei einzelne Handlungsmotive. Diese beiden Motive wiederholen sich, da sie in einer Schleife sind.

| | |
|---|---|
| 1. Händeklatschen | Motiv 1 |
| 2. Händeklatschen | Motiv 2 |
| 3. Händeklatschen | Motiv 1 |

Du kannst auch versuchen, zwei Motive pro Händeklatschen hinzuzufügen.

## Alternative Aufgabe

Wenn du keine Choreographie bzw. Tanz programmieren möchtest, dann kannst du einen Labyrinth aufstellen, durch welches dein Edison als Reaktion auf dein Händeklatschen fährt und als Antwort auf ein zweites Händeklatschen zu dir zurückkehrt.

Wie viele Motive hat dein Programm

_____

Beschreibe die Choreographie / Tanz / Hinderniskurs welcher dein Edison macht:

_____

_____

_____

Name: _____

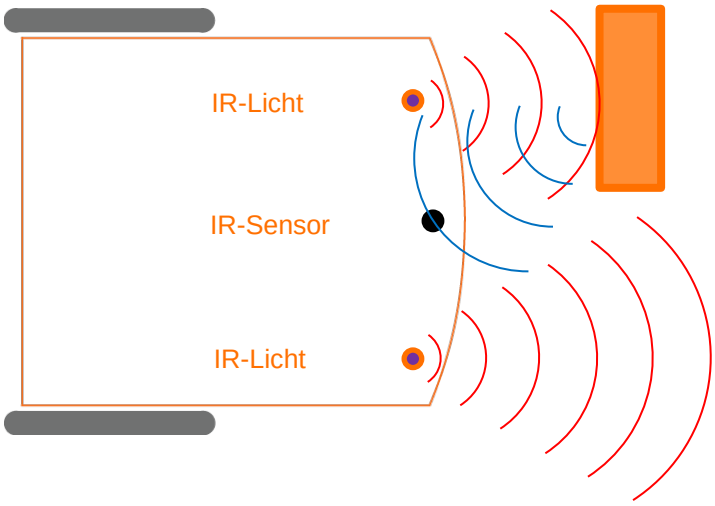
# Arbeitsblatt 7.1: Wie funktioniert die Infrarot-Hindernis-Erkennung?

Dein Edison ist mit Infrarot (IR) Sensoren ausgestattet. Infrarot-Licht ist für das menschliche Auge nicht sichtbar, also kannst du dieses Licht nicht sehen. Aber es erlaubt deinem Edison, im Dunkeln zu sehen.

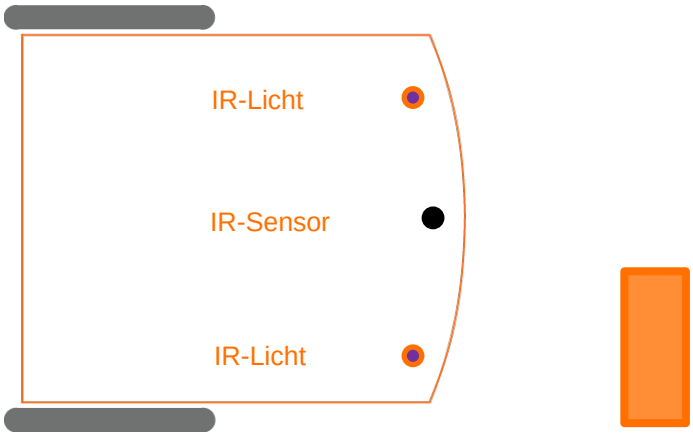
Dein Edison nutzt dieses IR-Licht um Hindernisse zu erkennen. Dazu wird IR-Licht von der linken und rechten Seite, von den Infrarot-LEDs deines Edisons nach aussen ausgesendet. Wenn das IR-Licht von einem Hindernis, wie zum Beispiel einer Wand, reflektiert wird, dann wird das reflektierte Licht vom IR-Sensor deines Edisons erkannt. Der IR-Sensor befindet sich vorne in der Mitte.

In der Abbildung unten steht ein Hindernis auf der linken Seite vom Edison. Nun wird nur IR-Licht aus der linken Infrarot-LED reflektiert.. Aus dem empfangenen Signal kann Edison feststellen, dass ein Hindernis auf der linken Seite steht, aber keines auf der rechten Seite.

Ausgesendetes Infrarot-Licht wird rot und reflektiertes IR-Licht wird blau dargestellt.



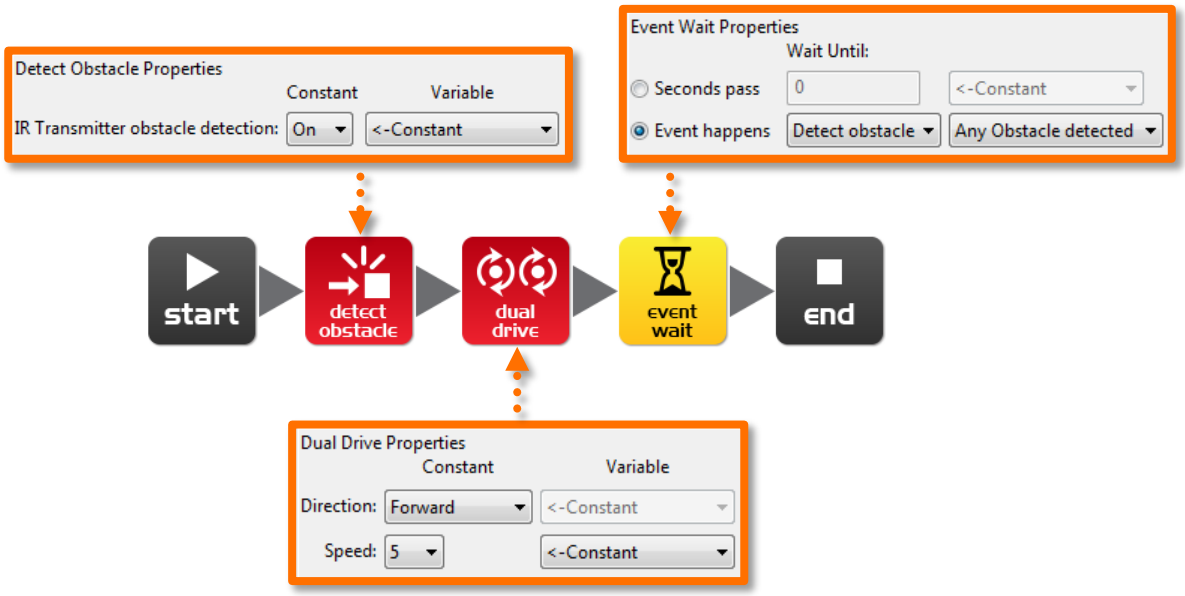

Zeichne das ausgesendete IR-Licht rot und reflektierendes IR-Licht unten blau ein



Name: _____

# Arbeitsblatt 7.2: Ein Hindernis erkennen und stoppen

Schreibe das folgende Programm, damit dein Edison fährt bis er ein Hindernis erkennt und dann stoppt.




Das rote Steuerungs-Icon „Hindernisse erkennen / detect obstacle“ ist zwingend am Anfang zu schreiben. So wird die Hindernis-Erkennung gestartet und dein Edison aktiviert die Infrarot-LEDs.

Die Geschwindigkeit ist auf 5 eingestellt, damit dein Edison genügend Zeit hat, ein Hindernis zu erkennen, bevor er mit ihm kollidiert. Wenn die Geschwindigkeit zu schnell eingestellt ist, wird dein Edison möglicherweise mit Hindernissen kollidieren.

## Ab welcher Distanz kann dein Edison Hindernisse erkennen?

_____

## Bist du dieser Art von Hindernis-Erkennung schon irgendwo begegnet? Wo?

_____

_____

## Wo denkst du, könnte diese Möglichkeit der Hindernis-Erkennung verwendet werden?

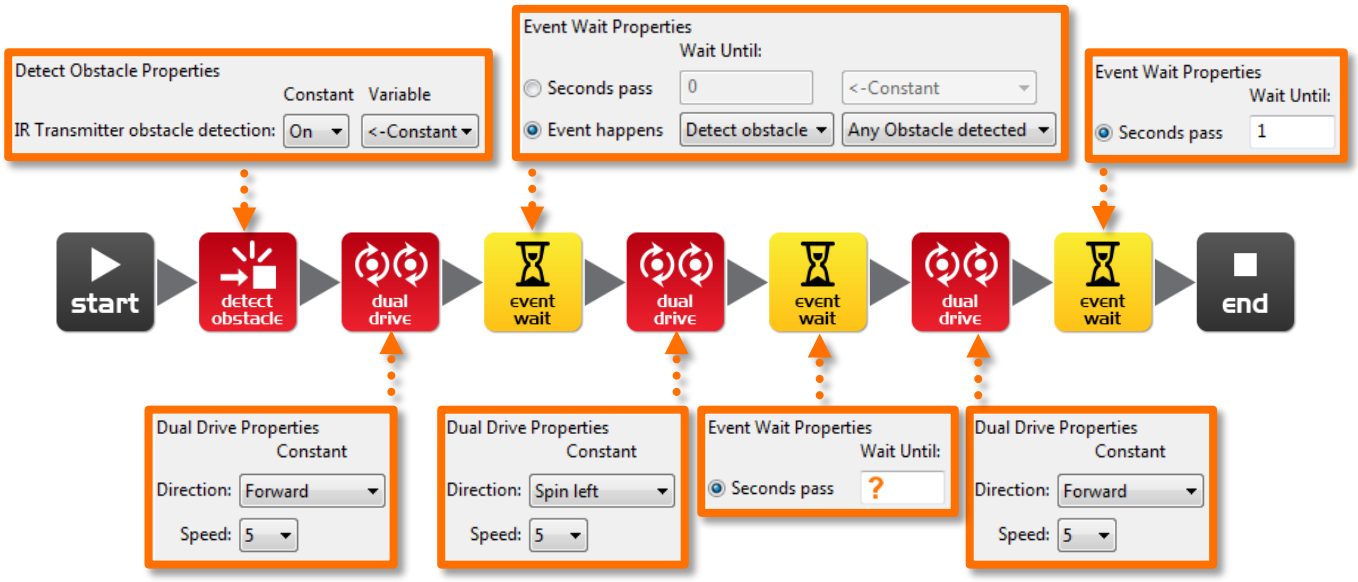
_____

_____

Name: _____

# Arbeitsblatt 7.3: Ein Hindernis erkennen und ausweichen I

Schreibe das folgende Programm, damit dein Edison fährt bis er ein Hindernis erkennt, sich dann um 180° dreht und 1 Sekunde vom Hindernis weg fährt.




In Lektion 3 (Arbeitsblätter 3.1 und 3.2) hast du herausgefunden, wie viel Zeit dein Edison für eine 180° Drehung braucht. Benutze wiederum diese Zeit und ersetzte sie mit dem orangen (**?**).

Welches ist die korrekte Zeitangabe, um deinen Edison um 180° drehen zu lassen?

_____

Was meinst du, warum ist dieses Programm nicht ganz vollständig? Wie könntest du es verbessern?

_____

_____

_____

_____

_____

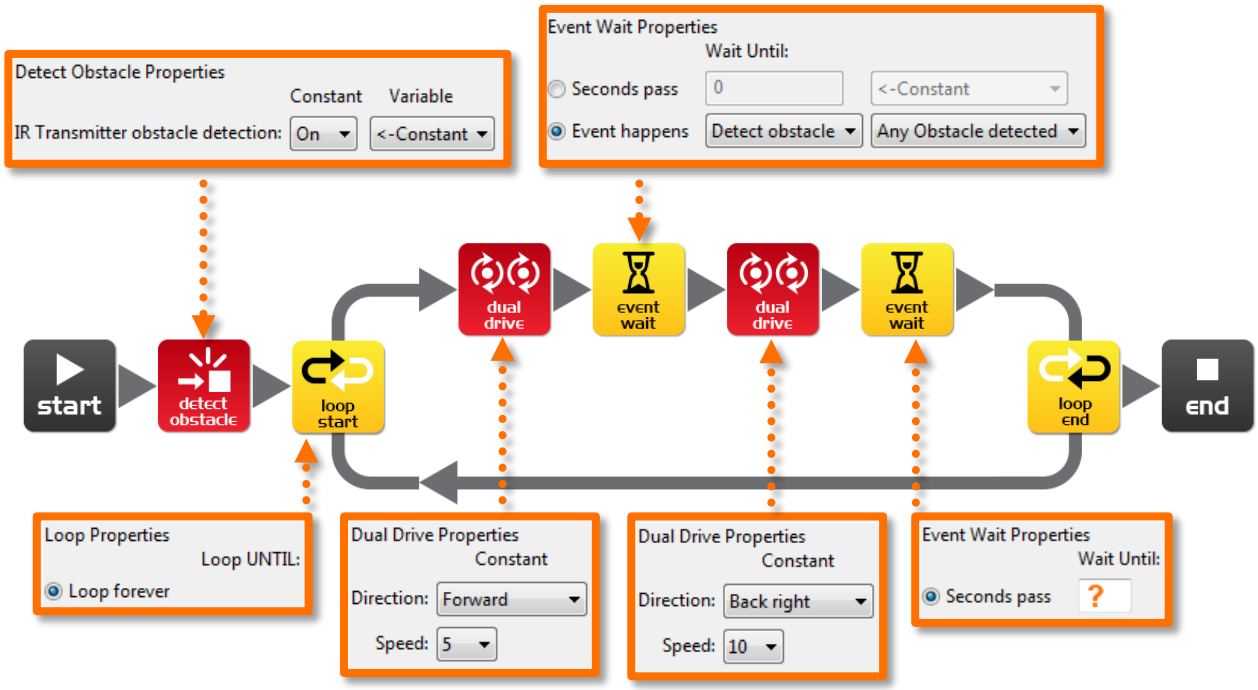
_____

_____

Name: _____

# Arbeitsblatt 7.4: Ein Hindernis erkennen und ausweichen II

Schreibe das folgende Programm, damit dein Edison immer wieder fährt bis er ein Hindernis erkennt, sich abdreht und weiterfährt.




Versuche, im zweiten „Ereignis warten / Event weit“, oben beim orangen (**?**) mit verschiedenen Zeiten zu experimentieren. Diese Zeiten bestimmen, wie lange sich dein Edison rückwärts dreht.

Welche Zeit denkst du, ist ideal für die Rückwärtsdrehung?

_____

Warum ist genau diese Zeit am besten? Was wäre bei kürzerer oder längerer Drehzeit anders?

Kürzere Drehzeit:

_____

_____

Längere Drehzeit:

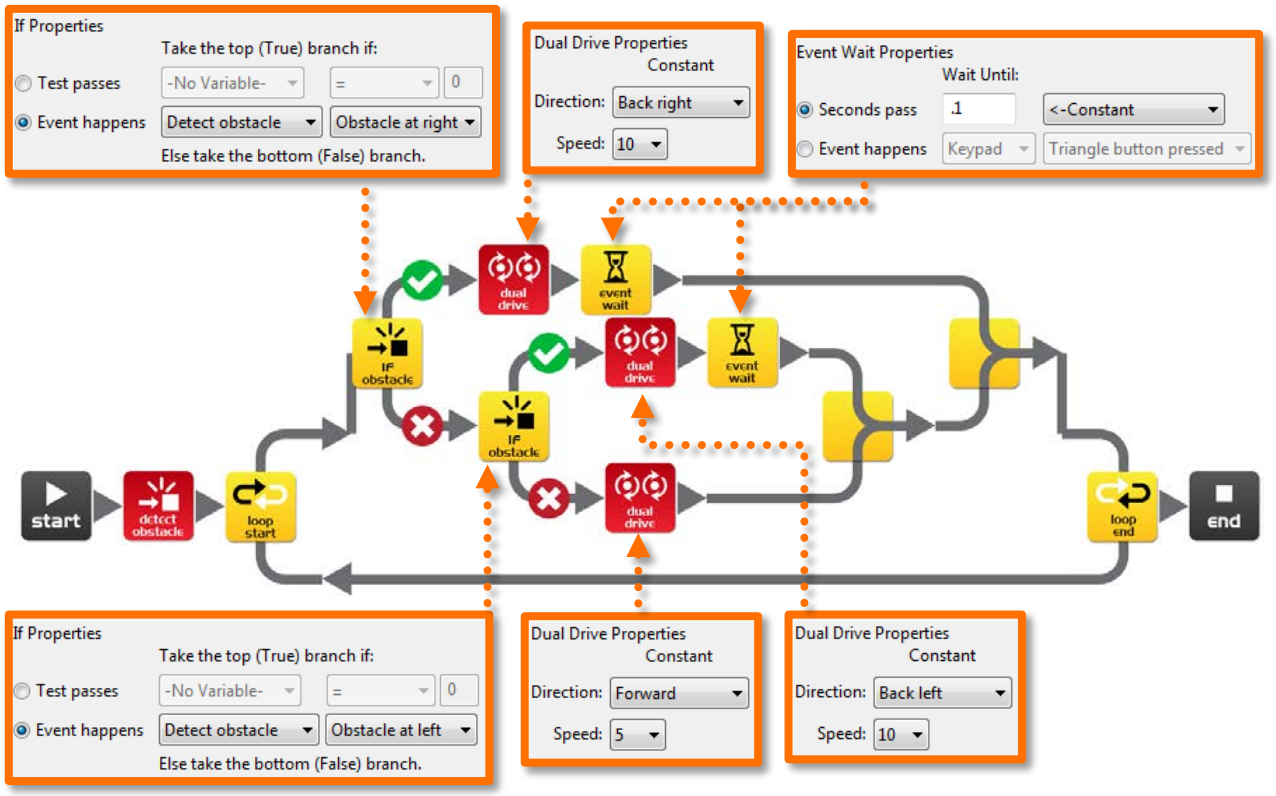
_____

_____

Name: _____

# Arbeitsblatt 7.5: Ein Hindernis erkennen und umfahren

Schreibe das folgende Programm, damit dein Edison immer wieder weiterfährt und den Hindernissen nach links und rechts ausweicht.




Im obigen Programm verwendest du das erste Mal eine „Wenn / If“-Abfrage. Das sind sehr wichtige Icons, da sie deinem Edison die Möglichkeit geben, Entscheidungen selber, das heisst ohne menschliche Führung zu treffen. Wenn dies ein Roboter tun kann, nennt man sie autonome Roboter, da er künstliche Intelligenz hat.

Ein „If“-Icon fragt ab, ob eine Bedingung wahr oder falsch ist. Wenn das Ergebnis wahr ist, nimmt das Programm den Pfad mit einem Häkchen. Wenn das Ergebnis falsch ist, nimmt das Programm den Weg mit dem Kreuz.

Das obige Programm hat drei verschiedene Wege, welches es auf der Grundlage eines Hindernisses nehmen kann.

Name: _____

Erkläre in eigenen Worten, was diese drei Pfade bei deinem Edison auslösen/bewirken.

Kein Hindernis erkannt: _____

_____

_____

Ein Hindernis auf der rechten Seite erkannt: _____

_____

_____

Ein Hindernis auf der linken Seite erkannt: _____

_____

_____

Dein Roboter trifft nun eigene Entscheidungen. Ist er daher lebendig?

_____

Warum denkst du so? Was sind deine Überlegungen?

_____

_____

_____

_____

Name: _____

# Arbeitsblatt 8.1: Wie funktioniert der Linien-Sensor?

Dein Edison ist mit einem Line-Tracking-Sensor (Linien-Sensor) ausgestattet. Der Sensor besteht aus zwei Hauptkomponenten:

1. Rotes LED-Licht ( LED =Light Emitting Diode)
2. Helligkeits-Sensor (Fototransistor)

Das rote LED-Licht leuchtet auf die Oberfläche, auf der dein Edison fährt. Wenn du die runde Taste auf Edison zweimal drückst, leuchtet die LED. Wenn du deinen Edison leicht anhebst, kannst du einen roten runden Lichtpunkt sehen, welcher die LED auf der Oberfläche erzeugt.

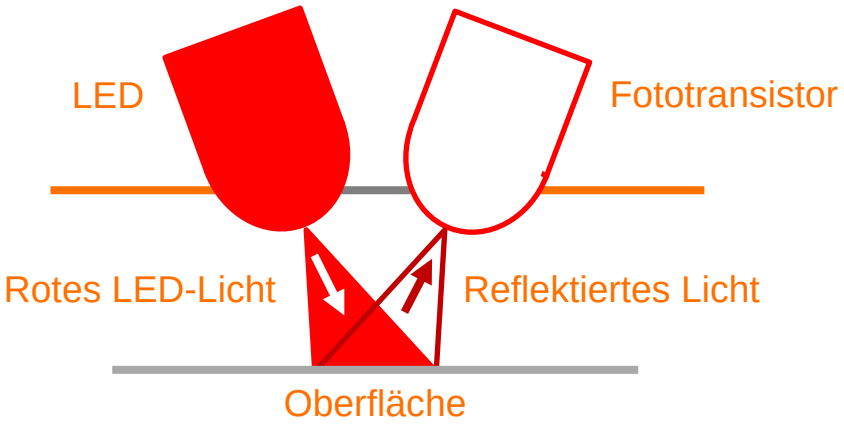
Vergleiche, wie hell der Lichtpunkt ist, wenn er auf eine schwarze oder weisse Oberfläche gestellt wird.

Ist der runde Lichtpunkt heller (das heisst er reflektiert mehr Licht) wenn er auf einer schwarzen oder einer weissen Oberfläche platziert wird?

_____

Der Fototransistor, als der Helligkeits-Sensor, ist ein spezieller Lichtsensor und er misst die Lichtmenge, die von der Oberfläche reflektiert wird.

Querschnitt des Line-Tracking-Sensors



Name: _____

Wie du aus der obigen Übung gesehen hast, wird mehr Licht von einer weissen Oberfläche als von einer schwarzen Oberfläche reflektiert. Daher misst der Fototransistor ein höheres Licht auf einer weissen Fläche als auf einer schwarzen Oberfläche. Dadurch kann dein Edison so programmiert werden, dass er auf die Oberfläche reagiert, auf der er fährt. Eine schwarze Oberfläche gilt als „Nicht-Reflektierend“ und eine weisse Fläche gilt als „reflektierend".

Was denkst du, wie würde der Linie-Tracking-Sensor auf farbige Oberflächen reagieren? Reflektierend oder Nicht-Reflektierend“?
Hinweis: Das LED-Licht vom Line-Tracking-Sensor ist rot.

Rote Oberfläche _____

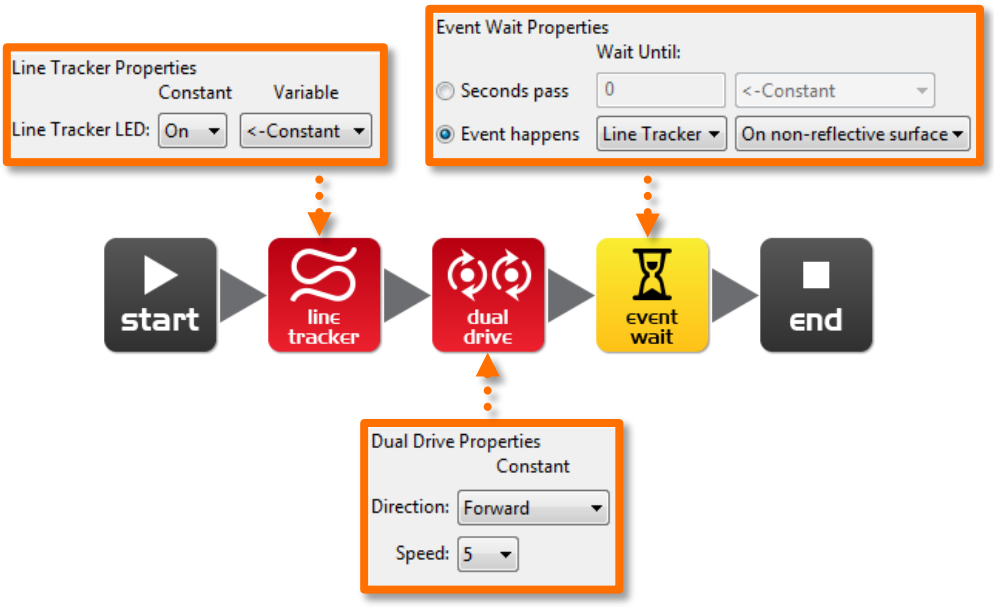
Grüne Oberfläche _____

Blaue Oberfläche _____

Name: _____

# Arbeitsblatt 8.2: Bis zu der schwarzen Linie fahren

Schreibe das folgende Programm, damit dein Edison so lange auf einer weissen Fläche (Reflektierend) fährt bis er eine schwarze Linie (Nicht-Reflektierend) kreuzt.




Um den Line-Tracking-Sensor in einem Programm zu benutzen, musst du zuerst den Sensor einschalten, dies aktiviert auch die rote LED. Dies tut das Icon „Linie verfolgen / line tracker“.

Benutze die schwarze Linie auf dem Arbeitsblatt „Acitvity 8.1“ oder male eine schwarze Linie auf ein weisses Papier. Stelle deinen Edison mit ein bisschen Abstand vo die schwarze Linie. Er wird nun fahren, bis er auf die schwarze Linie trifft und dann stoppen.

Auf dem Arbeitsblatt „Acitvity 8.1“ gibt es auch drei farbige Linien. Diese sind rot, blau und grün. Wiederhole die Aufgabe. Beobachte deinen Edison, wie er auf die unterschiedlich farbigen Linien reagiert.

Wie reagiert dein Edison auf die Farben? Gibt es eine Farbe, die Edison nicht sehr gut erkennen kann (siehe) Welche Farbe ist es? Kreuze an.

| | | |
|---|---|---|
| Grüne Linie | _____ | ☐ |
| Blaue Linie | _____ | ☐ |
| Rote Linie | _____ | ☐ |

Was denkst du, warum dein Edison so reagiert? Wieso ist dies so?

_____

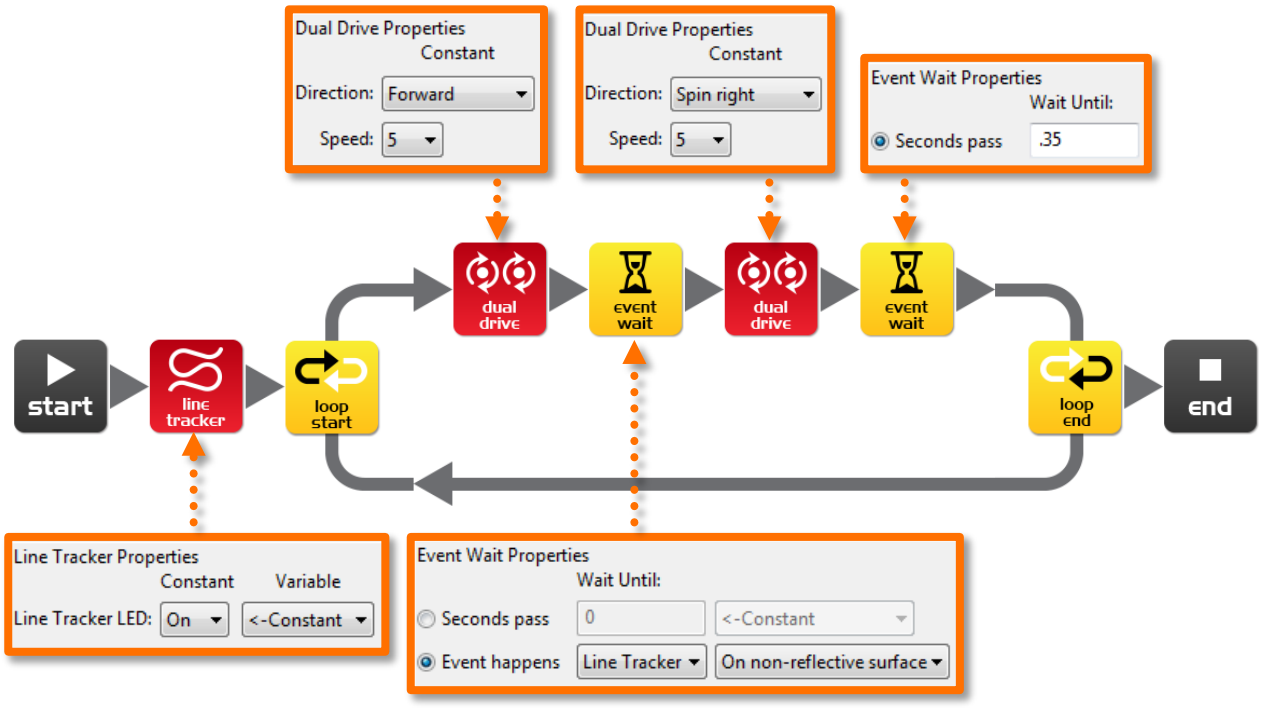
_____

_____

Name: _____

# Arbeitsblatt 8.3: Innerhalb einer Grenze fahren

Schreibe das folgende Programm, damit dein Edison innerhalb einer Grenze, einer schwarzen Linie fährt.




Benutze das Arbeitsblatt „Activity 8.2“ als Grenze oder zeichne eine eigene Grenze auf ein grosses Papier.

Wenn ihr ein grosses Papier mit einer Grenze habt: Versucht, mehrere Edisons hinzuzufügen. Beobachtet was passiert.

Du kannst auch mit verschiedenen Geschwindigkeiten experimentieren.

Wie schnell kann dein Edison fahren, damit der Line-Tracking-Sensor noch funktioniert?

_____

Ab welcher Geschwindigkeit gibt es Probleme/Schwierigkeiten?

_____

Was passiert, wenn dein Edison zu schnell fährt? Warum denkst du, ist das so?

_____

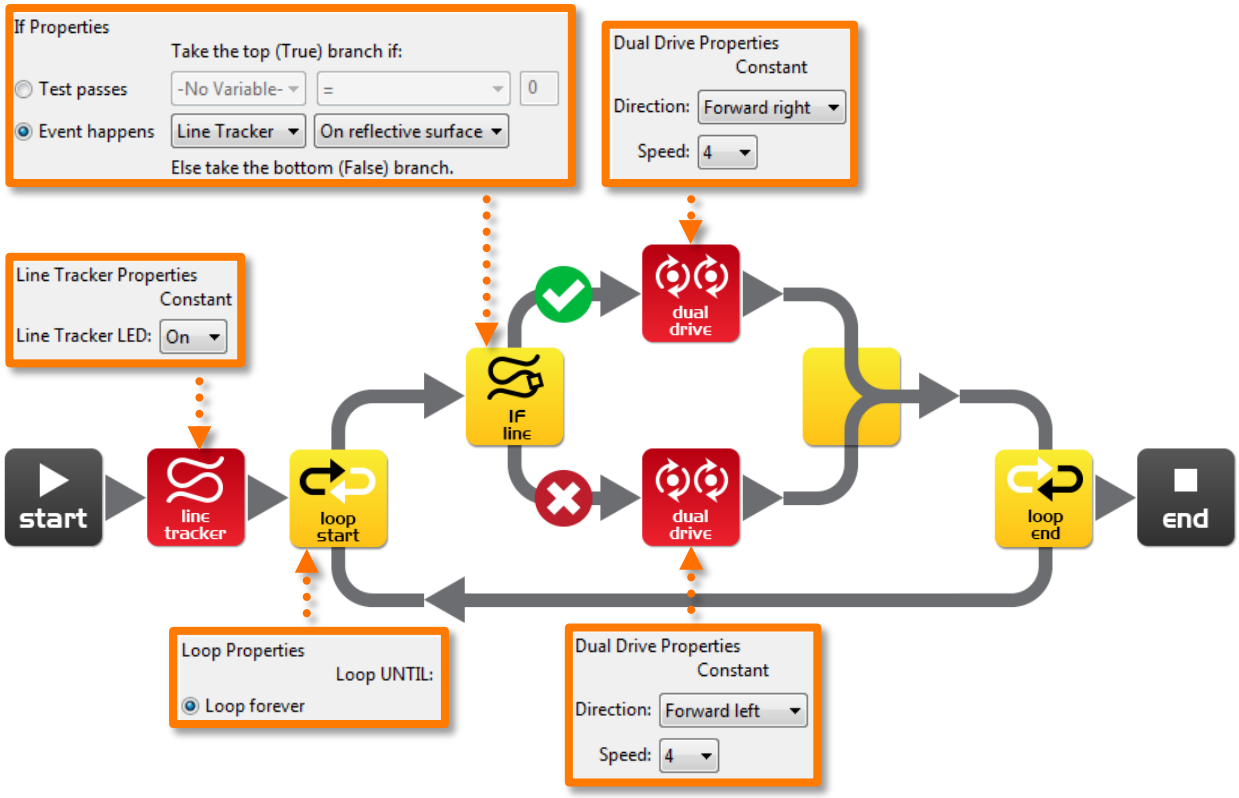
_____

_____

Name: _____

# Arbeitsblatt 8.4: Einer Linie folgen

Schreibe das folgende Programm, damit dein Edison einer schwarzen Linie folgt.




Dieses Programm verwendet wiederum die „if“-Abfrage, damit dein Edison entscheiden kann, was er tun soll. Du kannst selber bestimmen, wie er einer schwarzen Linie folgen soll:

Wenn der Line-Tracking-Sensor auf einer reflektierenden Oberfläche (weiss) trifft, fährt er mit der Geschwindigkeit 4 „Vorwärts-rechts“. Wenn der Line-Tracking-Sensor auf eine nicht-reflektierenden Oberfläche (schwarz) trifft, fährt er mit der Geschwindigkeit 4 „Vorwärts-Links“.

Lege deinen Edison auf die Linie von Arbeitsblatt „Activity 8.2“ und beobachte, wie er der Linie folgt.

In welche Richtung fährt dein Edison (mit dem obigen Programm) das Oval ab? Im oder gegen den Uhrzeigersinn?

_____

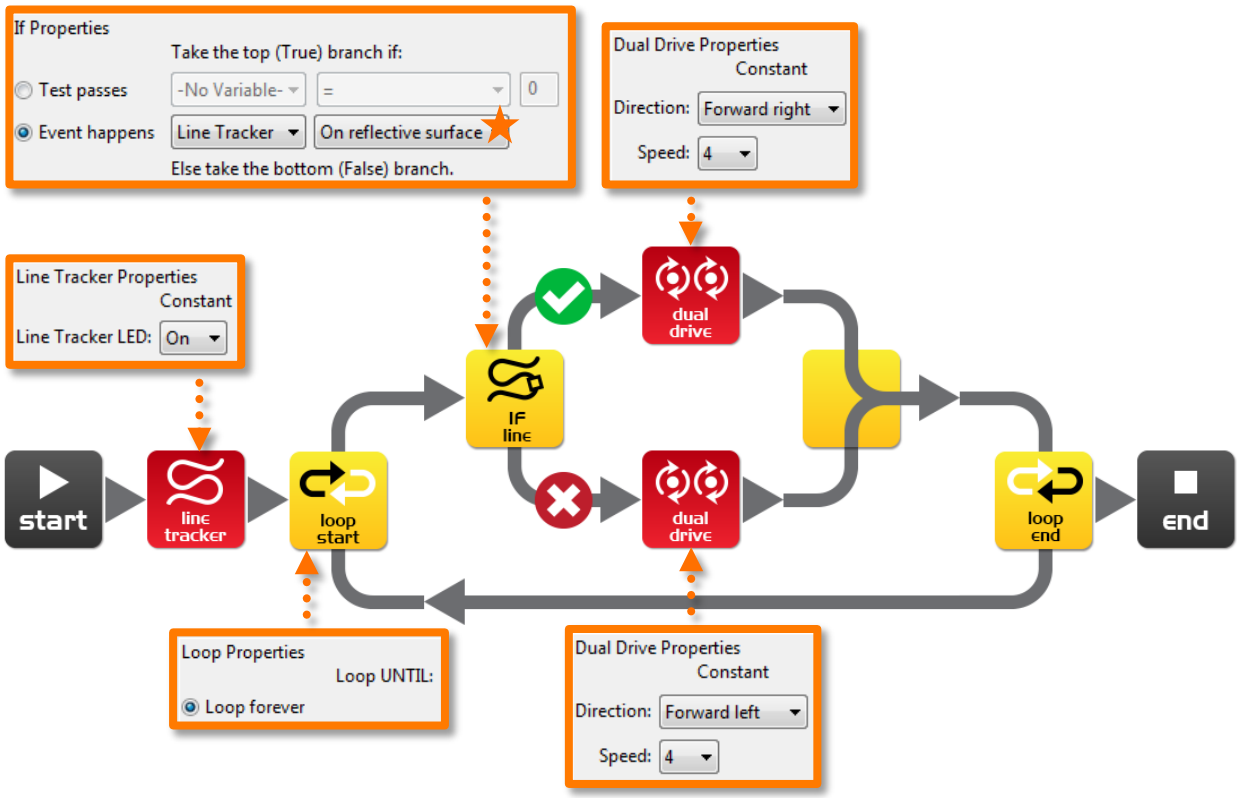
Warum denkst du, geht dein Edison in diese Richtung?

_____

_____

_____

Name: _____




Ändere nun die Einstellungen bei der „if“-Abfrage, beim orangen Stern so, dass er auf eine „nicht-reflektierende“ Oberfläche reagiert.

Wenn nun der Line-Tracking-Sensor auf einer nicht-reflektierenden Oberfläche (schwarz) trifft, fährt er mit der Geschwindigkeit 4 „Vorwärts-rechts“. Wenn der Line-Tracking-Sensor auf eine reflektierende Oberfläche (weiss) trifft, fährt er mit der Geschwindigkeit 4 „Vorwärts-links“.

Lege deinen Edison auf die Linie von Arbeitsblatt „Activity 8.2“ und beobachte wiederum, wie er der Linie folgt.

In welche Richtung fährt nun dein Edison (mit dem obigen Programm) das Oval ab? Im oder gegen den Uhrzeigersinn?

_____

Warum denkst du, geht dein Edison in diese Richtung?

_____

_____

_____

Kannst du die Unterschiede erklären?

_____

_____

_____

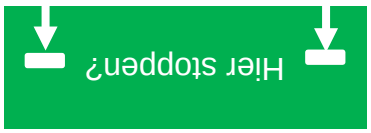
Name: _____

# Arbeitsblatt „Activity 8.1“

Benutze dieses Arbeitsblatt um die Farben zu testen.

Benutze dieses Arbeitsblatt um dein Programm von Arbeitsblatt 8.2 zu testen

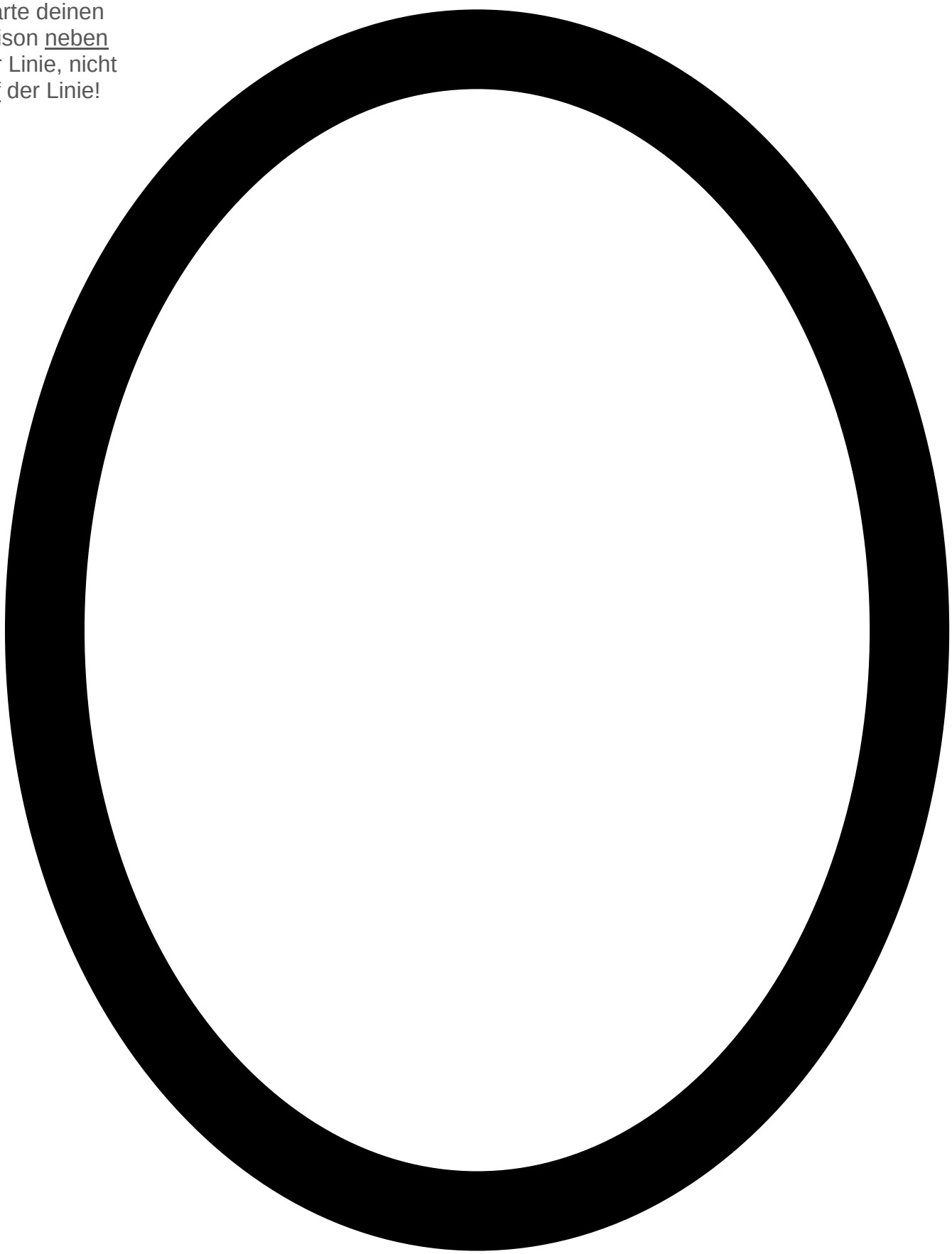


Name: _____

# Arbeitsblatt „Activity 8.2“

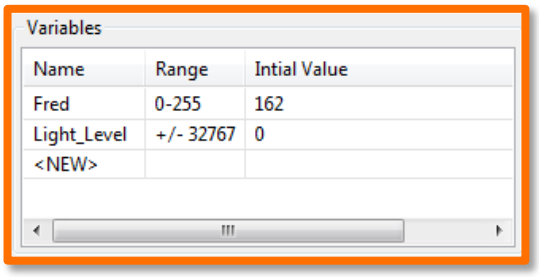
**Wichtig!**
Starte deinen Edison neben der Linie, nicht auf der Linie!



Name: _____

# Arbeitsblatt 9.1: Wie funktioniert der Licht-Sensor

Eine Variable ist ein kleines Stück Computer-Speicher für die Speicherung von Daten. Was Variablen so nützlich macht ist, dass diese Daten geändert werden können, währendem das Programm läuft, daher der Name Variable.



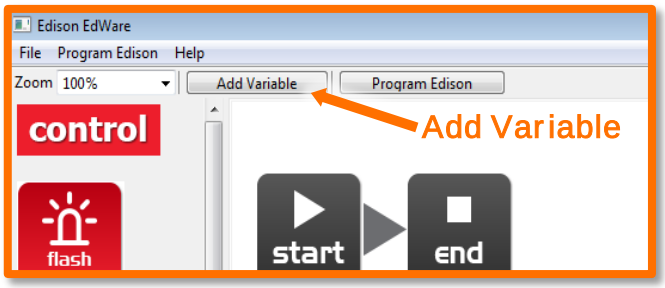

Variablen speichern Zahlen wie zum Beispiel 10, 106, 1.482 etc. und erlauben einem Computer-Programm mathematische Rechnungen zu machen. Das ist etwas, dass ein Computer perfekt machen kann.

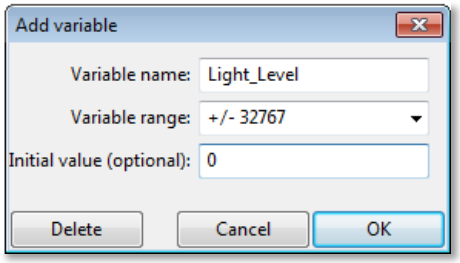
Dein Edison hat zwei Arten von Variablen, die „bytes" und „words". Byte-Variablen können Zahlen speichern, die von 0 bis 255 reichen. Word-Variablen können Zahlen speichern, die von -32.767 bis +32.767 reichen.

Damit du Variablen einfach benutzen kannst, gibst du ihnen Namen. Diese helfen dir, zu wissen, welche Art von Informationen in ihnen gespeichert ist. In EdWare kannst du deine Variablen genau so nennen, wie du willst. Du kannst einen „Fred" nennen. Aber das ist vielleicht kein sehr nützlicher Name als Erinnerung, welche Art von Informationen in Fred gespeichert ist. Ein besserer Name könnte „Licht_Level" sein. Diese Art von Namen macht es dir viel leichter zu merken, für was die Variable verwendet wird und welche Art von Daten dort zu finden ist. Benutze also immer hilfreiche, unterstützende Namen.

Da du nun mehr über Variablen weisst, kannst du in EdWare das nächste neue Programm schreiben.

Um eine neue Variable zu erstellen, klicke auf die Schaltfläche „Variable hinzufügen". Diese Schaltfläche findest du in der oberen linken Ecke. Darauf erscheint ein Popup-Feld.




Gib den Namen deiner Variablen ein, zum Beispiel eben „Light_Level". Wähle nun den Variablenbereich als +/- 32767 (word) und setze den Anfangswert auf 0. Klicke nun auf OK und deine Variable wird der Variablen-Tabelle in der rechten unteren Ecke hinzugefügt.

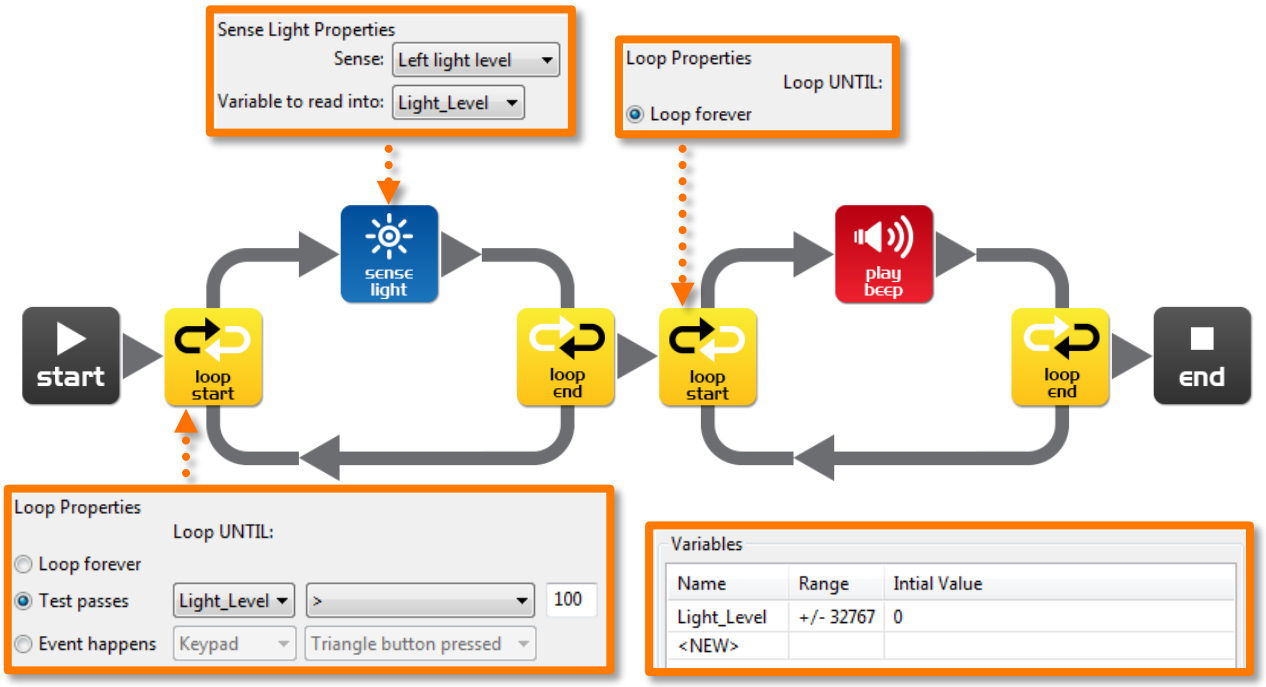
Welche Art von Variablen würdest du für die Speicherung der folgenden Zahlen (Byte oder Word) verwenden?

12 _____, 192 _____, 801 _____, -42 _____, 27,901_____

Name: _____

# Arbeitsblatt 9.2: Helligkeitsalarm

Schreibe das folgende Programm, damit dein Edison einen Alarm auslöst, wenn die Lichter im Raum eingeschaltet werden.




Das Icon „Licht erkennen / light level“ liest den Lichtpegel vom linken Lichtsensor und legt den Messwert in die Variable Light_Level.

Die erste Schleife benutzt Mathematik, um zu bestimmen, wie hell es ist. Die Schleife wird so lange wiederholt, bzw. die Helligkeit abgefragt, bis die Variable „Light_Level“ „grösser als“ (>) 100 ist.

Sobald der Wert in „Light_Level“ grösser als 100 ist, verlässt das Programm die erste Schleife und geht weiter zur nächsten Schleife. Diese Schleife löst den Alarm aus.

Stelle deinen Edison in einen dunklen Raum und drücke die Play-Taste. Wenn du nun das Licht einschaltest, ertönt der Alarm.

Kannst du dir vorstellen, wo ein solcher Alarm im echten Lebensinnvoll wäre?

_____

_____

Welche Änderungen müsstest du im Programm machen, damit dein Edison umgekehrt reagiert? Damit ein Alarm ertönt, wenn das Licht ausgeschaltet wird?

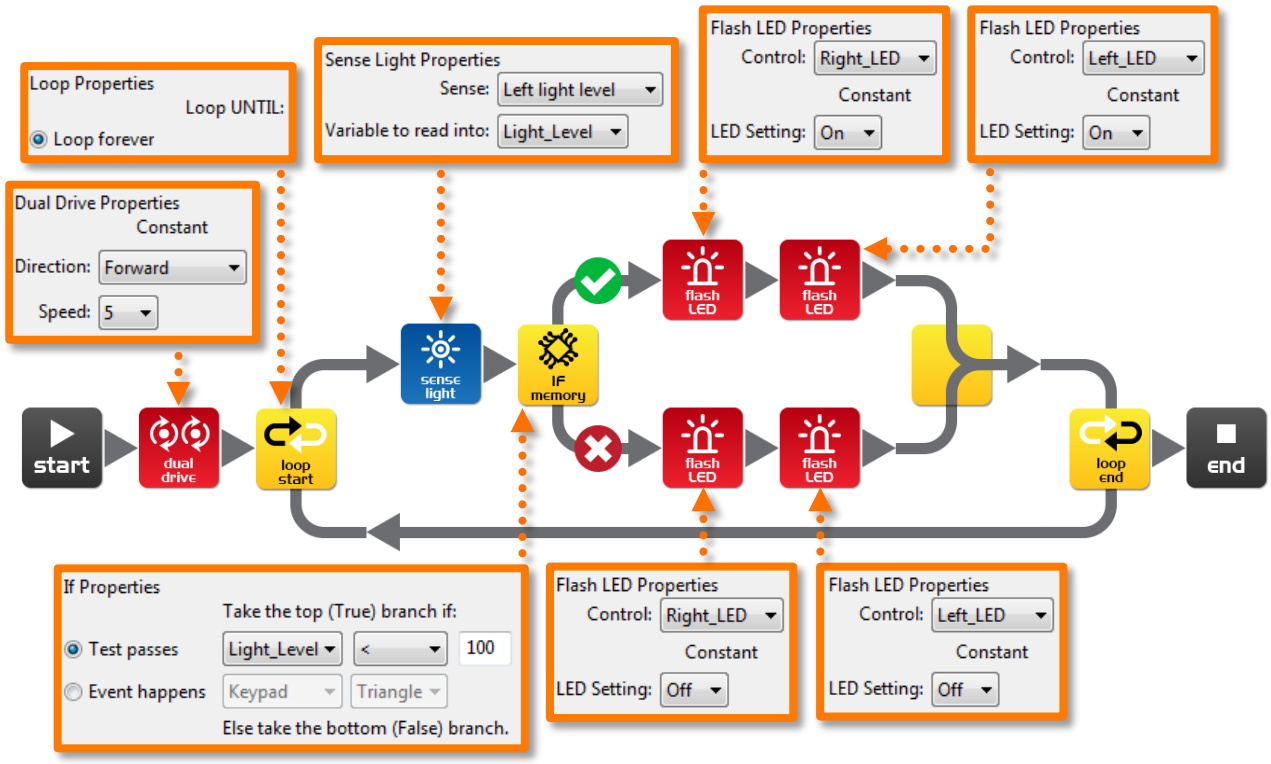
_____

_____

Name: _____

# Arbeitsblatt 9.3: Automatische Beleuchtung

Schreibe das folgende Programm, damit dein Edison die beiden roten LED-Lichter einschaltet, wenn es dunkel wird.




Lasse deinen Edison durch einen „Tunnel“ fahren. Beobachte vorne die beiden roten LED-Lichter.

Dieses Programm verwendet wiederum ein „If“-Icon, diesmal aber mit einer „kleiner als“ (<) Abfrage, um den Pfad des Programms zu bestimmen. Wenn die Variable Light_Level „kleiner als“ 100 ist, also das Ergebnis wahr ist, dann nimmt das Programm den Pfad mit dem Häkchen und die schaltet so die LEDs ein.

Experimentiere mit dem Wert 100 in der „if“-Abfrage.

Was passiert, wenn du den Wert höher als 100 einstellst?

_____

_____

Was passiert, wenn du den Wert tiefer als 100 einstellst?

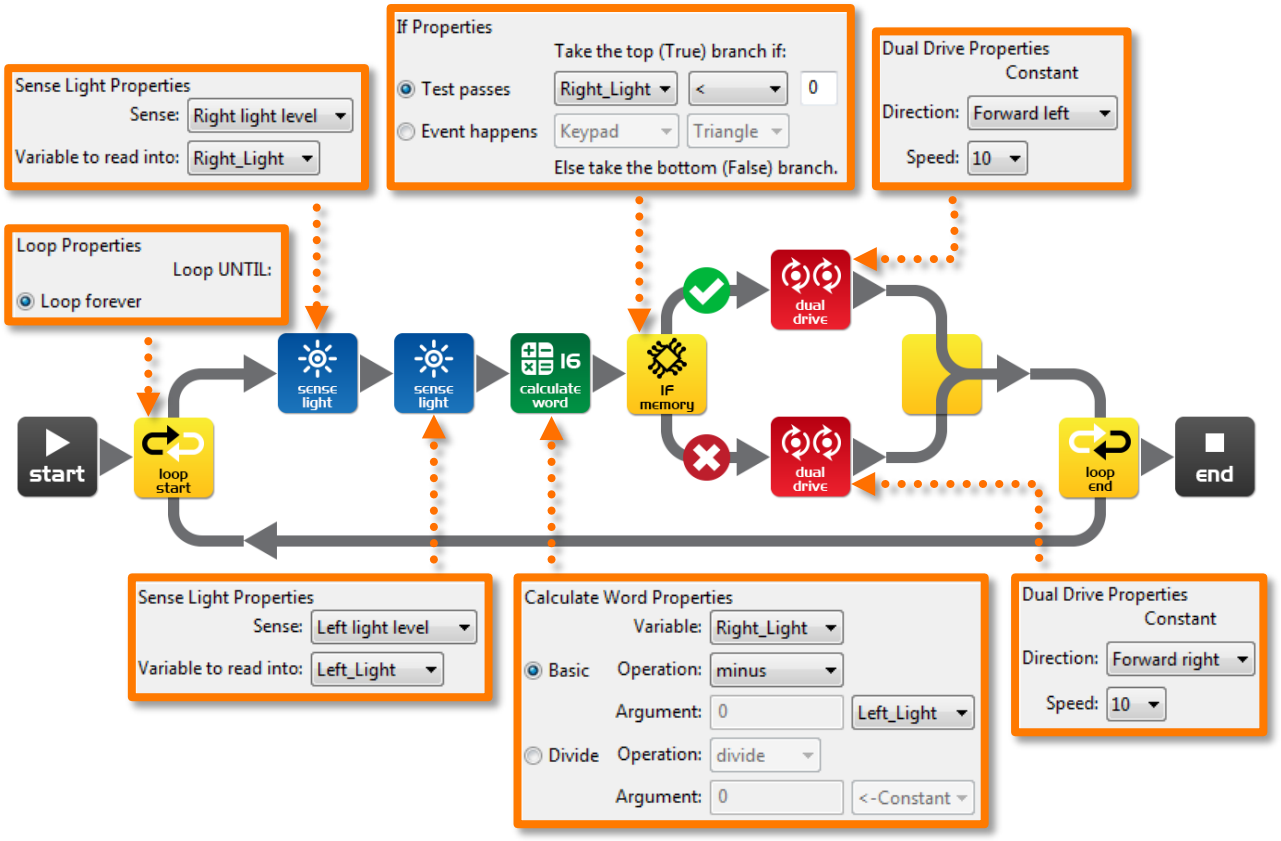
_____

_____

Name: _____

# Arbeitsblatt 9.4: Einem Licht folgen

Schreibe das folgende Programm, damit dein Edison dem Licht einer Taschenlampe folgt.




Leuchte mit einer Taschenlampe auf, bzw. kurz vor deinen Edison. Er wird dem Licht der Taschenlampe folgen.

Dieses Programm führt eine Berechnung zwischen zwei Variablen durch. Im Berechnungswort wird die Variable Right_Light von der Variablen Left_Light subtrahiert. Das Ergebnis (Antwort) wird wieder in die Variable Right_Light gesetzt. Wie das aussieht, kannst du hier sehen:

| | Right_Light | Left_Light | Calculation | Ergebnis |
|---|---|---|---|---|
| **Taschenlampe rechts** | 200 | 100 | 200 – 100 = | -100 |
| **Taschenlampe links** | 100 | 200 | 100 – 200 = | +100 |

Anhand des oben genannten Beispiels kannst du sehen, dass, wenn die Taschenlampe rechts ist, das Ergebnis unter Null (negative Zahl) ist. Wenn die Taschenlampe links , ist das Ergebnis über Null (positive Zahl) ist.

Das „If"-Icon fragt: Ist das Ergebnis kleiner als Null? Wenn das Ergebnis wahr ist, fährt dein Edison nach links, wenn das Ergebnis falsch ist, fährt dein Edison nach rechts.

Was würde passieren, wenn du das „kleiner als" (<) auf ein „grösser als" (>) änderst?

_____

_____

Name: _____

# Projekt II: Mein zweites Programm

Das Video " Humans need not apply " lieferte einen Einblick darüber, wie Roboter in Zukunft eingesetzt werden könnten. Überleg dir eine nützliche Aufgabe, die dein Edison ausführen kann und schreibe dann ein Programm, um diese Aufgabe zu erledigen.

Beispiele sind:

- Rettungsroboter - Der Edison fährt innerhalb einer Grenze, und sucht eine verlorene Person (Objekt, Minifigur oder Puppe). Wenn dein Edison die Person lokalisiert hat, ertönt ein Alarm.
- Fahrerloses Auto - Der Edison fährt auf einer bestimmten Strasse (Linie), ohne mit Menschen, anderen Autos oder Gebäuden (Spielzeug) zu kollidieren.

## 1. Ein Problem auswählen

Entscheide dich für eine nützliche Aufgabe, für eine Problemstellung. Diskutiere oder Besprich es mit einer Kameradin, einem Kameraden oder deiner Lehrperson, um zu entscheiden, ob du sie programmieren kannst oder nicht.

### Welche Ideen hast du?

_____

_____

_____

_____

_____

### Warum sind deine Ideen möglich, bzw. nicht möglich?

_____

_____

_____

_____

_____

Name: _____

## 2. Das Problem oder die Bewegungen beschreiben

Bevor du beginnst, die Problemstellung zu lösen, beschreibe das Problem, welches dein Edison lösen soll, in deinen eigenen Worten.

Das Problem ist …_____

_____

_____

_____

Als nächster Schritte versuche eine mögliche Lösung zu beschreiben, wie dein Programm aussehen könnte, welches deinem Edison sagt, wie er das Problem lösen soll.

Mein Edison wird dies lösen, indem er …_____

_____

_____

_____

## 3. Das Programm schreiben und testen

Bevor du dein Programm in der Software schreibst, mache dir Gedanken und skizziere/beschreibe diese.

Nun schreibe dein Programm, indem du die Icons nutzt, die du bereits kennen gelernt hast.

Teste dein Programm mit deinem Edison.

Name: _____

# 4. Fehler?

Nicht immer ist der erste Versuch erfolgreich. Ein zentraler Punkt bei der Programmierung ist es, dass Fehler machen dazu gehört! Fehler sind ein normaler Teil der Programmierung (oder einer Ingenieurdisziplin). Thomas Edison ist 10.000 Mal gescheitert, bevor es ihm gelang, die Glühbirne zu erfinden.

Beschreibe deine gemachten Fehler und versuche sie zu korrigieren.

_____

_____

_____

_____

_____

_____

Name: _____

# 5. Beschreiben der verwendeten Programmiersymbole und was sie bewirken

Zeichne die Icons, welche du in deinem Programm verwendet hast.

Beschreibe, was das Icon bewirkt, bzw. was dein Edison macht, wenn er es liest.

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

_____

_____

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

_____

_____

Wie heist das Icon? _____

Was bewirkt das Icon? _____

_____

_____

_____

_____

Name: _____

# Edison Pass

| Programm | Stamp |
|---|---|
| 2.1 Vorwärts fahren | |
| 2.2 Rückwärts fahren | |
| 2.3 Vorwärts- und Rückwärtsfahren | |
| 2.4 Geschwindigkeit anpassen | |
| 3.1 Rechtsdrehung | |
| 3.2 Linksdrehung | |
| 3.3 Rechts- & Linksdrehung | |
| 3.4 Mini-Labyrinth | |
| 4.1 Fahr-Challenge | |
| 4.2 La-Ola-Welle | |
| 5 Projekt I: Mein erstes Programm | |
| 6.1 LED reagiert auf Händeklatschen | |
| 6.2 Mit Händeklatschen fahren | |
| 6.3 Tanzen nach Händeklatschen | |

| Programm | Stamp |
|---|---|
| 7.1 Ein Hindernis erkennen und stoppen | |
| 7.2 Ein Hindernis erkennen und ausweichen I | |
| 7.3 Ein Hindernis erkennen und ausweichen II | |
| 7.4 Ein Hindernis erkennen und umfahren | |
| 8.2 Bis zu der schwarzen Linie fahren | |
| 8.3 Innerhalb einer Grenze fahren | |
| 8.4 Einer Linie folgen | |
| 9.2 Helligkeitsalarm | |
| 9.3 Automatische Beleuchtung | |
| 9.4 Einem Licht folgen | |
| 10 Projekt II: Mein zweites Programm | |
| Eigene Projekte/Programme 1. | |
| Eigene Projekte/ Programme 2. | |
| Eigene Projekte/ Programme 3. | |