

A photograph of a blue and white robot on a track. The robot is in the foreground, and a black ball is in the background. The track is a dark blue line with colorful segments (red, green, blue) on a light-colored tiled floor.

**CODING UND ROBOTIK IN
DER JUGENDMEDIENARBEIT**

**TIPPS UND IDEEN
AUS DER PRAXIS**

INHALT

- 03 >> Warum ist Coding wichtig?
- 04 >> Und was ist mit Robotik?
- 05 >> Was gibt es alles?
- 08 >> Ein Coding-Angebot planen und durchführen

UNSER AUTOR



Foto loosphoto.de

Johannes Rück arbeitet als Medienpädagoge in Berlin. Nach dem Studium der Sozialen Arbeit entwickelte Johannes Rück Workshops zu digitalen Medien und deren medienpädagogischen Herausforderungen. Zurzeit ist er Masterstudent der Spiel- und Medienpädagogik.

Gefördert vom

Ministerium für Kinder, Familie,
Flüchtlinge und Integration
des Landes Nordrhein-Westfalen



VORWORT

In der vorliegenden Ausgabe der **InterAktiv plus** werden Sie keine Schritt-für-Schritt-Anleitung zum Thema Coding finden. Auf diesem Feld wurde von anderen medienpädagogischen Akteuren bereits tolle Arbeit geleistet – sollten Sie auf der Suche nach solch einer Anleitung sein, schauen Sie sich bitte die Links am Ende dieser Publikation an! Wir möchten an dieser Stelle vielmehr unsere gewonnenen Erkenntnisse zur Durchführung von spannenden Workshops mit Ihnen teilen, einen Überblick über das Thema Coding und Robotik in der Jugendmedienarbeit geben und Ihnen Lust machen, erste eigene Schritte in diesem Bereich zu gehen oder Ihre Workshopkonzepte weiter zu verfeinern.

Impressum

Herausgeberin:

Landesarbeitsgemeinschaft Lokale Medienarbeit NRW e.V. (LAG LM)
Hedwigstr. 30-32 · 47058 Duisburg · Tel. 02 03/41 86 76 80
info@medienarbeit-nrw.de · www.medienarbeit-nrw.de

*V.i.S.d.P. Dr. Christine Ketzer · Autor Johannes Rück
Redaktion Arnold Hildebrandt, Dr. Christine Ketzer*

Korrektur Irina Ditter · Layout Alessandro Riggio · Fotos Johannes Rück

Wir übernehmen keine Haftung für die Richtigkeit, Vollständigkeit oder Aktualität der wiedergegebenen Informationen.



Namensnennung - Nicht kommerziell - Keine Bearbeitung

Warum ist Coding wichtig?

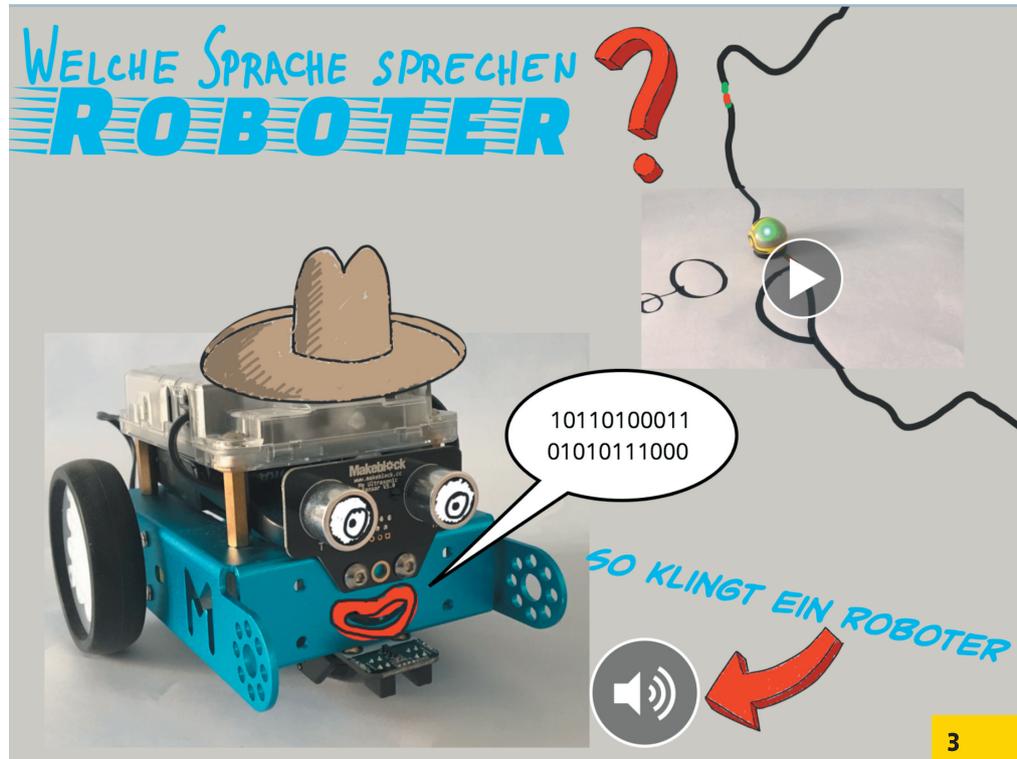
Coding bzw. das Erlernen einer Programmiersprache erlebt aktuell einen regelrechten Hype in der deutschen und internationalen Bildungslandschaft. Es gibt vermehrt kommerzielle und nicht kommerzielle Akteure, die Kinder und Jugendliche an das Thema heranführen möchten. Das Aneignen einer Programmiersprache wird als eine weitere Kulturtechnik neben dem Schreiben, Lesen und Rechnen angesehen. Sollen also alle Kinder und Jugendlichen später Programmierer oder Programmierinnen werden?

Diese Frage greift – wie auch die Frage, ob Kinder und Jugendliche nur noch mit dem Tablet lernen sollten – zu kurz. Beim Erlernen einer Programmiersprache geht es, ebenso wie bei der „Tablet-Frage“, um die dahinterliegenden zu erlernenden Kompetenzen und zu bearbeitenden Themen.

Durch die Beschäftigung mit dem Programmieren ist es möglich, Kindern und Jugendlichen ein grundlegendes Verständnis davon zu ermöglichen, wie ein Großteil der Technik um uns herum funktioniert. Wenn Kinder und Jugendliche die Logik hinter Algorithmen (siehe Kasten: Begriffe erklären) durchdrungen haben und beispielsweise für ein selbst programmiertes Quiz selbst einen entwickelt haben, ist der Grundstein gelegt, weitere Algorithmen, die uns täglich begegnen, zu identifizieren und zu besprechen. Es geht also ganz konkret um digitales Know-how.

Eine Programmiersprache ist, wie jede andere Sprache, nach speziellen Regeln wie Grammatik und Syntax aufgebaut. Maschinen arbeiten stumpf die Befehle ab, die wir ihnen mittels Code vorgeben. Sie besitzen keinen Interpretationsspielraum. Fehler in Grammatik oder Syntax können dazu führen, dass sich ein Programm gar nicht erst starten lässt oder dass es etwas Anderes tut als geplant. Genaues Lesen, Schreiben und Rechnen sind daher eine wichtige Fähigkeit beim Coden. Möchte ich in der Programmiersprache Scratch (siehe Kapitel: Coding digital) eine Figur genau platzieren, sollte ich deren Position im Code mit X- und Y-Koordinaten angeben; „oben links“ versteht das Programm nicht als Positionsangabe.

Zur Lese- und Schreibkompetenz gesellen sich logisches Denken und die Entwicklung von Problemlösungsstrategien, wenn Kinder und Jugendliche das Coden lernen. Jeder, der schon einmal etwas programmiert hat, kennt das Problem, dass das Programm einfach nicht das macht, was es soll. Um den Fehler zu



finden, der entweder ein Schreib- oder Logikfehler sein kann, ist es enorm wichtig, eine Problemlösungsstrategie zu entwickeln. Also beispielsweise den gesamten Code in einzelne Abschnitte zu unterteilen und diese nacheinander zu analysieren und sich auf die Fehlersuche zu begeben. Diese Frustrationstoleranz und Problemlösungsstrategien können in den unterschiedlichsten Lebenslagen hilfreich sein und einen großen Beitrag dazu leisten, die eigene Arbeitsweise zu strukturieren.

In meinen Workshops lasse ich die Kinder und Jugendlichen immer zu zweit arbeiten. Neben der Strukturierung der eigenen Arbeitsweise müssen die Teilnehmenden auch einen wertvollen kommunikativen und kooperativen Umgang miteinander finden.

Coding in der Jugendmedienarbeit kann auch Mädchen, Kinder mit Migrationshintergrund, Jugendliche mit Behinderung oder andere unterrepräsentierte Gruppen für das Thema Informatik begeistern. Dadurch kann Software, die einen Großteil unserer medialen Erlebnisse prägt, irgendwann auch aus nicht männlicher, nicht weißer Perspektive entstehen. Dies ist ein wichtiger Schritt zur Teilhabe möglichst aller an einer digital geprägten Gesellschaft.

Betrachtet man das Thema Coding also in einem größeren Kontext und bezieht diese Sichtweise in die Konzeption der eigenen Bildungsangebote mit ein, geht es für alle Beteiligten um weit mehr als das bloße Erlernen einer Programmiersprache.

Und was ist mit Robotik?



Coding und Robotik werden häufig in einem Atemzug genannt. Zu Recht! Roboter sind eine tolle Möglichkeit, Code in die reale, anfassbare Welt zu transferieren. Änderungen im Code können somit erfahrbar gemacht und über haptische Lernerfahrungen ergänzt werden. Wenn ich beispielsweise in einem Programm die Spielfigur falsch programmiert habe und diese mit voller Geschwindigkeit über den Bildschirm flitzt, ist das nicht so schlimm. Übertrage ich dieses Programm auf einen Roboter, können die Auswirkungen verheerend sein.

Viele Lernroboter wie der mBot (links im Bild) sind mit diversen Sensoren ausgestattet. Durch diese Sensoren können Verknüpfungen der digitalen mit der real erfahrbaren

Welt hergestellt werden, die somit mess- und auswertbar wird. Den mBot kann man zum Beispiel so programmieren, dass er eine Kurve fährt, sobald der Abstandssensor ein Objekt in unmittelbarer Nähe identifiziert. Die mittlerweile sehr beliebten Staubsaugerroboter funktionieren größtenteils mit derselben Logik und sind ein schönes Beispiel für das Zusammenspiel zwischen Code, Sensoren und Roboter.

Vielleicht saßen Sie vor Kurzem ja erst in einem Auto mit Parkdistanzsensoren, die ein schneller werdendes Piepen beim Rückwärtseinparken abgeben. Wie könnte der Algorithmus dazu aussehen?



Wiederhole so lange Motor = an; und Gang = R wenn Parksensoren <= 50 cm spiele Ton „Piep“

Projektidee

Voraussetzung:

- Computer oder Tablet mit einer Programmierumgebung (Scratch; mBlock Blockly)
- Lernroboter
- Filzstift und ein großes Blatt Papier (≥ DIN A2)

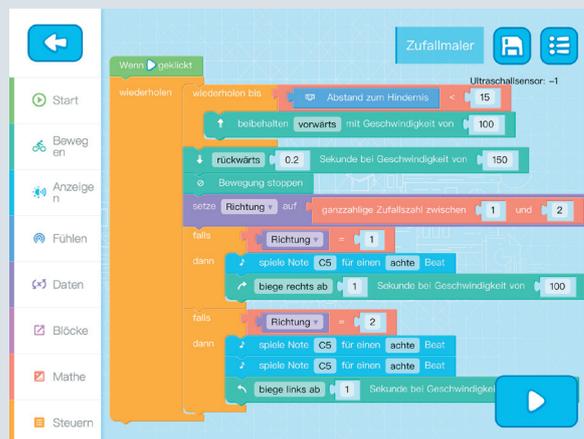
Analysieren Sie mit Kindern und Jugendlichen, wie ein Saugroboter aufgebaut ist und wie dessen Programmierung aussehen könnte.

Wenn die Teilnehmenden Hypothesen über die Funktionsweise von Saugrobotern aufgestellt und eine Logik bzw. einen Algorithmus entwickelt haben, können sie sich an die Programmierung machen.

Nun muss man einen Filzstift am Lernroboter befestigen und diesen auf ein großes Blatt Papier stellen. Wichtig: Bauen Sie auf jeden Fall eine Begrenzung um das Papier, damit der Roboter nicht den ganzen Fußboden bemalt.

Fährt der Roboter nun nach seiner Logik auf dem Blatt herum, werden die Auswirkungen der Software, genauso wie deren Veränderung, sichtbar, und die Kinder und Jugendlichen können nachvollziehen, wie so ein Saugroboter funktioniert oder wie die Logik in einem autonom fahrenden PKW aufgebaut sein könnte.

Der Code dazu könnte so aussehen:



Neben dem motivierenden, auffordernden Charakter, den ein piepender und blinkender Lernroboter haben kann, bietet die Robotik auch immer eine schöne Gelegenheit, kreativ zu werden. Einprogrammierte Melodien können mit der passenden Lightshow blinkender LEDs untermalt werden, und der Roboter kann modifiziert werden und neue Bauteile erhalten. Der mBot hat beispielsweise oben auf dem Gehäuse eine Fläche, die mit Lego kompatibel ist. Sie ahnen es bereits, daraus ergeben sich etliche neue Möglichkeiten für spannende Projekte! Dieser kreative Freiraum und auffordernde

Charakter motiviert viele Kinder und Jugendliche, die sich vielleicht ursprünglich nicht mit Logik und Problemlösungsstrategien beschäftigen wollten.

Ich persönlich setze die Roboter frühestens in der zweiten Workshophälfte ein, nachdem die Kinder und Jugendlichen erste Programmiererfahrungen gemacht haben. Dies hilft ihnen zum einen, die Verknüpfung von Coding und Robotik besser nachvollziehen zu können, und verringert zum anderen die Wahrscheinlichkeit einer Überforderung durch zu viele neue Inhalte auf einmal.

Was gibt es alles?

Coding analog – Programmieren ohne Internet und Strom

Nicht nur wegen der häufig mangelhaften technischen Ausstattung von Bildungs- und Jugendeinrichtungen lohnt es sich, Methoden zum Programmieren ohne Strom und Internet im Repertoire zu haben. Analoges Coding besteht nicht nur durch seine niedrigen materiellen Anforderungen, sondern eignet sich ideal, um Kinder und Jugendliche an das Thema Programmieren heranzuführen. Im Wesentlichen geht es darum, zu vermitteln, dass ein Programm oder ein Roboter, so clever er wirken mag, nur Befehle ausführt, die jemand anderes vorher einprogrammiert hat. Hinzu kommt, dass sich für Kinder und Jugendliche hier eine gute Gelegenheit bietet, sich vom Bildschirm wegzubewegen.

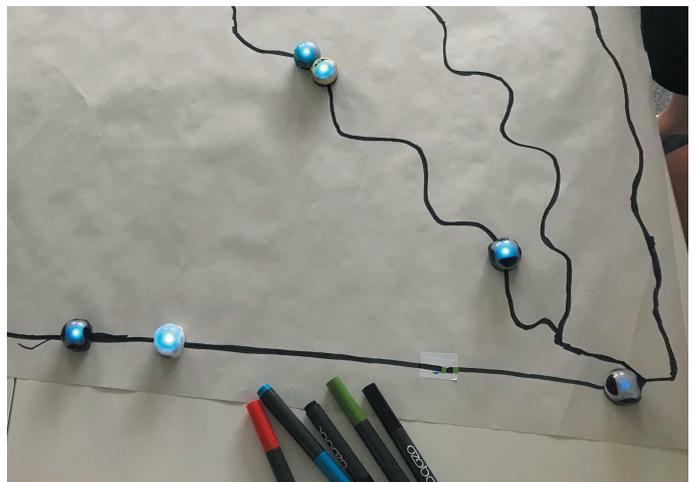
1. Die Kinder und Jugendlichen erhalten allein oder in einer Kleingruppe eine Grafik, welche die restliche Gruppe nicht sehen darf. Für einen Malroboter soll nun der Code geschrieben werden, damit er eben diese Grafik ausgibt.
2. Der Malroboter ist nicht wie im vorherigen Abschnitt ein tatsächlicher Roboter, sondern ein Kind/Jugendlicher aus einer anderen Gruppe.
3. Die Kinder und Jugendlichen können aus den unten aufgeführten Befehlen wählen.
4. Ist der Code geschrieben, wird er mit den anderen Teilnehmenden/Kleingruppen ausgetauscht, sodass jeder eine Programmierung erhält, zu der die Grafik unbekannt ist.
5. Der Roboterarm startet im Feld links oben. Die Kinder und Jugendlichen sollen sich nun wie Roboter verhalten und stumm den Befehlen folgen, die sie erhalten haben. In ein leeres Raster sollen sie dem Code entsprechend malen.
6. Entspricht die erstellte Grafik der ursprünglichen Grafik? Wenn nein, wo liegt der Fehler? Im Code oder im Roboter?

- Ein Feld nach rechts bewegen
- ← Ein Feld nach links bewegen
- ↑ Ein Feld nach oben bewegen
- ↓ Ein Feld nach unten bewegen
- ↻ Dieses Feld ausmalen

7. Die Kinder und Jugendlichen können eigene Grafiken wie Herzen und Smilies in Rasterform entwickeln.



Eine weitere tolle Methode für Coding ohne Computer bilden die Ozobots. Ozobots sind kleine Roboter, die Linien folgen und über Farbcodes Befehle ausführen können. Damit funktionieren sie unabhängig von der technischen Infrastruktur vor Ort und ermöglichen einen analogen und kreativen Einstieg in die Welt des Codings. Auch hier gibt es bereits viele Projektideen, sehen Sie also in den Links nach!

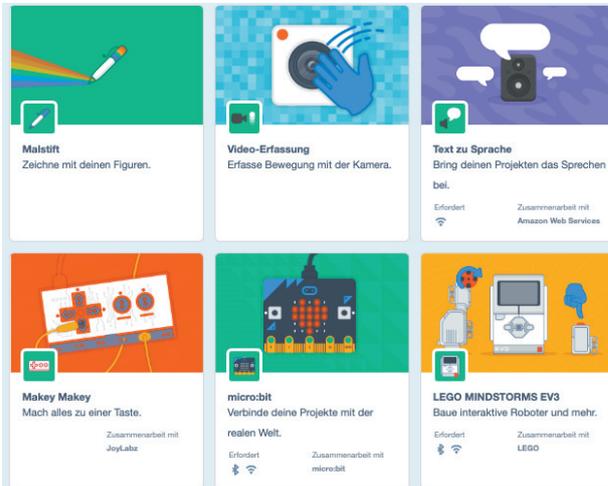


OzoBots sind kleine Roboter die auf der Unterseite mit Sensoren ausgestattet sind. Dies ermöglicht ihnen einer schwarzen Linie zu folgen, aber auch verschiedene Befehle, OzoCodes genannt, zu „lesen“.

Vorerst stehen den TN folgende Befehle zur Verfügung:

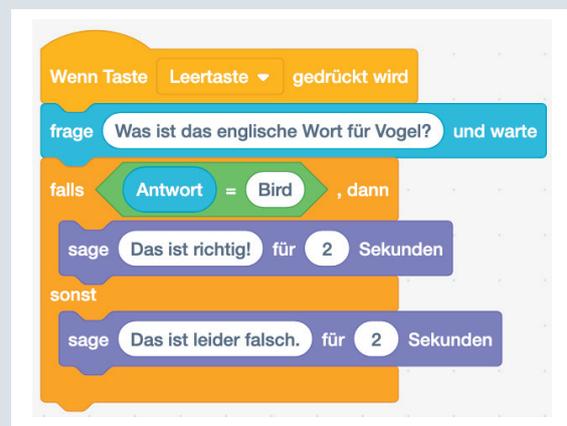
-  am Ende umdrehen
-  nach links springen
-  nach rechts springen
-  Ende

Coding digital – Programmieren am Tablet oder Computer



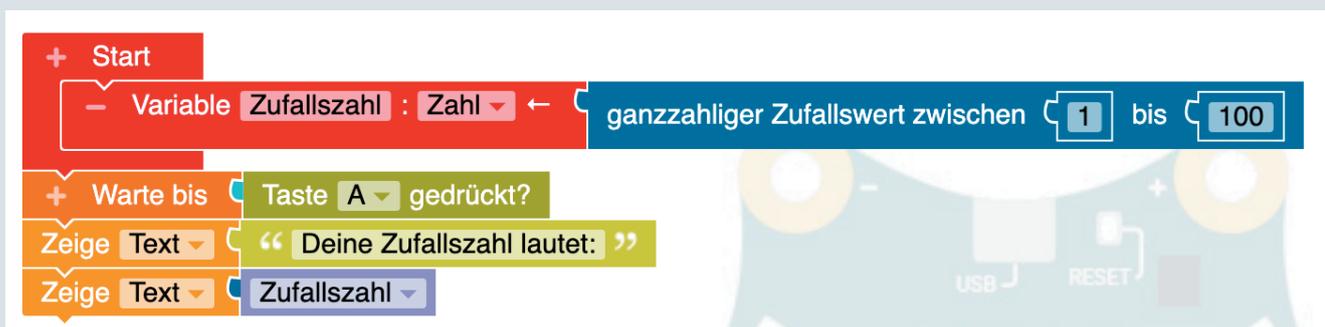
Es gibt die unterschiedlichsten Programmiersprachen für den Bildungsbereich. Viele dieser Programmiersprachen ähneln sich stark und funktionieren nach demselben Prinzip. Der Code besteht aus vielfältigen Blöcken, welche wiederum wie Legosteine aneinandergesteckt werden. Es existieren Bauteile, die ineinanderpassen, aber auch welche, die sich nicht zusammenstecken lassen. Dadurch können bereits einige Logikfehler vermieden werden (siehe Bilder unten).

Ein Vorteil von Scratch ist die große Verbreitung und die ständige Weiterentwicklung. Es lassen sich beispielsweise Module hinzufügen, die eine Programmierung von Lego-Mindstorms-Robotern oder die Steuerung des eigenen Programms über die Webcam ermöglichen. Zu Scratch selbst gibt es bereits tolle Handbücher unter OER-Lizenz (Open-Education-Resource); Sie finden eine Auswahl bei den Links.



So sieht ein Code in Scratch aus. Hier sieht man schön, dass die geometrische Form der Blöcke in manchen Bausteinen vorgegeben ist.

Im Vergleich dazu die Programmieroberfläche von Open Roberta. Diese wird für die Programmierung der Calliope Minis genutzt. Sie unterscheiden sich zwar in der Bezeichnung der Blöcke, die dahinterstehende Logik ist aber dieselbe wie bei Scratch oder anderen Programmiersprachen. Die in Scratch erlernten Kompetenzen können also auch abseits dieses Programms eingesetzt werden.



Scratch – Die Community und der Pädagogen-Account

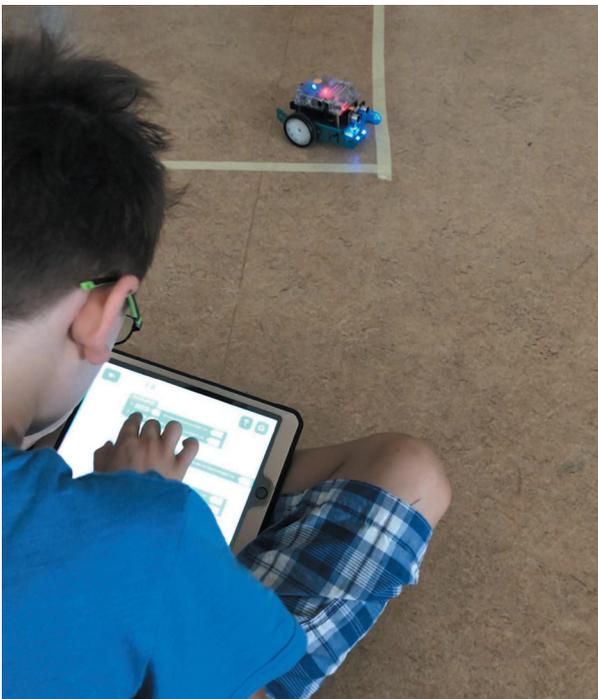
Scratch ist die beliebteste Programmiersprache im Bildungsbereich. Dies hat mehrere Gründe:

- Scratch ist kostenlos.
- Scratch wurde für Kinder und Jugendliche entwickelt.
- Scratch kann offline als eigenständiges Programm genutzt werden oder online im Browser ohne vorherige Installation.
- Scratch hat eine großartige Community: Kinder und Jugendliche können ihre eigenen Programme teilen, kommentieren und remixen. Somit bietet sich zudem eine erstaunliche Möglichkeit, mit Kindern und Jugendlichen den Umgang in Online-Communitys zu üben.
- Sie können sich bei Scratch einen Pädagogen-Account anlegen. Damit lassen sich Accounts in Gruppenstärke organisieren, und Sie können sehen, was ihre Jugendlichen programmieren.

Coding digital – Programmieren am Tablet oder Computer mit zusätzlicher Hardware

mBots

Die mBots sind robuste Lernroboter, die sich am Computer mit Scratch programmieren lassen. Der geschriebene Code wird per Kabel oder Bluetooth-Verbindung auf die Roboter übertragen und diese führen ihn aus. Alternativ können die mBots auch mit dem Tablet programmiert werden. Sollten Sie diese Möglichkeit haben und Ihnen stehen ausreichend Tablets zur Verfügung, kann ich dies nur empfehlen. Kinder und Jugendliche können in der Regel gut mit dem Tablet umgehen und lernen hierbei eine produktive Anwendung abseits von konsumorientierten Apps wie Instagram und YouTube kennen.



TIPPS

Die App mBlock Blockly bietet ein tolles Tutorial, das die Teilnehmenden nach und nach in die Programmierung der Roboter einführt. Hier sind wirklich keine Vorkenntnisse erforderlich!

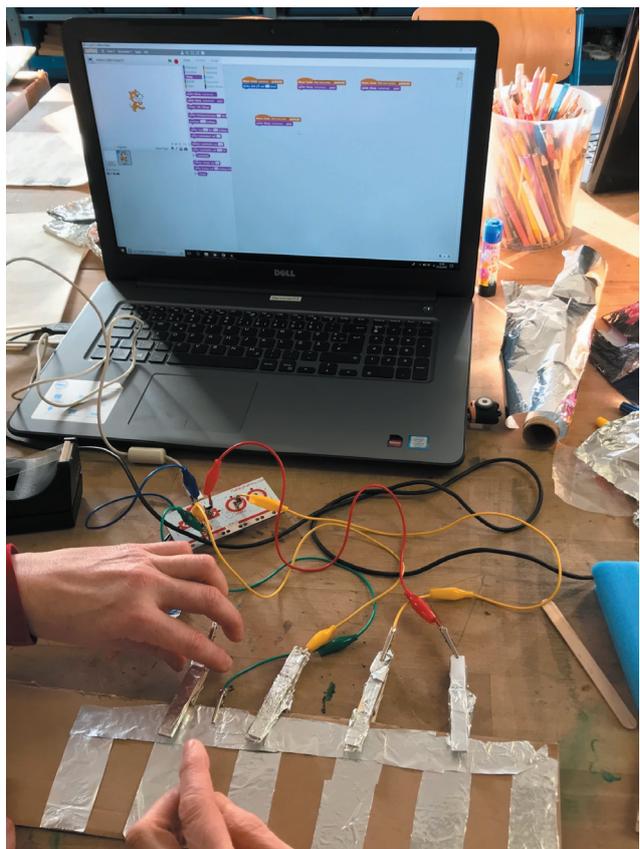
Wenn Sie mit Tablets und mBots arbeiten können, lassen Sie die Kinder und Jugendlichen auf dem Boden agieren. So besteht keine Gefahr, dass die mBots von den Tischen fallen. Mit Klebeband grenze ich auf dem Boden Vierecke ab, die den Teilnehmenden als Begrenzung dienen, innerhalb derer sich die mBots frei bewegen können.

Makey Makey

Makey Makeys sind kleine Platinen, die von jedem Computer als Tastatur erkannt werden. Hier ist keine Installation nötig. An den Makey Makey lassen sich mittels Krokodilklemmen selbst gebaute Tasten anschließen. Dadurch wird die kognitive Dimension des Codens um eine haptische erweitert. Gelerntes kann so besser verankert werden, sorgt oftmals für Abwechslung und bietet die Möglichkeit, sich gestalterisch und kreativ auszudrücken.

Des Weiteren bietet Makey Makey eine tolle Option, mit Kindern und Jugendlichen über assistive Technologien und Barrierefreiheit zu sprechen. Wie bedienen Menschen ein Computerprogramm, wenn sie ihre Hände nicht einsetzen können?

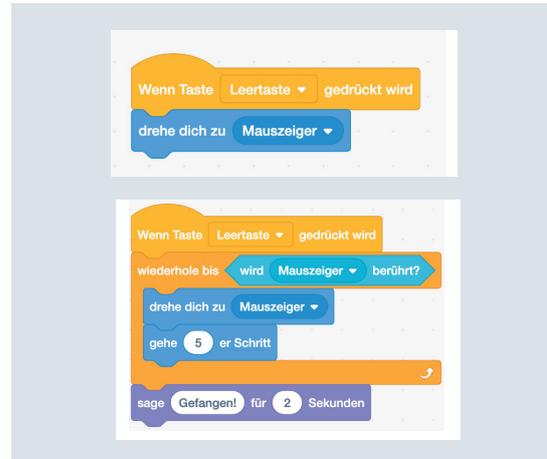
Ein umfangreiches Handbuch zum Einsatz von Makey Makey finden Sie bei den Links.



Ein Coding-Angebot planen und durchführen

Ein grober Ablauf

- **Begrüßung und Einführung ins Thema** – Worum soll es heute gehen?
- **Gemeinsam relevante Begriffe klären** (siehe Kasten) – Worüber sprechen wir hier eigentlich? Warum ist das wichtig bzw. wo betrifft dieses Thema die Kinder und Jugendlichen in ihrem Alltag?
- **Einen Alltagsbezug herstellen** – Welche Algorithmen kennen die Kinder und Jugendlichen bereits bzw. welche führen sie selbst regelmäßig aus?
- **Analoges Coding** – mittels entsprechender Methode oder Material
- **Digitales Coding I** – Das erste Programm analysieren – Die Kinder und Jugendlichen bekommen die Programmieroberfläche (hier Scratch) gezeigt und sollen Hypothesen zu einem simplen Programmierbeispiel aufstellen. Was macht dieses Programm?
- **Digitales Coding II** – Das erste Programm schreiben – Die Kinder und Jugendlichen können nun die ersten eigenen Schritte gehen und das Programm um einige wenige Zeilen sinnvoll erweitern.
- **Digitales Coding III** – Das erste Programm erstellen – Die Kinder und Jugendlichen entwickeln erste kleine Programme anhand vorbereiteter Coding-Projektvorlagen (beispielsweise App Camps oder Coding for Tomorrow, siehe Links).



Links und Materialien

- **Scratch**
- **Open Roberta**
- **LightBot** Im Browser oder als iOS/Android-App nutzbar
- **mBots**
- **App mBlock Blockly** Die App mBlock Blockly ermöglicht das Programmieren der mBots mit dem Tablet. Hier gibt es tolle Einsteiger-Aufgaben, mit denen die Kinder und Jugendlichen nach und nach die wichtigsten Befehle kennenlernen.

Handbücher

- **Scratch-Handbuch** der Pädagogischen Hochschule Schwyz
- **Makey-Handbuch** der Pädagogischen Hochschule Schwyz
- **Otobot-Handbuch** der Pädagogischen Hochschule Schwyz

Projektmaterialien und Projektideen

- **Digital Learning Lab**
- **Coding for Tomorrow – Vodafone Stiftung**
- **App Camps** Projektvorlagen – Unterrichtsmaterialien zu Scratch und weiteren Coding-Themen. Anmeldung erforderlich
- **kinderlabor.ch** Unterrichtsmaterialien zu Scratch/Coding-Handbücher für Lehrer
- **Swisseduc** Projektvorschläge zu Scratch
- **Generation Robots** Sammlung von Unterrichtsmaterialien für diverse Lernroboter
- **Ozobot-Unterrichtsmaterialien**
- **Programmieren ohne Strom**
- **inf-schule.de** Computer und Digitales allgemein

8

Begriffe erklären – Alle auf einen Stand bringen

So divers wie viele Jugendgruppen sind meistens auch deren Vorkenntnisse in Bezug auf Coding oder andere technologische Themen. Zusammen mit der Zielgruppe wichtige Begriffe zu klären, ist aber nicht nur deshalb wichtig, um auf einem gemeinsamen Stand zu sein, sondern auch, weil sich dabei viele spannende Punkte und Gespräche ergeben. Zudem können so weitere Kontexte für die Kinder und Jugendlichen hergestellt werden.

TIPP

Sie suchen nach einer einfachen Begriffsdefinition für Kinder? Nutzen Sie eine Kindersuchmaschine wie beispielsweise www.blinde-kuh.de.

Begriff	Definition
App/Programm	Apps oder Programme enthalten Befehle, die dem Computer sagen, was er tun soll. Apps und Programme nennt man auch Software.
Chip/Prozessor	Chips oder Prozessoren führen diese Befehle aus und geben beispielsweise eine Eingabe über die Tastatur an den Bildschirm weiter. Sie sind wie eine Schaltzentrale.
Browser	Der Browser ist ein Computerprogramm, mit dem man sich Internetseiten anschauen kann.
Algorithmus	Ein Algorithmus ist eine Abfolge mehrerer Befehle.
Code	Code werden die Befehle genannt, die in einem Programm stecken.
Sensor	„Sensus“ ist lateinisch und bedeutet „Gefühl“. Sensoren ermöglichen es Maschinen und Robotern, beispielsweise die Temperatur zu messen, quasi zu fühlen.