



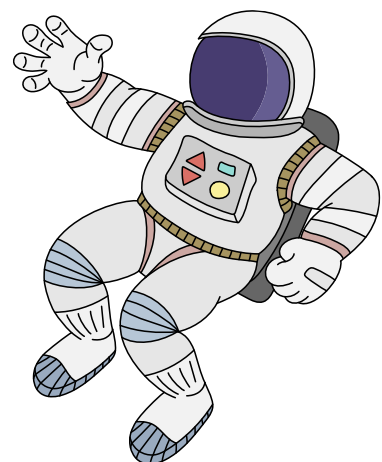
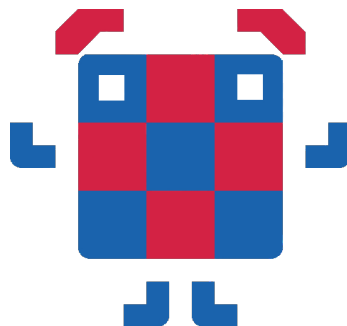
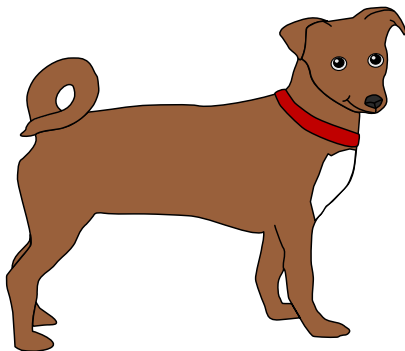
5. – 7. Klasse

Alle Schulformen

Informatik erleben

– Teil 1 –

Hier gibt es Unterrichtsverlaufspläne, Arbeitsblätter,
Kopiervorlagen und Programmieraufgaben für den
Einstieg in die Welt der Algorithmen und der Informatik.



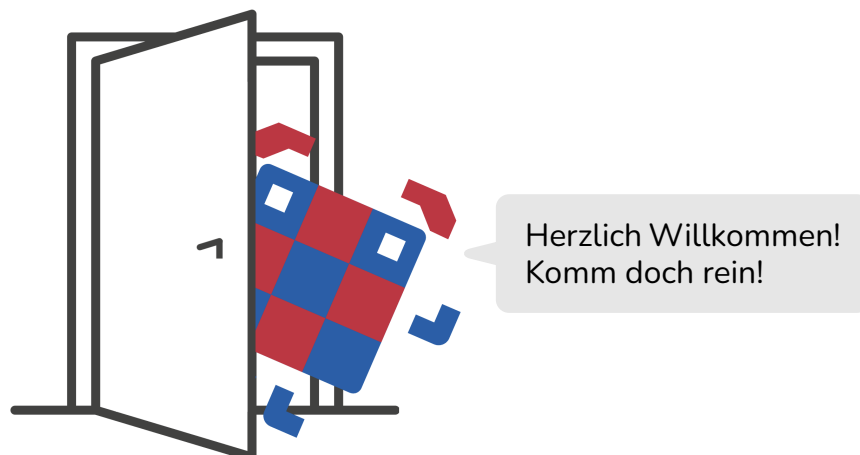
Herzlich Willkommen zu unserer Lernreihe

Wie schön, dass Du da bist! 😊 Mit dieser Unterrichtsreihe bekommst Du alles, was Du brauchst, um den Einstieg in die Programmierung mit Deiner Lerngruppe sorglos zu gestalten an die Hand. Mithilfe der für den Unterricht konzipierten Lernsoftware **Cubi** kannst Du das Thema **Programmierung** kleinschrittig und ganzheitlich mit Deiner Klasse entdecken.

Das IT4Kids-Material zu **Schleifen**, **Verzweigungen**, **Variablen** und Co. vermittelt die grundlegenden Programmierkenntnisse, um das Informatik-Thema **Algorithmen** vollständig zu behandeln.

Keine Sorge: Es wird kein Vorwissen benötigt. Durch unsere Materialien kannst Du Dir die Welt der Programmierung Schritt für Schritt erschließen. Mithilfe vorgefertigter Programmieraufgaben für die Schüler*innen und ausgearbeiteter Unterrichtsverlaufspläne für Dich als Lehrkraft, wollen wir Dir so viel Unterrichtsvorbereitung abnehmen wie möglich. Dazu stellen wir Dir auch Arbeitsblätter, Kopiervorlagen und Musterlösungen zur Verfügung.

Du möchtest Dich erst einmal mit unserer Lernsoftware vertraut machen? Kein Problem! Du findest den Cubi-Editor unter `editor.i4k.org`. Das **Benutzerhandbuch für die Lernsoftware Cubi** verrät Dir alles, was Du bei der Nutzung der Lernsoftware wissen solltest. Du findest es im Begleitmaterial.



Das vorliegende Lehrmaterial von IT4Kids und zugehörige Begleitmaterialien für Schüler*innen stehen, soweit nicht anders angegeben, unter der Creative Commons-Lizenz CC BY-NC-SA 4.0. Weitere Informationen zu der Lizenz findest Du hier: <http://creativecommons.org/licenses/by-nc-sa/4.0/>



Informatik als Fachunterricht in der Sekundarstufe I

In immer mehr Bundesländern erhält das Fach Informatik einen festen Platz im Stundenplan der Jahrgangsstufen 5 bis 7 oder wird dort erprobt. Das Ziel der vorliegenden Lernreihe von IT4Kids ist, Schüler*innen einen ganzheitlichen Einstieg in den **Inhaltsbereich Algorithmen** zu bieten. Dabei vermitteln wir insbesondere die Prozessbereiche **Modellieren und Implementieren, Begründen und Bewerten** und **Kommunizieren und Kooperieren**. Wir folgen hiermit den *Grundsätze[n] und Standards für die Informatik in der Schule* der Gesellschaft für Informatik e.V.

Schon gewusst?

Die Inhalte von IT4Kids entsprechen dem Strategiepapier der KMK für *Bildung in der digitalen Welt* und den Zielen für nachhaltige Entwicklung.

Im Laufe der vorliegenden Unterrichtsstunden lernen die Schüler*innen verschiedene Anweisungen in der grafischen Programmierumgebung **Cubi** kennen. Mit diesen können sie **sequentielle Algorithmen** und Algorithmen mit **Schleifen** und **bedingten Anweisungen** modellieren und implementieren. Im zweiten Teil der Lernreihe kommen **Variablen** und **Funktionen** hinzu. Außerdem wird das große Thema **Fehlersuche und Testen** aufge-

arbeitet und Programme werden mit Stift und Papier geplant. Den Abschluss bildet ein kreatives Projekt, in dem eigene Spiele entwickelt werden.

Die Lernentwicklung der Schüler*innen wird über die gesamte Lernreihe hinweg auch durch **überfachliche Kompetenzen** gefördert. Dadurch, dass sie die Konsumperspektive verlassen und erfahren, wie sie die digitale Welt kreativ mitgestalten können, werden **personale Kompetenzen** gestärkt, die auf die Förderung der Selbstwirksamkeit, -behauptung und -reflexion abzielen.

Auch die **motivationale Einstellung** der Schüler*innen wird mit den Lehrinhalten gesteigert. Die Neugierde der Schüler*innen für den neuen Themenbereich der Informatik wird geweckt, sodass sie sich für diesen begeistern und neuen Problemstellungen ausdauernd begegnen können. Dabei wird eine positive Einstellung gegenüber experimentellem Lernen und die Frustrationstoleranz der Schüler*innen ausgebaut.

Durch eine Varianz an Sozialformen und die Integration von Partner- und Gruppenarbeiten werden **soziale Kompetenzen** wie das Agieren in kooperativen Lernprozessen oder der konstruktive Umgang mit Konflikten und Vielfalt gefördert und gefordert.

Die Schüler*innen erweitern ihre **Methodenkompetenz**, indem sie beim Lernen strukturiert sowie systematisch vorgehen und eigene Arbeitsprozesse planen und organisieren. Das Lösen von Programmieraufgaben fordert ein hohes Maß an Problemlösefähigkeit, das im Verlauf der Lernreihe auf- und ausgebaut wird. Bei der Arbeit an ebendiesen Programmieraufgaben sowie den damit verbundenen Recherche- und Präsentationsaufträgen ist die Förderung der Medienkompetenz der Schüler*innen allgegenwärtig.

Verankerungen von Inhalten zu Algorithmen in Bildungsplänen

Die Inhalte wurden für die verschiedenen Anforderungen der länderspezifischen Bildungspläne entwickelt. Um deren Varianz gerecht zu werden, wurden auch Unterrichtsstunden konzipiert, dessen Kernkompetenzen nur in einzelnen Bundesländern gefordert sind. In der folgenden Tabelle findest Du eine Übersicht über die Unterrichtsstunden. Aus ihr kannst Du entnehmen, welche Unterrichtsstunden im Bildungsplan Deines Bundeslandes verankert sind.

Zuordnung der Unterrichtseinheiten zu den landesspezifischen Bildungsplänen (Stand: Juli 2024)

Bundesland	Unterrichtseinheit						
	Einführung in die Algorithmen	Sequenzen	Schleifen	Verzweigungen	Pseudocode	Fehlersuche & Testen	Eigenes Spiel
Baden-Württemberg		X	X	X		X	
Bayern	X	X	X	X	X	X	X
Berlin/Brandenburg		X	X	X			
Hamburg	X	X	X	X		X	
Hessen	X	X	X	X	X	X	X
Mecklenburg-Vorpommern	X	X	X	X	X	X	X
Niedersachsen	X	X	X	X	X	X	X
Nordrhein-Westfalen	X	X	X	X	X	X	X
Rheinland-Pfalz	X	X	X	X	X	X	X
Saarland	X	X	X	X			X
Sachsen	X	X	X	X	X		X
Schleswig-Holstein	X	X	X	X	X	X	
Thüringen	X	X	X	X	X	X	X

Anmerkung: In Bremen gibt es keinen Informatikunterricht. In Sachsen-Anhalt gibt es Informatik im Wahlpflichtbereich nur in höheren Jahrgangsstufen.

Inhaltsverzeichnis

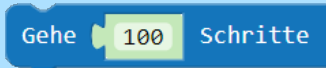
Algorithmen



90 Minuten
unplugged

Seite 6


Sequenzen



90 Minuten
plugged

Seite 14

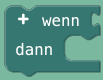
Schleifen



45 Minuten
plugged

Seite 23


Verzweigungen



45 Minuten
plugged

Seite 30

Pseudocode



45 Minuten
unplugged

Seite 35

Fehlersuche & Testen



45 Minuten
plugged


Seite 40


Eigenes Spiel

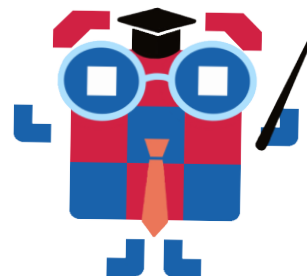


90 Minuten
plugged

Seite 46

 = plugged

 = unplugged





Einführung in die Algorithmen

Mit den beiden Stunden zu **Algorithmen** starten die Schüler*innen in das große Thema der **Programmierung**. Dabei steht im Fokus, einen Bezug zu der Lebenswelt der Kinder aufzubauen und sie für das Thema zu motivieren.

Die beiden Stunden können unabhängig voneinander unterrichtet werden. Dir steht frei, ob Du mit Deiner Klasse nur die erste Stunde durchführst oder Du den Kerngedanken mithilfe der zweiten Stunde vertiefen möchtest.

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 5; Informationstechnologie, Anfangsunterricht; Natur und Technik, Jgs. 7), **Hamburg**¹ (Informatik, Jgs. 7; Naturwissenschaften – Gymnasium/Stadtteilschule, Jgs. 5/6), **Hessen** (Digitale Welt, Jgs. 5), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Saarland** (IKT, S1 – S2), **Sachsen** (Informatik, Jgs. 7), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.
- ... verhalten sich in Konflikten angemessen, verstehen die Sichtweisen anderer und gehen darauf ein.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.
- ... können Informationen sammeln, aufbereiten, bewerten und präsentieren.
- ... setzen sich für Dinge ein, die ihnen wichtig sind und zeigen Einsatz und Initiative.

¹Weitere Rechercheaufträge finden sich in der Stunde **EVA** des zweiten Teils der Lernreihe.

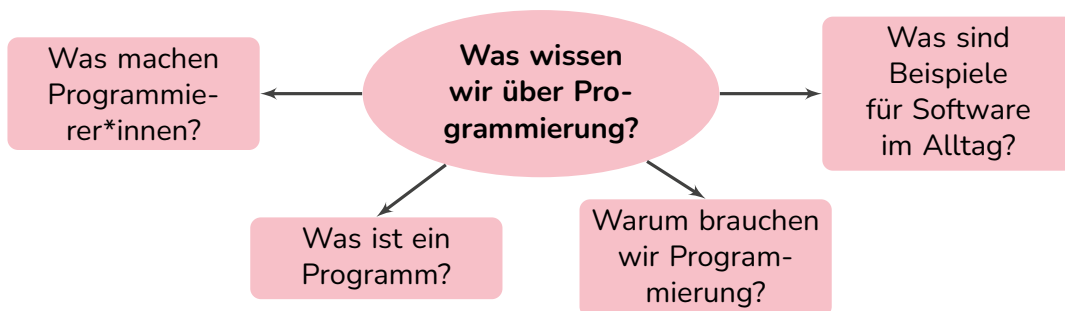
Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... äußern Vermutungen zu informatischen Sachverhalten auf der Basis von Alltagsvorstellungen oder Vorwissen.
- ... formulieren zu Abläufen aus dem Alltag eindeutige Handlungsvorschriften.
- ... führen Algorithmen in ihrer Lebenswelt schrittweise aus.
- ... benennen und beschreiben Handlungsabläufe aus ihrer Lebenswelt, z.B. zur intelligenten Steuerung eines Roboters in Haushalt oder Garten.
- ... erläutern mögliche Auswirkungen des Einsatzes von Informatiksystemen.

Vorbereitung

Mache einen gedanklichen Rundgang durch Deinen eigenen Alltag und frage Dich, wo Du Dinge benutzt, hinter denen eine Software und Programmierung stecken. Präpariere außerdem die Tafel, indem Du die folgende Mindmap auf die Innenseite der Tafel malst und diese zuklappst:



Bereite für das Tafelbild außerdem Zettel für die Schüler*innen vor, auf denen sie ihre Gedanken zu den Fragen festhalten können.

Schau Dir die Kopiervorlage **Kunstgalerie** an und überlege Dir, welche Bilder Du für die Unterrichtsphase verwenden möchtest. Beachte dabei, dass die Zahlen und Buchstaben für den Arbeitsauftrag sehr viel einfacher sind als die Piktogramme. Drucke die Kopiervorlage aus, schneide die Bilder zu und verstau sie in einem Hut oder Sack, aus dem die Schüler*innen blind ziehen können.

Was genau ist Software?

Als Software werden Computerprogramme, Daten und Dokumentationen bezeichnet, die benötigt werden, um ein Computersystem zu benutzen. Software als immaterielles Gut grenzt sich bei einem Computer zu der Hardware ab, die man sehen und anfassen kann, wie z.B. Prozessor, Festplatte oder Bildschirm. Obwohl Software nicht verschleißt, muss sie immer wieder angepasst und erneuert werden, da sie veraltet, wenn sie nicht gepflegt bzw. gewartet wird.

Unterrichtsverlaufsplan – 1. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Beispiele aus dem Alltag sammeln, Motivation zum Programmierenlernen	P	
10	Erarbeitung	Mindmap zum Thema Programmierung erstellen	P	<input type="checkbox"/> Tafelbild <input type="checkbox"/> Zettel <input type="checkbox"/> Klebestreifen/Tafelmagneten
20	Arbeitsphase	Bearbeitung des ABs Sequenz schreiben und malen	EA/- PA	<input type="checkbox"/> AB Sequenz schreiben und malen <input type="checkbox"/> KV Kunstgalerie
10	Präsentation und Reflexion	Museumsrundgang, Erkenntnisse aus Arbeitsphase besprechen	P	<input type="checkbox"/> Ausgefüllte ABs Sequenz schreiben und malen <input type="checkbox"/> KV Kunstgalerie

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Einstieg

Eröffne die Stunde, indem Du der Klasse sagst, dass ihr euch mit Programmierung befassen werdet. Schaffe Transparenz, indem Du den Schüler*innen erklärst, dass ihr euch zu Beginn mit den Begriffen auseinandersetzen werden, bevor ab der dritten Unterrichtsstunde auch mit Tablets, Laptops oder PCs gearbeitet wird. Frage die Schüler*innen, woran sie denken, wenn sie das Wort **Programm** hören und ob ihnen Beispiele aus dem Alltag einfallen. Was denken die Kinder, warum es wichtig ist, Programmieren zu lernen? Mögliche Argumente für das Aufbauen eines Verständnisses darüber, wie ein Computer denkt und wie er als Werkzeug genutzt werden kann, sind:

- Es gibt viele Geräte, die man im Alltag benutzt, die programmiert sind.
- Um ein eigenes Spiel zu programmieren.
- Um einen Roboter programmieren zu können.
- Programmieren macht Spaß.

Sollten den Schüler*innen keine Gründe einfallen, ist das kein Problem, denn in der nächsten Phase wird sich tiefer mit der Allgegenwärtigkeit von Programmen befasst.

Erarbeitung

In dieser Phase wird gemeinsam erarbeitet, was die Schüler*innen bereits über Programmierung wissen. Klappe dazu die Tafel auf und lass ein Kind die Mindmap vorlesen. Stelle den Schüler*innen Zettel zur Verfügung und erteile ihnen den Auftrag, ihre Gedanken zu den Fragen auf die Zettel zu schreiben. Weise



darauf hin, dass sie für jeden Gedanken einen neuen Zettel nehmen. Ausgefüllte Zettel werden mithilfe von Klebestreifen oder Tafelmagneten zu der jeweilige Frage geheftet. Besprich mit den Schüler*innen das dadurch entstandene Tafelbild. Überlegt gemeinsam, welche Zettel sich doppeln oder ähneln und sortiert diese entsprechend. Frage die Schüler*innen welche Schlüsse man aus dem Tafelbild ziehen kann. Es sollte deutlich werden, dass Software ein wichtiger und unumgänglicher Bestandteil des Alltags ist. Deshalb brauchen wir Leute, die mittels Programmierung dafür sorgen, dass Programme geschrieben werden und funktionieren.

Gehe bei der Frage **Was ist ein Programm?** auf den Begriff **Sequenz** ein. Erläutere in diesem Kontext, dass **Programmieren** bedeutet, Anweisungen an einen Computer in der entsprechenden Computersprache zu geben. Ein Computerprogramm kann mit einem Kochrezept, einer Bauanleitung für Lego oder mit Musiknoten verglichen werden, denn auch hier wird eine Reihe präziser Anweisungen gegeben, damit das Rezept, die Anleitung oder das Musikstück umgesetzt werden kann. In der Programmierung nennt man eine solche Reihe von Anweisungen eine Sequenz.

Arbeitsphase

In der Arbeitsphase arbeiten die Schüler*innen zunächst in Einzelarbeit. Stelle dazu als erstes das Arbeitsblatt **Sequenz schreiben und malen** vor und lass die Schüler*innen aus einem Hut oder Sack jeweils einen Zettel der Kopiervorlage **Kunstgalerie** ziehen. Das Bild auf dem Zettel ist geheim. Die Schüler*innen schreiben auf der linken Seite des Arbeitsblatts **Sequenz schreiben und malen** eine Anleitung zu ihrem geheimen Bild.



Kinder, die mit dem Arbeitsauftrag fertig sind, gehen mit ihren Sitznachbar*innen in Partnerarbeit. Dabei tauschen die Partnerkinder die Arbeitsblätter **Sequenz schreiben und malen**, sodass sie das Blatt des jeweils anderen haben. Nun versuchen sie die Arbeitsanweisungen zu befolgen und auf der rechten Seite des Arbeitsblatt das geheime Bild des anderen zu malen. Wenn beide Partnerkinder fertig sind, vergleichen sie die Bilder aus der Kopiervorlage mit den Kunstwerken, die durch die Anweisungen entstanden sind. Konnten die geheimen Bilder herausgefunden werden?



Als Differenzierung nach oben kannst Du schnellen Partnerkindern auftragen zu überlegen, was beim Schreiben und Befolgen der Anweisungen gut geklappt hat und was Herausforderungen waren.

Auch kannst Du die Partnerkinder neu durchmischen. Die rechte Seite mit der Lösung wird dann nach hinten weggeknickt und das neue Partnerkind versucht auf einem separaten Zeichenpapier der Anleitung zu folgen. Bevor die Partnerkinder neu durchgemischt werden, kannst Du ihnen Gelegenheit geben, ihre Anweisungen zu verbessern.

Sollte sich die Aufgabenstellung für die Schüler*innen als zu schwierig herausstellen, kannst Du anstelle der Bilder die Buchstaben und Zahlen herausgeben. Diese sind weniger komplex, weshalb es einfacher ist, für sie eine Anleitung zu schreiben.

Präsentation und Reflexion

Für die Präsentation der Ergebnisse kann eine kleine Ausstellung im Klassenraum veranstaltet werden. Dazu werden jeweils die Originalbilder aus der Kopiervorlage gemeinsam mit den Anleitungen und den nachgemalten Bildern ausgelegt. Die Schüler*innen können sich frei im Klassenraum bewegen und die Bilder vergleichen.

Dann werden im Plenum die Erfahrungen und Herausforderungen der Schüler*innen besprochen. Dabei kann auf Schwierigkeiten beim Schreiben und Umsetzen der Anleitungen eingegangen werden.

Das Fazit sollte die Erkenntnis sein, dass das Geben genauer Anweisungen eine Herausforderung darstellt. Gleichzeitig brauchen wir genaue Anweisungen, damit das Ergebnis am Ende den Erwartungen entspricht.

Werft nochmal einen Blick auf das Tafelbild. Konnten viele Beispiele für Programme im Alltag gefunden werden? Hier sollte festgestellt werden, dass Software ein fester Bestandteil unseres alltäglichen Lebens sind.

Beende die Stunde mit einem Ausblick auf die nächste Unterrichtsstunde.

Vorbereitung

Lies Dir die drei Artikel der Kopiervorlage **Artikel zu Robotern** durch und überlege Dir, wie diese Roboter der Gesellschaft helfen und welche Risikofaktoren sie mit sich bringen. Für eine tiefere Auseinandersetzung kannst Du die Quellen, die unter den Artikeln angegeben sind, nachschlagen.

Unterrichtsverlaufsplan – 2. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Unterrichtsgespräch zum Thema Roboter	P	
10	Erarbeitung	Spiel: Klassenlabyrinth	PA	<input type="checkbox"/> Augenbinde <input type="checkbox"/> ggf. Tische/Stühle/Taschen
15	Arbeitsphase	Bearbeitung der Artikel	EA/ GA	<input type="checkbox"/> KV Artikel zu Robotern <input type="checkbox"/> ggf. KV Freitext
10	Sicherung	Arbeitsergebnisse zusammentragen	GA	<input type="checkbox"/> Fertige Notizen aus der Arbeitsphase zu KV Artikel Roboter
5	Abschluss	Zusammenfassung der Ergebnisse	P	

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Einstieg

Leite ein Unterrichtsgespräch an, in dem die Schüler*innen ihr Vorwissen aus der vorangegangenen Unterrichtsstunde aktivieren. Lass sie sich daran erinnern, wo Programmierung im Alltag zu finden ist und was es mit einer Sequenz auf sich hat. Wenn das Tafelbild aus der vorherigen Stunde noch hängt oder Du Fotos gemacht hast, kannst Du darauf verweisen.

Starte mit der Impulsfrage **Was ist ein Roboter?** in das Thema dieser Unterrichtsstunde. Lass die Schüler*innen sich gegenseitig drannehmen, sodass eine Meldekette entsteht.



Erarbeitung

Die Schüler*innen kriegen einen Einblick in die Arbeitsweise von Programmierer*innen mit dem Computer. Dafür lotsen sie sich gegenseitig als Partnerkinder von einem Sitzplatz zu einem anderen. Wenn genügend Zeit und Platz vorhanden ist, kann auch ein

Labyrinth aus Tischen, Stühlen und Rucksäcken gebaut werden, durch das sich die Schüler*innen gegenseitig navigieren.

Dann werden zwei Rollen verteilt:

Programmierkind: Das Programmierkind gibt Anweisungen für das Roboterkind.

Roboterkind: Das andere Kind übernimmt die Rolle des Roboters. Mache deutlich, dass ein Roboter eine Maschine ist, die von einem Computer gesteuert wird. Er hat sozusagen einen Computer im Kopf und kann nicht selber denken. Deshalb werden dem Roboterkind die Augen verbunden. Das Roboterkind hat keine Kenntnis über seinen Standort und soll lediglich die Anweisungen des Programmierkinds befolgen. Wenn keine Augenbinden vorhanden sind, soll das Roboterkind die Augen geschlossen halten. Das Programmierkind kontrolliert, ob das Roboterkind die Augen auch tatsächlich zu hat (wer blinzelt, verliert!).

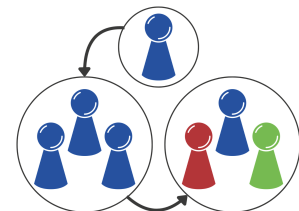
Zum Navigieren können die Programmierkinder folgenden Befehle nutzen:

- Gehe einen Schritt.
- Drehe Dich nach links (um 90 Grad).
- Drehe Dich nach rechts (um 90 Grad).
- Angekommen!

Die Befehle sind bewusst auf ein Minimum reduziert. Sobald das Roboterkind erfolgreich zum Ziel navigiert wurde, tauschen die Partnerkinder ihre Rollen.

Arbeitsphase

In der Arbeitsphase befassen sich die Schüler*innen mit den verschiedenen Einsätzen von Robotern in der Gesellschaft. Dazu werden die Inhalte verschiedener Artikel mithilfe eines Gruppenpuzzels erarbeitet. Die Artikel beziehen sich auf unterschiedliche Roboter und sind in der Kopiervorlage **Artikel zu Robotern** zu finden.



Im Gruppenpuzzle bekommt jedes Kind einen Artikel, den es zunächst in Einzelarbeit liest. Die Schüler*innen können sich hier bereits wichtige Teile im Text markieren.

Anschließend treffen sich die Schüler*innen in Gruppen, in denen alle den gleichen Text bearbeitet haben. Hier werden folgende Fragestellungen diskutiert und bearbeitet:

- Was sind die Vorteile und Chancen, die der Roboter aus Deinem Text mit sich bringt?
- Wo siehst Du potentielle Nachteile und Risiken bei dem Roboter aus Deinem Text?

Gehe durch die Klasse und stelle sicher, dass alle Gruppen mit dem Text und den Fragestellungen zurecht kommen. Weise sie gegebenenfalls darauf hin, dass sie versuchen können, in den Quellenangaben nach den Titeln der Originalquellen zu suchen. Manchmal finden sich hier Hinweise auf Chancen und Risiken zu ihren Robotern.

Sicherung

Die Sicherungsphase findet im letzten Teil des Gruppenpuzzels statt. Dabei werden die Gruppen neu gemischt, sodass in jeder Gruppe jeder Artikel mindestens einmal vertreten ist. Hier stellen sich die Schüler*innen die Arbeitsergebnisse aus der ersten Gruppenarbeit gegenseitig vor und machen sich Notizen zu den Robotern aus den anderen Artikeln. Achte darauf, dass die Schüler*innen Fakten aus dem Text und eigene Meinungen (beispielsweise zu Chancen und Risiken) ihren Mitschüler*innen gegenüber deutlich kommunizieren.

Abschluss

Schließe die Stunde mit einem Unterrichtsgespräch ab, in dem Du mit den Schüler*innen auf die Fragen **Wie werden Roboter im Alltag und der Gesellschaft genutzt?** und **Wie würde ich einen Roboter in meinem Alltag nutzen?** eingehst.

Roboter haben in der Gesellschaft auf verschiedenen Ebenen immer mehr Relevanz. Sie werden für alltägliche Dinge wie den Haushalt genutzt, aber auch für zukunftsorientierte Themen wie die Weltraumforschung. Frage die Schüler*innen außerdem, was für sie neu war und welche Chancen und Risiken sie in der Nutzung von Robotern sehen.

Die Roboter aus den Artikeln sind hilfreich und bergen zu gleich Risiken für uns. So ist ein Staubsaugroboter beispielsweise im Alltag sehr praktisch, da er eine Arbeit im Haushalt abnimmt. Allerdings kam es bereits vor, dass die Roboter die Informationen, die sie in unseren vier Wänden über uns sammeln, an ihre Hersteller weitergeben.

Das Da-Vinci-Operationssystem hingegen erleichtert vielen Chirurg*innen ihre Arbeit. Allerdings sind sie sehr kostspielig. Da die Instrumente immer wieder erneuert werden müssen, handelt es sich hierbei nicht um eine Einmalanschaffung, sondern um laufende Kosten, die gedeckt werden müssen.

Am kontroversten sind allerdings die unbemannten Erkundungsroboter. Sie lassen uns in gefährlichen oder weit entfernten Gebieten forschen und ermöglichen Rettungen aus Krisengebieten. Auf der anderen Seite werden sie auch oft im Krieg eingesetzt, um in fremden Gebieten großen Schaden anzurichten. Beurteile selber, wie tief Du mit Deiner Lerngruppe auf die verschiedenen Risiken und Einsatzmöglichkeiten eingehen möchtest. Beende die Stunde mit einem Ausblick auf das nächste Thema. Frage die Schüler*innen im Zuge dessen, worauf sie sich in der Unterrichtsreihe besonders freuen oder wo sie noch skeptisch gegenüber eingestellt sind.



Sequenzen

Bisher haben sich die Schüler*innen nur analog mit dem Thema **Programmierung** auseinandergesetzt. Mit den beiden Stunden zum Thema **Sequenzen** wird den Schüler*innen ein Einstieg in die grafische Programmierung am Tablet oder Computer geboten. Dazu wird die Lernsoftware **Cubi** genutzt. In diesen beiden Stunden werden die unterschiedlichen Lernstände der Schüler*innen ermittelt und synchronisiert. Dabei gilt es herauszufinden, wer schonmal auf welchem Niveau programmiert hat und für wen das Thema noch ganz neu ist.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 5; Informationstechnologie, Anfangsunterricht; Natur und Technik, Jgs. 7), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7; Naturwissenschaften – Gymnasium/Stadtteilschule, Jgs. 5/6), **Hessen** (Digitale Welt, Jgs. 5; Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern**² (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Saarland** (IKT, S1 – S2; Medienbildung und informatische Bildung, Jgs. 5/6), **Sachsen** (Informatik, Jgs. 7), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... sind motiviert, Neues zu lernen und Dinge zu verstehen, strengen sich an, um sich zu verbessern.
- ... gehen beim Lernen strukturiert und systematisch vor, planen und organisieren eigene Arbeitsprozesse.
- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.
- ... verhalten sich in Konflikten angemessen, verstehen die Sichtweisen anderer und gehen darauf ein.
- ... setzen sich für Dinge ein, die ihnen wichtig sind, zeigen Einsatz und Initiative.

²Je nach Vorwissen der Schüler*innen kann hier nur die erste oder nur die zweite Stunde unterrichtet werden.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... führen einen gegebenen Algorithmus aus.
- ... interpretieren sequenzielle Abläufe.
- ... nutzen eine grafische Programmierumgebung, um ein Programm in einer blockbasierten/grafischen Programmiersprache zu erstellen.
- ... planen und verwenden eine strukturierte, algorithmische Sequenz zur Lösung eines Problems.

Neue Bausteine

- Start: **Wenn Start geklickt wurde**
Bewegung: **Gehe ... Schritte, Drehe links/rechts um ... Grad**
Aussehen: **Sage „ ... “ für ... Sekunden**
Kontrolle: **Warte ... Sekunden**

Vorwissen

Es wird kein Vorwissen benötigt. Ein grundlegendes algorithmisches Verständnis wird jedoch empfohlen.

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben. Löse zudem die Level **Esuri und der Apfel**, **Esuri und die Birne** und **Iruse und die Erdbeere** einmal selber. Du findest sie, indem Du oben links im **Menü** ☰ auf **Öffnen** 📁 gehst und sie anschließend unter dem Tab **Entwicklerreihe** auswählst oder die QR-Codes am Ende des Einstiegs scannst.

Unterrichtsverlaufsplan – 1. Stunde



Zeit	Phase	Unterrichtsschritte	SF	Material
10	Einstieg	Einführung der Lernsoftware Cubi mit dem Level Esuri und der Apfel	P	<input type="checkbox"/> Präsentationstechnik
5	Erarbeitung	Spiel: Winkelwettbewerb*	P	
20	Arbeitsphase	Bearbeitung der Level Esuri und der Apfel , Esuri und die Birne , Iruse und die Erdbeere , ggf. Labyrinth	EA/ PA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> ggf. KV QR-Codes Sequenzen
10	Präsentation und Reflexion	Vorstellen der Arbeitsergebnisse, Thematisierung von Herausforderungen und Erkenntnissen	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik



*Je nach Lerngruppe kannst Du die Erarbeitungsphase und den Einstieg tauschen. Dann programmieren die Schüler*innen direkt im Anschluss ans Kennenlernen der Lernsoftware **Cubi** ihr erstes Programm.

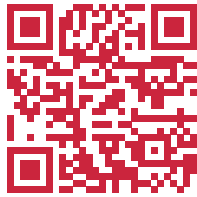
EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum, S = Sitzkreis, SF = Sozialform

Einstieg

Eröffne die Stunde, indem Du den Schüler*innen sagst, dass ihr heute damit anfangen werdet, am Computer oder Tablet zu programmieren. Dazu nutzt ihr die Lernsoftware **Cubi**.

Öffne Cubi über die digitale Tafel oder andere Präsentationstechnik, indem Du im Browser die Seite `editor.i4k.org` eingibst. Gehe nun oben links auf **Menü**  und anschließend auf **Öffnen** . Es erscheint eine Liste an Leveln. Klicke links in der Leiste auf **Entwicklerreihe**, wähle das Level **Esuri und der Apfel** und öffne es.

Zeige den Schüler*innen, dass hier programmiert werden kann, indem sie Bausteine aus dem Werkzeugkasten links auf der Programmierfläche in der Mitte zu einem Programm zusammenpuzzeln. Nimm dazu den **Startbaustein Wenn Start gedrückt wurde** und den **Bewegungsbaustein Gehe 100 Schritte**. Zeige den Schüler*innen, dass sie oben in der grauen Bedienleiste auf den Startknopf  drücken müssen, damit sie auf der Bühne sehen, was ihr Programm macht. Weise sie darauf hin, dass sie ihr Programm unbedingt zurücksetzen  müssen, bevor sie es nochmal testen. Mit dem Programm aus dem **Start-** und dem **Bewegungsbaustein** werdet ihr feststellen, dass Raupe Esuri noch nicht bis zum Apfel kommt. In der Arbeitsphase sollen die Schüler*innen herausfinden, was noch programmiert werden muss, damit sie ihr Ziel erreicht.



Levelvorlage:

level.i4k.org/esuri_apfel_sek



Levellösung:

level.i4k.org/esuri_apfel_sek_lsg

Erarbeitung

Spieler mit den Schüler*innen ein kleines Warm-Up-Spiel zum Thema **Drehungen**. Dazu sind alle Kinder an ihren Plätzen. Dann startet das Spiel **Winkelwettbewerb**. Es gibt folgende Befehle, die die Schüler*innen auf Zuruf ausführen sollen:

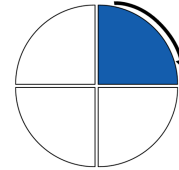
- Mache eine Drehung um 90° nach rechts.
- Mache eine Drehung um 90° nach links.
- Mache eine Drehung um 180° .
- Mache eine Drehung um 360° .

Sind den Schüler*innen Drehungen um 90° , 180° und 360° bereits vertraut? Als Hilfestellung kannst Du sie an der Tafel visualisieren.

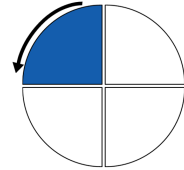
Veranstaltet einen Wettbewerb nach dem Prinzip von **Kommando Pimperle**. Dieses Spiel ist auch unter den Namen **Edgar sagt** oder **Simon Says** bekannt.

- Zu Beginn stehen alle Kinder hinter ihren Stühlen mit dem Gesicht zur Tafel.
- Die Schüler*innen führen die Befehle auf Zuruf aus.
- Die Schüler*innen führen einen Befehl nicht aus, wenn vor dem Befehl kein **Kommando** gerufen wurde, weil der Computer nur die Bausteine ausführt, die mit einem **Startbaustein** verbunden sind.
- Wenn beispielsweise anstatt **Kommando: Mache eine Drehung um 360°** nur der Befehl **Kommando: 360°** kommt, darf man sich nicht bewegen, weil auch der Computer einen Befehl nur versteht und ausführt, wenn er exakt so formuliert ist wie vereinbart.
- Wenn ein Kind einen Fehler macht, muss es sich hinsetzen. Wer schafft es ins Finale?
- Um die Schwierigkeit zu erhöhen, kannst Du mitmachen und Fehler einbauen.

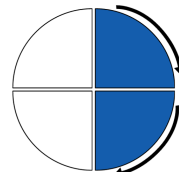
Mache eine Drehung um 90° nach rechts.



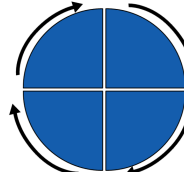
Mache eine Drehung um 90° nach links.



Mache eine Drehung um 180° .



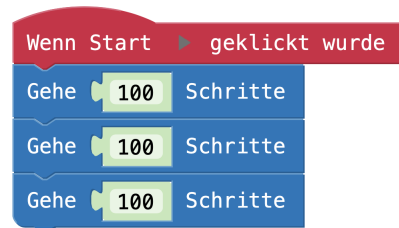
Mache eine Drehung um 360° .




Arbeitsphase

In der Arbeitsphase bearbeiten die Schüler*innen die Level **Esuri und der Apfel**, **Esuri und die Birne** und **Iruse und die Erdbeere**. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Stelle sicher, dass alle Schüler*innen zu dem Level **Esuri und der Apfel** finden. Zum Öffnen des Editors kann das Merkblatt **Der Weg zu Cubi** als Hilfsmittel dienen. Arbeiten die Schüler*innen an Tablets, können sie die QR-Codes einscannen, die sie direkt zu der Vorlage der jeweiligen Level führen. Kopiervorlagen findest Du im Begleitmaterial.

Lass die Schüler*innen mit **Esuri und der Apfel** beginnen. Um Raupe Esuri zum Apfel zu führen und das Level zu lösen, gibt es mehrere Möglichkeiten. Die Schüler*innen können den Baustein **Gehe 100 Schritte** dreimal aneinanderpuzzeln oder die **100** zu **300** ändern, indem sie auf die Zahl klicken und den neuen Wert eingeben.



Schüler*innen, die mit **Esuri und der Apfel** fertig sind, können mit **Esuri und die Birne** weitermachen. Hier wird ein neuer Baustein wichtig: **Drehe nach rechts/links um ... Grad**. Mithilfe dieses Bausteins, kann sich die Raupe drehen. Indem die Kinder auf den runden Pfeil  klicken, können sie die Drehrichtung ändern. Stelle bei Bedarf einen Bezug zu dem Spiel **Winkelwettbewerb** her, bei dem ihr euch bereits mit dem Thema **Drehungen** befasst habt.



Als Differenzierungsaufgabe können die Kinder versuchen, das Level nur mit Recht- oder Linksdrehungen zu lösen. Dafür müssen sie bei einem **Drehe**-Baustein den Wert **270** als Gradzahl eingeben.

Eine andere Idee wäre, den Schüler*innen aufzutragen, den kürzesten Weg zu finden, für den die wenigsten Bausteine benötigt werden. Schaffen sie es, die Raupe diagonal laufen zu lassen?

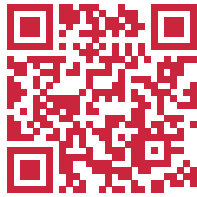
Zuletzt bearbeiten die Schüler*innen das Level **Iruse und die Erdbeere**. Das Level ist anders strukturiert als die bisherigen, da es sich hier um ein Code-Detektiv-Level handelt. Diese Level thematisieren das Thema **Fehlersuche**, das in der Informatik einen sehr hohen Stellenwert hat. Die Aufgabe der Schüler*innen ist herauszufinden, wo der Fehler im Programm liegt, aufgrund dessen es die Raupe nicht zur Erdbeere schafft. Der Fehler im Programm liegt dabei bereits im ersten **Bewegungsbaustein**. Die Raupe geht zu viele Schritte,



sodass sie beim zweiten **Drehe rechts um 90 Grad**-Baustein mit dem Hinterteil an die Ringelblumen stößt. Werden die Schritte auf **250** reduziert, wird das Problem behoben.



Besonders schnelle Schüler*innen können zusätzlich das Level **Labyrinth** bearbeiten. Hier helfen sie der Maus durch das Labyrinth, um erst zum Schlüssel und dann zum Tor zu gelangen.



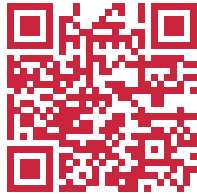
Levelvorlage:

level.i4k.org/esuri_birne_sek



Levellösung:

level.i4k.org/esuri_birne_sek_lsg



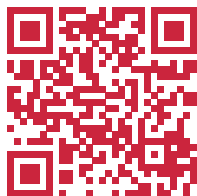
Levelvorlage:

level.i4k.org/cd_iruse_sek



Levellösung:

level.i4k.org/cd_iruse_sek_lsg



Levelvorlage:

level.i4k.org/labyrinth_sek



Levellösung:



level.i4k.org/labyrinth_sek_lsg







Präsentation und Reflexion

Zum Abschluss der Stunde können die Schüler*innen ihre Arbeitsergebnisse vorstellen und darauf eingehen, was gut geklappt hat oder was ihnen schwer gefallen ist. Wenn eine digitale Tafel oder andere Präsentationstechnik vorhanden ist, können einzelne Lösungswege dort in groß gezeigt werden. Mache den Schüler*innen bewusst, dass sie in dieser Stunde einen großen Schritt getan haben: Sie haben ihr erstes Programm geschrieben und können stolz auf sich sein!

Beende die Stunde mit einem Ausblick auf die nächste Unterrichtsstunde.

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben. Mache Dich zudem mit dem Level **Synchronsprechen** vertraut und entwirf einen eigenen Dialog. Du findest das Level, indem Du oben links im **Menü**  auf **Öffnen**  gehst und es anschließend unter dem Tab **Entwicklerreihe** auswählst oder die QR-Codes am Ende des Einstiegs scannst.

Die Schüler*innen entwerfen in der Unterrichtsstunde ein Medienprodukt in Cubi. Ist eine digitale Tafel oder andere Präsentationstechnik vorhanden, bietet es sich an, die Arbeitsergebnisse der Kinder dort in groß zu zeigen. Mache Dich dafür mit der Funktion **online speichern**  vertraut. Öffne dafür das **Menü**  und wähle **speichern**  aus. Es öffnet sich ein Fenster, wo **online speichern**  ausgewählt werden kann. Nun wird ein Codewort generiert. Mithilfe dieses Codeworts kann das Level auf einem anderen Gerät geöffnet werden. Wähle dafür im **Menü**  das Stichwort **Öffnen**  aus und gib das Codewort rechts im entsprechenden Feld ein.

Unterrichtsverlaufsplan – 2. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Einführung von Kostümen und neuen Bausteinen in Cubi	P	<input type="checkbox"/> Präsentationstechnik
10	Erarbeitung	Planung eines Dialogs	PA	<input type="checkbox"/> ABs Unser Dialog
15	Arbeitsphase	Bearbeitung des Levels Synchronsprechen , Üben der Präsentation	PA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> ggf. KV QR-Codes Sequenzen
10	Präsentation und Reflexion	Vorstellen der Szenen, Thematisierung von Herausforderungen und Erkenntnissen	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum, S = Sitzkreis, SF = Sozialform

Einstieg

Eröffne die Stunde, indem Du den Schüler*innen sagst, dass ihr heute mehr Kategorien und Bausteine in Cubi kennenlernen werdet. Stelle einen Bezug zur Lebenswelt der Kinder her, indem Du sie nach Kostümen im Theater oder an Halloween befragst. Was verbinden sie mit Kostümen?

Öffne dann über die digitale Tafel oder andere Präsentationstechnik das Level **Synchronsprechen** in der Lernsoftware **Cubi**. Frage die Klasse, welche Kategorien und Bausteine neu für sie sind. Probiert die neuen Bausteine **Sage „ ... “ für ... Sekunden** und **Warte ...**

Sekunden aus, indem ihr gemeinsam einen kurzen Dialog zwischen den beiden Figuren programmiert. Weise darauf hin, dass Figur B warten muss, wenn Figur A etwas sagt, um ihr nicht ins Wort zu fallen. Hier kann es helfen, nicht erst das Programm von Figur A fertig zu schreiben und dann das Programm von Figur B zu schreiben, sondern immer nur einen **Sage**-Baustein zu ergänzen und dann gleich bei der anderen Figur den entsprechenden **Warte**-Baustein hinzuzufügen. Zeige den Schüler*innen, dass sie zwischen den Figuren wechseln können, indem sie sie auf der Bühne anklicken.

Zeige zudem, dass die Schüler*innen die Optik der Figuren ändern können, indem sie deren Kostüm wechseln. Um ein Kostüm hinzuzufügen, müssen sie unter der Bühne auf **Kostüm hinzufügen** und dann auf **Aus Bibliothek hinzufügen** klicken. Hier können die Schüler*innen aus verschiedenen Grafiken auswählen und so ihr Level nach eigenem Geschmack gestalten. Euer Programm kann schließlich so aussehen:



Schaffe Transparenz, indem Du den Schüler*innen sagst, dass sie heute in Cubi ein kleines Theaterstück programmieren werden. Dieses Medienprodukt kann zum Schluss vorgestellt werden.



Levelvorlage:

level.i4k.org/synchronsprechen_sek



Levellösung:

level.i4k.org/synchronsprechen_sek_lsg

Erarbeitung

Die Schüler*innen bereiten ihr Medienprodukt vor, indem sie die Arbeitsblätter **Unser Dialog** in Partnerarbeit bearbeiten. Sollte es eine Dreiergruppe geben, können sie neben den beiden Figuren noch eine Erzählrolle einbinden.

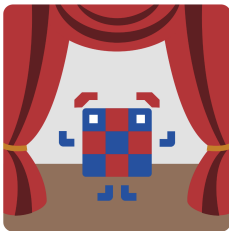
Händige den Schüler*innen zunächst das Arbeitsblatt **Unser Dialog 1** aus. Hier planen sie gemeinsam, um was es in ihrem kleinen Theaterstück gehen wird. Dabei legen sie die Figuren und den groben Inhalt fest. Partnerkinder, die mit dem ersten Arbeitsblatt fertig sind, können sich das Arbeitsblatt **Unser Dialog 2** vorne abholen. Achte darauf, dass die Schüler*innen spätestens bei der Hälfte der Zeit in die Planung des Dialogs auf dem Arbeitsblatt **Unser Dialog 2** übergehen. Weise sie darauf hin, dass sie nicht die ganze Tabelle ausfüllen müssen oder auch auf der Rückseite des Arbeitsblattes weiterschreiben können.

Arbeitsphase

In der Arbeitsphase realisieren die Partnerkinder ihren Dialog in Cubi mittels des Levels **Synchronsprechen**. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Stelle sicher, dass die Schüler*innen zu dem Level **Synchronsprechen** finden. Arbeiten die Schüler*innen an Tablets, können sie den QR-Code einscannen, der sie direkt zu der Vorlage des Levels führt. Eine Kopiervorlage für den QR-Code findest Du im Begleitmaterial.

Weise die Schüler*innen darauf hin, dass sie ihr Produkt am Ende der Stunde vorstellen können. Dabei leihen die Schüler*innen den Figuren aus dem Level ihre Stimme, indem sie die Inhalte der Sprechblasen laut vorlesen. Dazu müssen sie ermitteln, wie lange man braucht, um die Texte laut wiederzugeben, und den Wert in den Bausteinen **Sage „ ... “ für ... Sekunden** und **Warte ... Sekunden** entsprechend anpassen. Es bietet sich also an, Zeit für das Üben des Medienproduktes einzuplanen.

Präsentation und Reflexion



In der Präsentationsphase stellen die Schüler*innen ihre Arbeitsergebnisse vor, indem sie den Dialog nachsprechen während das Level läuft. Geht darauf ein, was gut geklappt hat und was ihnen schwergefallen ist. Ihre Mitschüler*innen können im Anschluss Feedback zu den Produkten geben.

Ist eine digitale Tafel oder ähnliche Präsentationstechnik vorhanden, können die digitalen Theaterstücke dort in groß gezeigt werden. Einen zusätzlichen Effekt hat es, wenn die Bühne in Cubi größer gezogen wird. Dazu kannst Du den Mittelbalken zwischen Programmierfläche und Bühne drücken und beim Gedrückthalten nach links oder rechts ziehen.

Beende die Stunde mit einem Ausblick auf das nächste Thema.



Schleifen

Bisher haben die Schüler*innen jeden Befehl einzeln in der Lernsoftware **Cubi** implementiert. Mit dem Thema **Schleifen** lernen die Kinder, ihre Programme effizienter zu gestalten und Bausteine automatisch zu wiederholen. Im Zuge dessen lernen sie die Vorteile von Schleifen kennen, mit deren Hilfe das Programm lesbarer wird und Flüchtigkeitsfehler vermieden werden können.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 5; Informationstechnologie, Anfangsunterricht; Natur und Technik, Jgs. 7), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7; Naturwissenschaften – Gymnasium/Stadteilschule, Jgs. 5/6), **Hessen** (Digitale Welt, Jgs. 5; Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Saarland** (IKT, S1 – S2; Medienbildung und informatische Bildung, Jgs. 5/6), **Sachsen** (Informatik, Jgs. 7), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... interpretieren Wiederholstrukturen mit einer festen Anzahl von Wiederholungen.
- ... implementieren den algorithmischen Grundbaustein **Schleifen** in der Lernsoftware **Cubi**.
- ... ersetzen eine Abfolge gleicher Sequenzen durch eine Wiederholungsstruktur mit einer festen Anzahl von Wiederholungen.

Neue Bausteine

Schleifen: **Wiederhole ... mal, Wiederhole fortlaufend**

Weitere verwendete Bausteine

Start: **Wenn Start geklickt wurde**

Bewegung: **Gehe ... Schritte, Drehe links/rechts um ... Grad**

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben. Löse zudem die Level **Nataris Kunststück** und **Muschelsuche** einmal selbstständig und mache Dich mit den Leveln **Wellenreiten mit Natari** und **Sternlabyrinth** vertraut. Du findest die Level, indem Du oben links im **Menü** ☰ auf **Öffnen** 📄 gehst und sie anschließend unter dem Tab **Entdeckerreihe** auswählst oder die QR-Codes weiter unten scannst.

Schreibe außerdem die folgende Sequenz an die Innenseite der Tafel und klappe sie zu:

Algorithmus 1 : Tafelbild für die Erarbeitungsphase zum Thema **Schleifen**.

Gehe 100 Schritte

Drehe links um 90 Grad

Gehe 100 Schritte



Drehe rechts um 90 Grad



Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
10	Einstieg	Wiederholung von Programmierung in der Lernsoftware Cubi anhand des Levels Wellenreiten	P	<input type="checkbox"/> Präsentationstechnik
5	Erarbeitung	Erkunden von Schleifen	P	<input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> Tafelbild
25	Arbeitsphase	Bearbeitung der Level Wellenreiten mit Natari , Nataris Kunststück , Muschelsuche , ggf. Natari weicht aus , ggf. Sternlabyrinth	EA/ PA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> ggf. KV QR-Codes Schleifen
5	Präsentation und Reflexion	Thematisierung von Herausforderungen und Erkenntnissen sowie Vorteilen von Schleifen	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik

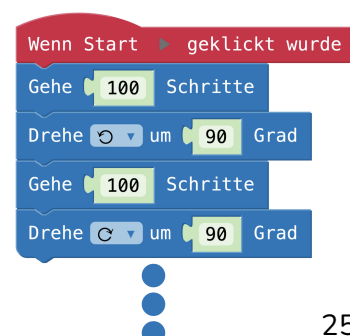
EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

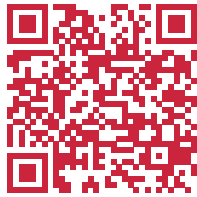
Einstieg

Eröffne die Stunde, indem Du das Level **Wellenreiten mit Natari** über die digitale Tafel oder andere Präsentationstechnik öffnest. Du findest es im **Menü**  und **Öffnen**  unter dem Tab **Entwicklerreihe**. Programmiert nun gemeinsam die dunkle Robbe. Wiederholt dabei folgende Programmierregeln:

- Ein Programm muss mit einem **Startbaustein** beginnen.
- Um ein Programm zu testen, muss man den **Startknopf**  über der Programmierfläche drücken.
- Bevor man weiterprogrammiert muss man sein Programm zurücksetzen .
- In den **Bewegungsbausteinen** können die Werte der Schritt- und Gradzahl sowie die Drehrichtungen geändert werden.

Die dunkle Robbe ist sehr aufwendig zu programmieren, denn sie bekommt jede Bewegungsanweisung einzeln. Lass einzelne Kinder nach vorne kommen und die Programmierbausteine an das Programm anpuzzeln. Navigiert die Robbe entlang der Zick-Zack-Linie. Gibt es Auffälligkeiten im Programm?





Levelvorlage:

level.i4k.org/wellenreiten_sek

Levellösung:

level.i4k.org/wellenreiten_sek_lsg

Erarbeitung

Schaut euch das Programm der dunklen Robbe nochmal genau an. Fällt euch etwas auf? Haben die Schüler*innen eine Idee, wie man das Programm vereinfachen könnte? In dem Programm der dunklen Robbe ist ein Muster zu finden, da die Bausteine ab einer bestimmten Stelle wiederholt werden. Sobald die Schüler*innen das Muster gefunden haben, kannst Du die Tafel öffnen. Die Sequenz, die Du vorab hier notiert hast, sind die Bausteine, die im Programm der dunklen Robbe wiederholt werden.

Frage die Schüler*innen, wie oft die Sequenz wiederholt werden muss, damit die dunkle Robbe ihr Ziel erreicht. Schreibe nach Beantwortung der Frage **Wiederhole 3 mal:** über die Sequenz. Achte dabei darauf, dass die **Bewegungsbausteine** ein wenig eingerückt sind und male eine große Klammer um sie. So wird deutlicher, dass sich die **Bewegungsbausteine** in der Schleife befinden und diese wiederholt werden. Besprich mit den Schüler*innen ebendiese Struktur.

Frage die Schüler*innen, was nun noch fehlt, damit es sich um ein komplettes Programm handelt. Ergänze dann **Wenn Start geklickt wurde** vor der Schleife.

Algorithmus 2 :

Tafelbild am Ende der Erarbeitungsphase zum Thema **Schleifen**.

Wenn Start geklickt wurde

Wiederhole 3 mal:



Gehe 100 Schritte

Drehe links um 90 Grad

Gehe 100 Schritte

Drehe rechts um 90 Grad

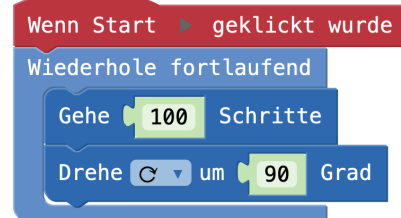
Arbeitsphase

In der Arbeitsphase wenden die Schüler*innen Schleifen in der Lernsoftware **Cubi** an. Dazu programmieren sie zunächst die helle Robbe des Levels **Wellenreiten mit Natari** und bearbeiten anschließend die Level **Nataris Kunststück** und **Muschelsuche**. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Stelle sicher, dass alle Schüler*innen über **Menü** , **Öffnen**  und den Tab **Entwicklerreihe** zu den Leveln finden. Arbeiten sie mit Tablets, können die Kinder den jeweiligen QR-Code einscannen, der sie direkt zu der Vorlage des Levels führt. Eine Kopiervorlage für die QR-Codes findest Du im Begleitmaterial.

Um die helle Robbe im Level **Wellenreiten mit Natari** zu programmieren, müssen die

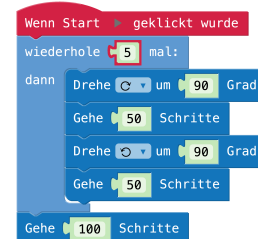
Schüler*innen das Programm, das an der Tafel abgebildet ist, in Cubi umsetzen. Achte darauf, dass alle Schüler*innen die **Bewegungsbausteine** in die Schleife puzzeln und nachvollziehen, dass nur Bausteine innerhalb der Schleife wiederholt werden.

Anschließend bearbeiten die Schüler*innen das Level **Nataris Kunststück**. Hier soll die Robbe wieder und wieder ein Quadrat schwimmen. Dafür lernen die Kinder die Schleife **Wiederhole fortlaufend** kennen. Die Bausteine in dieser Schleife, werden unendlich oft wiederholt. Deshalb ist es auch nicht möglich, unterhalb der Schleife, weitere Bausteine anzupuzzeln.



Schnelle Schüler*innen können die Robbe noch weitere Muster schwimmen lassen. Wie müssen die Werte beispielsweise geändert werden, damit die Robbe ein Dreieck oder Fünfeck schwimmt? Besonders knifflig ist hier, die Robbe einen Kreis schwimmen zu lassen.

Zum Schluss bearbeiten die Schüler*innen ein weiteres Level mit dem Schwerpunkt **Fehlersuche**. Dazu öffnen sie das Level **Muschelsuche**. Hier schwimmt ein Fisch in eine Unterwasserhöhle. Allerdings schwimmt er zum Schluss in den Boden. Das Fehler liegt hier in der Anzahl der Wiederholungen der Schleife. Dort muss der Wert von **6** auf **5** reduziert werden.

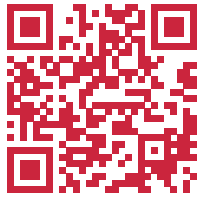


Besonders schnelle Schüler*innen können ein weiteres Fehlersuche-Level bearbeiten. In **Natari weicht aus** will die Robbe um einen Eisberg schwimmen. Allerdings geht auf dem Weg etwas schief, weshalb sie ihr Ziel nicht erreicht.

Der Fehler liegt dabei in dem Baustein **Drehe rechts um 90 Grad** der Kategorie **Bewegung**, der mit in die Schleife eingebettet werden muss. Außerdem muss zum Ende des Programms ein **Gehe 50 Schritte** ergänzt werden.

Auch das Level **Sternlabyrinth** bietet sich gut als Differenzierungsaufgabe an. Hier soll die Maus zum Schlüssel geführt werden. Davor muss sie allerdings noch den Stern, das Herz und die Münze einsammeln. Vorsicht: Dieses Level ist deutlich kniffliger als die bisherigen zum Thema **Schleifen**, da hier zwei Schleifen ineinander verschachtelt werden müssen. Weise die Kinder darauf hin, dass sie versuchen sollen herauszufinden, wo sich Muster in dem Level wiederholen. Um die Maus zu bewegen, wird in diesem Level der Baustein **Gehe vorwärts** aus der Kategorie **Funktionen** benötigt. Bausteine aus dieser Kategorie wurden extra für das jeweiligen Level erstellt und nehmen den Kindern Programmierarbeit ab. Hier müssen die Schüler*innen zum Beispiel nicht herausfinden, wie viele Schritte die Maus vorwärts gehen muss, um sich an keiner Wand zu stoßen.

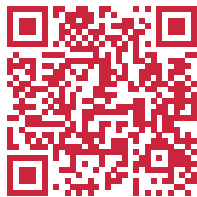




Levelvorlage:
level.i4k.org/kunststueck_sek



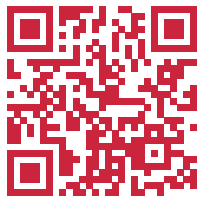
Levellösung:
level.i4k.org/kunststueck_sek_lsg



Levelvorlage:
level.i4k.org/muschelsuche_sek



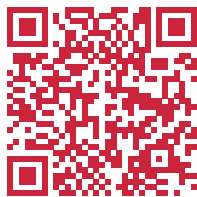
Levellösung:
level.i4k.org/muschelsuche_sek_lsg



Levelvorlage:
level.i4k.org/ausweichen_sek



Levellösung:
level.i4k.org/ausweichen_sek_lsg



Levelvorlage:
level.i4k.org/sternlabyrinth_sek



Levellösung:
level.i4k.org/sternlabyrinth_sek_lsg

Präsentation und Reflexion

Zum Abschluss der Stunde können die Schüler*innen ihre Arbeitsergebnisse vorstellen und darauf eingehen, was gut geklappt hat oder was ihnen schwer gefallen ist. Wenn eine digitale Tafel oder andere Präsentationstechnik vorhanden ist, können einzelne Lösungswege dort in groß gezeigt werden.

Besprecht zudem die Vorteile, die Schleifen in der Programmierung mit sich bringen. Diese umfassen:

- Programme mit Schleifen sind einfacher zu lesen.

- Mit Schleifen kann man schneller programmieren.
- Ein Programm mit einer Schleife kann man sich einfacher merken.
- Schleifen machen das Programm übersichtlicher und verständlicher.
- Schleifen verringern das Risiko, Flüchtigkeitsfehler zu machen.

Beende die Stunde mit einem Ausblick auf das nächste Thema.



Verzweigungen

Bisher haben die Schüler*innen Abläufe in der Lernsoftware **Cubi** implementiert, bei denen jeder Baustein ausgeführt wurde, der mit dem **Startbaustein** verbunden ist, wenn auf **Start** ► geklickt wurde. Mit dem Thema **Verzweigungen** lernen die Schüler*innen, ihre Programme vielfältiger zu gestalten. Hier entscheidet das Programm je nach Situation, welche Befehle ausgeführt werden und welche nicht.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 5; Informationstechnologie, Anfangsunterricht; Natur und Technik, Jgs. 7), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7; Naturwissenschaften – Gymnasium/Stadtteilschule, Jgs. 5/6), **Hessen** (Digitale Welt, Jgs. 5; Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Saarland** (IKT, S1 – S2; Medienbildung und informatische Bildung, Jgs. 5/6), **Sachsen** (Informatik, Jgs. 7), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... sind motiviert, Neues zu lernen und Dinge zu verstehen, strengen sich an, um sich zu verbessern.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.
- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... formulieren Fallunterscheidungen und entscheiden, ob eine Aussage wahr oder falsch ist.
- ... wenden **Verzweigungen** als algorithmische Grundstruktur sicher an.
- ... nutzen Hintereinanderausführung, Fallunterscheidung und Wiederholung in einfachen Programmen.

Neue Bausteine

Kontrolle: **Wenn ... dann ...**, **Wenn ... dann ... sonst**

Fühlen: **berühre ich Farbe ... ?**, **berühre ich ... ?**

Weitere verwendete Bausteine

Aussehen: **Sage „ ... “ für ... Sekunden**

Schleifen: **Wiederhole fortlaufend**

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben. Überlege Dir zwei/drei Beispielsätze aus dem Alltag nach folgendem Muster: **Wenn** es draußen regnet, **dann** braucht man eine Regenjacke. Mache Dich zudem mit dem Level **Farben** vertraut und löse das Level **Training mit Pilu** einmal selbstständig. Du findest die Level, indem Du oben links im **Menü** ☰ auf **Öffnen** 📁 gehst und sie anschließend unter dem Tab **Entwicklerreihe** auswählst oder die QR-Codes weiter unten scanst.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Wenn..., dann... - Sätze bilden	P	
10	Erarbeitung	Einführungen in das Thema Verzweigungen mittels des Levels Farben	P	<input type="checkbox"/> Präsentationstechnik
25	Arbeitsphase	Bearbeitung der Level Farben und Training mit Pilu	EA/ PA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> ggf. KV QR-Codes Verzweigungen
5	Präsentation und Reflexion	Thematisierung von Herausforderungen und Erkenntnissen	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik



EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Einstieg

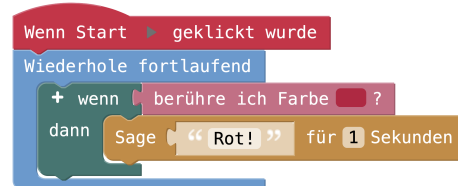
Eröffne die Unterrichtsstunde, indem Du mit den Schüler*innen einen **Wenn..., dann-**Satz formulierst. Starte beispielsweise mit folgender Frage: **Was brauchst du, wenn es draußen regnet?** Eine mögliche Antwort wäre **eine Regenjacke**. Formulieren daraufhin einen vollständigen **Wenn..., dann-**Satz: **Wenn es draußen regnet, dann braucht man eine Regenjacke**. Starte jetzt einen weiteren **Wenn..., dann-**Satz, beispielsweise **Wenn draußen die Sonne scheint...** Lasse den **dann-**Teil des Satzes von einem Kind ergänzen. Dieses Kind kann sich nun einen neuen **Wenn-**Satzanfang ausdenken und ein weiteres Kind drannehmen, das diesen wiederum vervollständigt. Diese Meldekette kann einige Male fortgeführt werden. Achte genau darauf, dass die Schüler*innen immer die Formulierung **Wenn..., dann...** nutzen.



Erarbeitung


Denkt nochmal über die **Wenn..., dann-**Sätze aus dem Einstieg nach. Was hatten all diese Sätze gemeinsam? Abgesehen von dem Satzbau, konnte bei allen Sätzen bestimmt werden, ob die **Wenn-**Teile zutreffen oder nicht. In der Informatik nennt man dies eine Bedingung – also eine Aussage, die wahr oder falsch sein kann. Stelle den Bezug zur Programmierung her, indem Du die Lernsoftware **Cubi** über die digitale Tafel oder andere Präsentationstechnik öffnest. Gehe über das **Menü**  und **Öffnen**  und wähle das Level **Farben** unter dem Tab **Entwicklerreihe** aus.

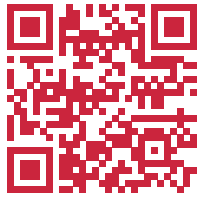
Findet gemeinsam den Baustein **Wenn-Dann** der Kategorie **Kontrolle**. In dem Level wird die Biene programmiert. Sie soll die Farben der Blumen sagen, wenn sie diese berührt. Erstellt nun gemeinsam das Programm für die erste Blume: Schaut euch die Struktur des **Wenn-Dann**-Baustein an.



Was fällt euch auf? Hinter **wenn** kann ein weiterer Baustein angepuzzelt werden. Allerdings muss dieser Baustein eine besondere Form haben. Bausteine mit dieser Form befinden sich in der Kategorie **Fühlen**. Sie sind besonders geformt, da sie als **Bedingungen** fungieren. Die Bausteine im **Wenn-Dann**-Baustein werden nur ausgeführt, wenn die Bedingung wahr ist.

Puzzelt als Bedingung hinter das **Wenn** den Baustein **berühre ich Farbe ...?**. Wählt das **Rot** der Blume aus. Jetzt fehlt noch ein Baustein: der **Sage**-Baustein, um eine Rückmeldung zu geben. Gebt als Text **Rot!** ein. Die **Wiederhole fortlaufend**-Schleife sorgt dafür, dass das Programm immer wieder prüft, welche Farbe die Biene berührt.

Beachte, dass die Biene mit der Maus oder dem Finger zu der Blume gezogen werden muss, damit sie die Farbe sagt. Zeige dies den Schüler*innen, indem Du das Level startest  und die Biene zu der roten Blume ziehst.





Levelvorlage:
level.i4k.org/farben_sek



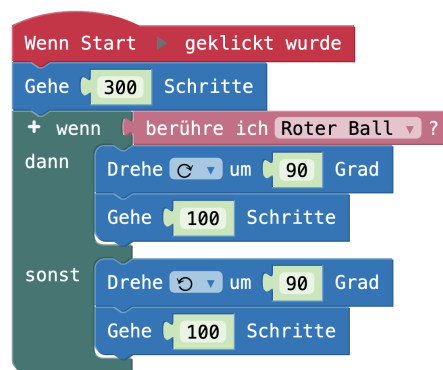
Levellösung:
level.i4k.org/farben_sek_lsg

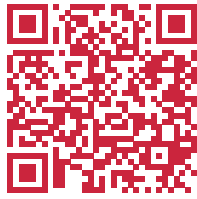
Arbeitsphase

In der Arbeitsphase wenden die Schüler*innen **Verzweigungen** in der Lernsoftware **Cubi** an. Dazu vervollständigen sie das Programm der Biene aus dem Level **Farben** und bearbeiten anschließend die Level **Pilu trifft Entscheidungen**. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Stelle sicher, dass alle Schüler*innen über **Menü** , **Öffnen**  und den Tab **Entwicklerreihe** zu den Leveln finden. Arbeiten sie mit Tablets, können die Kinder den jeweiligen QR-Code einscannen, der sie direkt zu der Vorlage des Levels führt. Eine Kopiervorlage für die QR-Codes findest Du im Begleitmaterial.

Die Schüler*innen beginnen mit dem Level **Farben**. Dafür setzen sie das Programm aus der Erarbeitungsphase selber um und erweitern es um zwei **Wenn-Dann**-Bausteine. Die **Wiederhole fortlaufend**-Schleife umfasst alle drei Verzweigungen.

Schüler*innen, die mit dem Level **Farben** fertig sind, können mit **Pilu trifft Entscheidungen** weiterarbeiten. Hier soll der Hund Pilu, den Ball ins richtige Körbchen bringen. Weise die Schüler*innen darauf hin, dass sie das Level auf zwei verschiedene Weisen lösen können. Sie können ähnlich zum Level **Farben** zwei **Wenn-Dann**-Bausteine untereinander puzzeln oder aber den **Wenn-Dann-Sonst**-Baustein verwenden, den sie im Werkzeugkasten finden. Lass die Kinder das Level über beide Varianten lösen und über die Unterschiede zwischen den Lösungen mit ihren Sitznachbar*innen in Partnerarbeit diskutieren.





Levelvorlage:

level.i4k.org/entscheidung_sek



Levellösung:

level.i4k.org/entscheidung_sek_lsg

Präsentation und Reflexion

Zum Abschluss der Stunde können die Schüler*innen ihre Arbeitsergebnisse vorstellen und darauf eingehen, was gut geklappt hat oder was ihnen schwer gefallen ist. Wenn eine digitale Tafel oder andere Präsentationstechnik vorhanden ist, können einzelne Lösungswege dort in groß gezeigt werden.

Überlegt gemeinsam, was die Programme der heutigen Stunde von denen aus den bisherigen Stunden unterscheidet. In Programmen mit **Wenn-Dann**-Bausteinen können Teile des Programms übersprungen werden. Durch diese **Verzweigungen** handelt es sich um keinen Ablauf mehr, bei dem **alle** Bausteine von oben nach unten ausgeführt werden. Sprecht zudem über den **Wenn-Dann-Sonst**-Baustein. Was unterscheidet ihn vom **Wenn-Dann**-Baustein? Der **Wenn-Dann-Sonst**-Baustein definiert eine Alternative, sollte die **Bedingung** hinter **Wenn** nicht erfüllt sein. Egal welcher Teil des Programms ausgeführt wird – der hinter **dann** oder der hinter **sonst** – danach wird der Programmteil hinter der Verzweigung ausgeführt. Bei dem **Wenn-Dann**-Baustein hingegen verlässt das Programm direkt die Verzweigung, wenn die **Bedingung** nicht erfüllt wurde.

Beende die Stunde mit einem Ausblick auf das nächste Thema.



Pseudocode

Künstler*innen fertigen eine Skizze an, bevor sie ein Bild malen, und Journalist*innen machen sich Stichpunkte, ehe sie einen Artikel schreiben. Auch in der Programmierung ist eine solche Planung des Programms möglich. Dabei wird häufig Pseudocode verwendet. In dieser Unterrichtsstunde lernen die Schüler*innen Pseudocode kennen und schreiben eigenen Pseudocode in der deutschen Variante der Programmiersprache **Pascal**.

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 5/6), **Hessen** (Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Sachsen** (Informatik, Jgs. 8), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... sind motiviert, Neues zu lernen und Dinge zu verstehen, strengen sich an, um sich zu verbessern.
- ... gehen beim Lernen strukturiert und systematisch vor.
- ... entwickeln eine eigene Meinung, treffen eigene Entscheidungen und vertreten diese gegenüber anderen.
- ... zeigen Toleranz und Respekt gegenüber anderen und gehen angemessen mit Widersprüchen um.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... beschreiben die Idee eines gegebenen Algorithmus.
- ... stellen Algorithmen in verschiedenen Darstellungsformen dar.
- ... skizzieren die Idee eines Programms mithilfe von Pseudocode (deutsches Pascal).

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Weitere verwendete Bausteine

Schleifen: **Wiederhole fortlaufend**

Kontrolle: **Wenn ... dann ... sonst**

Vorbereitung

Mache Dich mit der Thematik **Pseudocode** vertraut. Dazu kannst Du beispielsweise das *Skript des Studienvorkurs Informatik der Universität Osnabrück*³ hinzuziehen.

Schreibe den folgenden Pseudocode zum Thema **Waffeln backen** an die Innenseite der Tafel und klappe sie zu:

Algorithmus 3 : Tafelbild für die Erarbeitungsphase zum Thema Pseudocode.

Starte das Programm

Heize Waffeleisen vor

Rühre Teig an

Wiederhole fortlaufend

wenn *Ist noch Teig übrig?* **dann**

 Fette Waffeleisen ein

 Fülle Teig in Waffeleisen

 Warte bis Waffel fertig

 Nimm Waffel heraus

sonst

 Schalte das Waffeleisen aus

 Serviere die Waffeln

 Iss die Waffeln

 Beende das Programm

Versuche für Dich einen Weg zu finden, wie Du die verschiedenen Befehle an der Tafel voneinander abgrenzen kannst. Was hier durch **fett** oder *kursiv* hervorgehoben ist, kannst Du an der Tafel beispielsweise durch Unterstreichen, Kästen oder Schreibschrift vom Rest des Pseudocodes abheben.

³Quelle: Brinkmeier; Haldenwang (2018/19): *Skript. Studienvorkurs Informatik*. Universität Osnabrück: <https://www.informatik.uni-osnabrueck.de/fileadmin/documents/Arbeitsgruppen/Didaktik/Vorkurs/Skript.pdf> (zuletzt aufgerufen am 29.05.2024).

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Beispiele für Skizzierungen von Projekten	P	
10	Erarbeitung	Betrachtung eines Pseudocodes	P	<input type="checkbox"/> Tafelbild
20	Arbeitsphase	Bearbeitung AB Bens Tag	EA/ PA	<input type="checkbox"/> AB Bens Tag <input type="checkbox"/> ggf. KV Freitext
10	Reflexion	Thematisierung von Herausforderungen und Erkenntnissen	P	

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Einstieg

Überlegt gemeinsam, wie man ein Programm planen und skizzieren kann. Führt euch weitere Beispiele für Planungsphasen vor Augen. Beispielsweise machen sich Journalist*innen Stichpunkte, bevor sie einen Artikel schreiben, und Künstler*innen fertigen vor ihrem Kunstwerk eine Skizze an.

Stellt euch die Frage, wie Stichpunkte oder Skizzen für ein Programm aussehen könnten.

Erarbeitung

Klappe nun die Tafel auf und zeige den Schüler*innen den Pseudocode, den Du im Vorfeld dort notiert hast. Lass ein Kind den Code vorlesen. Schreibe **Pseudocode** als Überschrift über das Tafelbild. Frage die Schüler*innen, was sie glauben, was es mit **Pseudocode** auf sich hat. Kennen sie noch weitere Begriffe mit dem Präfix **Pseudo**? Mit welchem Wörtern aus der Alltagssprache kann man es übersetzen? **Pseudo** bedeutet soviel wie **scheinbar** oder **unecht**.

Interpretiert den Pseudocode gemeinsam und formuliert ihn in vollständigen Sätzen aus. Besprecht die Struktur des Pseudocodes. Sprecht über folgende Auffälligkeiten:

- Anweisungen, die in eine andere Anweisung eingebettet sind, werden eingerückt.
- Bedingungen werden in Klammern geschrieben.

Markiert die Auffälligkeiten im Pseudocode.

Arbeitsphase

In der Arbeitsphase bearbeiten die Schüler*innen das Arbeitsblatt **Bens Tag**. Dazu lesen sie den Text durch und markieren sich zunächst alle Aussagen, die für den Pseudocode informativ sind. Danach schreiben sie den Pseudocode zu dem Text auf. Dieser kann wie in Algorithmus 4 aussehen.

Algorithmus 4 : Beispiellösung für das Arbeitsblatt **Bens Tag**.

```

Starte das Programm
Ziehe Badesachen an
wenn Ist es ein kalter Tag? dann
  | Ziehe Bademantel an
Wiederhole fortlaufend
  | Gehe um Beckenrand herum
  | wenn Springt jemand vom Beckenrand? oder Rennt jemand am Beckenrand?
  | dann
  | | Pfeifen
  | | Person ermahnen
  | wenn Gibt es einen Notfall? dann
  | | Ins Wasser springen
  | | Zur Person schwimmen
  | | Person aus Wasser ziehen
  | | wenn Ist die Person ansprechbar? dann
  | | | Person beruhigen
  | | | Person fragen, was sie braucht
  | | sonst
  | | | Notruf wählen
  | | | Erste Hilfe leisten
  | wenn Ist Feierabend? dann
  | | Ziehe Alltagskleidung an
  | | Kollegen verabschieden
  | | Nach Hause gehen
  | | Beende das Programm
  
```

Lass die Schüler*innen, die einen Pseudocode zum Arbeitsblatt **Bens Tag** verfasst haben, in Partnerarbeit gehen. Dort vergleichen sie ihre Arbeitsergebnisse. Dabei werden sie wahrscheinlich feststellen, dass ihr Pseudocode nicht identisch ist, da sie beispielsweise unterschiedliche Formulierungen gewählt haben. Animiere die Schüler*innen dazu, Vermutungen aufzustellen, wie es zu den Unterschieden kam, und was das für die Umsetzung eines Pseudocodes in ein richtiges Programm bedeuten kann.



Schnelle Kinder können als Vertiefung überlegen, ob sie das Programm in der Lernsoftware **Cubi** umsetzen könnten. Die Umsetzung ist mit dem derzeitigen Wissensstand der Schüler*innen schwierig, da der Befehlssatz in Cubi begrenzt ist. Auch kannst Du zwei Schüler*innen mit unterschiedlichen Arbeitsergebnissen ihre Pseudocodes an die Tafel schreiben lassen, sodass ihr sie in der Reflexionsphase im Plenum vergleichen könnt.

Reflexion

Zum Abschluss der Stunde können die Schüler*innen ihre Arbeitsergebnisse vorstellen und darauf eingehen, was gut geklappt hat oder was ihnen schwer gefallen ist. Wenn ihr zwei Pseudocodes an die Tafel geschrieben habt, stellt diese in den direkten Vergleich. Welche Unterschiede gibt es in den Pseudocodes der Kinder? Was sind ihre Vermutungen, wie diese zustande kamen, und welche Herausforderungen bringen sie mit sich? Als Fazit kann gezogen werden, dass Pseudocode helfen kann, Gedanken zu sortieren und zu visualisieren. Es ist allerdings kein Garant dafür, dass der Code in der Programmiersprache auch umgesetzt werden kann. Beende die Stunde mit einem Ausblick auf das nächste Thema.



Fehlersuche und Testen

Fehlersuchen im Quellcode – oder auf Englisch **Debugging** – ist eine der Hauptherausforderungen von Programmierer*innen: Warum kompiliert mein Programm nicht? Was bedeutet diese Fehlermeldung? Denn auch Expert*innen schreiben keinen fehlerfreien Code auf Anhieb. Dafür gibt es zu viele Abhängigkeiten zwischen verschiedenen Programmteilen, die es zu beachten gilt. Daher ist es umso wichtiger – auch als Programmierneuling – eine positive Haltung zu Fehlern aufzubauen und diese als Ausgangspunkt für neues Wissen zu sehen, um sich nicht von ihnen demotivieren zu lassen. Mindestens genauso wichtig ist es, sein Programm zu testen, wenn es vermeintlich fehlerfrei läuft: Wie verhält es sich in Grenzfällen? Wurden wirklich alle möglichen Eingaben berücksichtigt?

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 5; Informationstechnologie, Anfangsunterricht), **Hamburg** (Naturwissenschaften – Gymnasium, Jgs. 5/6), **Hessen** (Digitale Welt, Jgs. 5), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... gehen beim Lernen strukturiert und systematisch vor.
- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... überprüfen und bewerten Modelle und Implementierungen hinsichtlich ihrer Korrektheit.
- ... wenden Techniken zur schrittweisen Prüfung der Korrektheit von Algorithmen an.

... stellen in einem einfachen Programm in verschiedenen Situationen fest, ob es sich so verhält, wie es soll und identifizieren die Ursachen von Fehlern.

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Weitere verwendete Bausteine

Schleifen: **Wiederhole fortlaufend**

Kontrolle: **Wenn ... dann ... sonst**

Vorbereitung

Schau Dir die Quizfragen aus der Phase **Sicherung und Reflexion** an und löse sie einmal selber.

Schreibe zudem den folgenden Pseudocode an die Innenseite der Tafel und klappe sie zu:

Algorithmus 5 :

Tafelbild für die Erarbeitungsphase zum Thema **Fehlersuche und Testen**.

Betrete die Rakete
Ziehe Raumanzug an
Gehe ins Cockpit
Schnalle dich an
Frage: „Alles startklar?“
wenn *Leuchtet das grüne Licht?* **dann**
 └ Tief durchatmen
 └ Rakete starten
sonst
 └ Warte 2 Minuten

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
10	Einstieg	Einführung zu Fehler in Programmen	P	<input type="checkbox"/> Präsentationstechnik
10	Erarbeitung	Fehlersuche im Pseudocode zum Text Raketenstart	P	<input type="checkbox"/> Tafelbild <input type="checkbox"/> KV Raketenstart
15	Arbeitsphase	Aufschreiben der Spielregeln zum Level Astronautenflug	EA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> ggf. KV Freitext <input type="checkbox"/> ggf. KV QR-Codes Fehlersuche & Testen
10	Sicherung und Reflexion	Fehlersuche-Quiz, Reflexion über die Nützlichkeit von Fehlern	P	<input type="checkbox"/> Präsentationstechnik

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum, S = Sitzkreis, SF = Sozialform

Einstieg

Eröffne die Stunde mit einem Unterrichtsgespräch. Thematisiert hier, wie die Schüler*innen bisher mit Fehlern in ihren Programmen umgegangen sind. Können sie sich an die Code-Detektiv-Level erinnern, in denen herausgefunden werden musste, warum ein Programm nicht funktioniert?

Betone, dass Fehler in Programmen etwas ganz Alltägliches sind. Programmierer*innen verbringen viel Zeit damit, Fehler in Programmen zu suchen und zu beheben. Sogar in der Raumfahrt kommen Programmierfehler vor. Ein bekanntes Beispiel ist das der ersten **Ariane 5**-Rakete, die kurz nach dem Start abgestürzt und explodiert ist, weil es einen Fehler im Programm gab, den zuvor keiner gesehen hatte. Mehr Informationen findest Du im Artikel *Darum explodierte die Ariane 5-Rakete bei ihrem Jungfernflug* von David Beck des SWR⁴. Zeige den Schüler*innen das Beispiel mithilfe des YouTube-Videos *Vor 20 Jahren: Der Ariane-5-Absturz | SPIEGEL TV*⁵. Weise die Schüler*innen darauf hin, dass 1996, als die Ariane 5 abgestürzt ist und das Video gedreht wurde, die Geräte, an denen Programmier*innen gearbeitet haben, ganz anders aussahen als heute. Sprecht kurz darüber, dass sogar absolute Programmier-Expert*innen, die Raketen ins Weltall schicken, nicht immun gegen Programmierfehler sind. Was hätte den Absturz der Rakete verhindern können? Durch ausführliche Tests hätte auffallen können, dass das Programm der Ariane 4 für das Nachfolgemodell nicht geeignet ist.

⁴Quelle: Beck: *Darum explodierte die Ariane 5-Rakete bei ihrem Jungfernflug*. SWR Wissen: <https://www.swr.de/wissen/beruehmte-technik-fails-start-der-ariane-5-100.html> (zuletzt aufgerufen am 13.06.2024).

⁵Quelle: DER SPIEGEL: *Vor 20 Jahren: Der Ariane-5-Absturz | SPIEGEL TV*. YouTube: <https://www.youtube.com/watch?v=fd0x2K8ZCJs> (zuletzt aufgerufen am 03.06.2024).

Erarbeitung

Öffne die Kopiervorlage **Raketenstart** über die digitale Tafel oder andere Präsentationstechnik. Alternativ kannst Du sie für die Schüler*innen ausdrucken und austeilen. Lest den Text gemeinsam, indem immer ein Kind einen Abschnitt vorliest. Öffne nun die Tafel und zeige den Schüler*innen den Pseudocode, den Du im Vorfeld dort notiert hast. Findet gemeinsam heraus, ob der Pseudocode das widerspiegelt, was im Text steht. Insgesamt gibt es vier Ungereimtheiten zwischen Pseudocode und Text:



1. Der Text sagt, dass es losgehen kann. Im Pseudocode wird das Programm allerdings nicht gestartet.
2. Astrid zieht den Raumanzug an **bevor** sie die Rakete betritt. Die ersten beiden Befehle müssen also getauscht werden.
3. Astrid fragt immer wieder nach, ob sie die Rakete starten kann. Deshalb wird eine **Wiederhole fortlaufend**-Schleife benötigt, die Astrids Frage sowie die **Wenn-Dann**-Verzweigung einklammert. Beachte, dass sich durch die Schleife die Einrückung ändert.
4. Mit der Schleife hört Astrid nie auf zu fragen, auch wenn sie schon längst grünes Licht bekommen hat. Deshalb muss das Programm beendet werden, nachdem die Rakete gestartet wurde.

Passt den Pseudocode dem Text an, sodass die Lösung aus Algorithmus 6 entsteht.



Algorithmus 6 :

Tafelbild am Ende der Erarbeitungsphase zum Thema **Fehlersuche und Testen**.


```

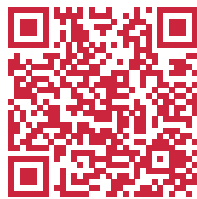
Starte das Programm
Ziehe Raumanzug an
Betrete die Rakete
Gehe ins Cockpit
Schnalle dich an
Wiederhole fortlaufend
  Frage: „Alles startklar?“
  wenn Leuchtet das grüne Licht? dann
    Tief durchatmen
    Rakete starten
  sonst
    Beende das Programm
    sonst
    Warte 2 Minuten
  
```

Arbeitsphase

In der Arbeitsphase führen die Schüler*innen in der Lernsoftware **Cubi** Tests anhand des Levels **Astronautenflug** durch. Dabei sollen sie herausfinden, wann sich welche Figur wie verhält. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über `editor.i4k.org`. Stelle sicher, dass alle Schüler*innen über **Menü** , **Öffnen**  und den Tab **Entwicklerreihe** zu den Levels finden. Arbeiten sie mit Tablets, können die Kinder den QR-Code einscannen, der sie direkt zu der Vorlage des Levels führt. Eine Kopiervorlage für den QR-Codes findest Du im Begleitmaterial.

Ermutige die Schüler*innen das Programm auch zu ändern und so zu erkunden, wie es sich in Randfällen verhält. So können sie beispielsweise die Werte der **Bewegungsbausteine** ändern oder die **Fühlenbausteine** anpassen. Zeige den Schüler*innen,

dass sie die Startposition einer Figur ändern können, indem sie die Figur auf der Bühne auf eine andere Position ziehen und unterhalb der Bühne neben dem Namen der Figur auf die Stecknadel drücken 



Levelvorlage:

`level.i4k.org/astronautenflug_sek`

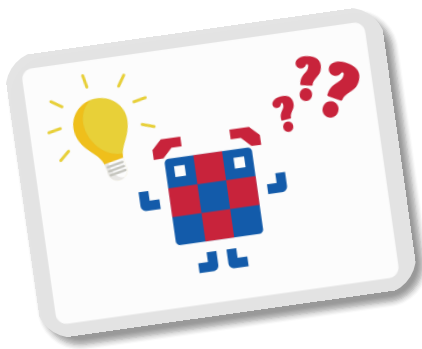
Sicherung und Reflexion

Veranstalte mit den Schüler*innen ein kleines Quiz zu Fehlern in Programmen. Öffne dazu das Level **Astronautenflug** über die digitale Tafel oder andere Präsentationstechnik. Führe nun das Quiz durch, indem Du beispielsweise folgende Fragen stellst:

- Was macht der gelbe Astronaut Anton, wenn er die weiße Astronautin Astrid berührt? – **Er bleibt stehen und sagt: „Hallo!“**
- Was macht Astrid, wenn sie Anton berührt? – **Sie wechselt ganz schnell zwischen den Kostümen hin und her. Das liegt daran, dass das Programm immer wieder überprüft, ob Astrid Anton berührt.**
- Wie kann man dafür sorgen, dass Astrid nicht ganz schnell zwischen den Kostümen wechselt, wenn sie Anton berührt? – **Indem der Baustein **Nächstes Kostüm** zu **Wechsle zu Kostüm Astronaut 2** geändert wird. Alternativ kann nach dem **Nächstes Kostüm** ein **Warte**-Baustein eingefügt werden.**
- Was passiert, wenn Astrid den Planeten berührt? – **Sie dreht sich zwar noch, kann sich aber nicht mehr von dem Planeten wegbewegen. Dies kann man ändern, indem der Baustein **Gleite zu Planet in 1 Sekunden** entfernt wird.**

- Was passiert, wenn man Astrids Startposition ändert, sodass sie in einer Ecke startet? – **Astrid kommt aus der Ecke nicht mehr raus und dreht sich dort im Kreis.**

Zieht ein gemeinsames Schlussfazit, in dem ihr festhaltet, dass Fehler sehr nützlich sind, da durch sie Neues gelernt werden kann. Beende die Stunde mit einem Ausblick auf das nächste Thema.



Eigenes Spiel

In diesem Abschluss des ersten Teils zum Einstieg in die Programmierung programmieren die Schüler*innen ihr eigenes Level. Das gibt ihnen die Möglichkeit, ihr gelerntes Wissen anzuwenden und für eine kreative Aufgabe zu nutzen. Hier werden alle Kompetenzen der vorangegangenen Unterrichtsstunden zusammengeführt. Dabei gehen die Schüler*innen zunächst in eine Planungsphase, um strukturiert an den Entwicklungsprozess herangehen zu können. Während des gesamten Projektes steht die kreative Arbeit im Vordergrund.

Anknüpfung an Bildungspläne

Bayern (Natur und Technik, Jgs. 7), **Hessen** (Digitale Welt, Jgs. 5), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 5), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Rheinland-Pfalz** (IPS, Jgs. 5/6), **Saarland** (IKT, S1 – S2), **Sachsen** (Informatik, Jgs. 8), **Thüringen** (Medienbildung und Informatik, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... schätzen eigene Fähigkeiten realistisch ein und nutzen eigene Potenziale.
- ... haben Vertrauen in die eigenen Fähigkeiten und glauben an die Wirksamkeit des eigenen Handelns.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.
- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.
- ... verhalten sich in Konflikten angemessen, verstehen die Sichtweisen anderer und gehen darauf ein.
- ... zeigen Toleranz und Respekt gegenüber anderen und gehen angemessen mit Widersprüchen um.
- ... bewerten Projektergebnisse nach vorgegebenen Kriterien.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... planen, entwickeln und implementieren einen Algorithmus in einer grafischen Programmiersprache auf experimentelle Weise.
- ... beschreiben die Idee eines selbsterstellten Algorithmus.
- ... interpretieren und kommentieren Algorithmen in einer grafischen Programmierumgebung.

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Vorwissen

In diesen Unterrichtsstunden wird das Wissen aus allen vorherigen Stunden angewendet und damit eigene Level entwickelt.

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben.

Unterrichtsverlaufsplan – 1. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Rückblick	Aktivierung des Vorwissens aus den vergangenen Stunden	P	
15	Erarbeitung	Eigenes Level mit Stift und Papier planen	EA/ PA/ GA	<input type="checkbox"/> ABs Eigenes Level
20	Arbeitsphase	Implementierung der eigenen Level	EA/ PA/ GA	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> ABs Eigenes Level <input type="checkbox"/> ggf. KV QR-Codes Eigenes Level
5	Abschluss	Speichern der Zwischenstände der Level	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> ggf. KV Meine Level

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Einstieg

Eröffne die Stunde, indem Du den Schüler*innen sagst, dass sie in den kommenden zwei Stunden ein eigenes Level entwerfen werden.

Öffne das Level **Eigenes Spiel** über die digitale Tafel oder andere Präsentationstechnik. Was steht den Schüler*innen für die Gestaltung ihres eigenen Levels zur Verfügung? Besprecht in diesem Kontext kurz die Kategorien sowie den Wechsel von Kostümen und Figuren.

Frage die Kinder, wie sie an das Projekt drangehen wollen. Erwähne sie daran, dass es wichtig ist, ein Projekt im Vorfeld zu planen und zu skizzieren, bevor wild drauflos programmiert wird. Folgende Planungswerkzeuge können die Schüler*innen verwenden:

- Einen Pseudocode aufschreiben.
- Eine Geschichte in Stichpunkten notieren.
- Stichpunkte zu den Spielregeln machen.
- Einen Dialog aufschreiben.
- Eine Skizze für die Bühne anfertigen, aus der hervorgeht, wann welche Figur wo stehen soll.

Du kannst den Schüler*innen freistellen, ob sie alleine, zu zweit oder in einer Kleingruppe an dem eigenen Level arbeiten wollen.





Erarbeitung

Die Schüler*innen planen ihr Projekt zunächst mit Stift und Papier. Dafür erstellen sie Stichpunkte, Skizzen und Pseudocodes für ihr Level. Ihre Gedanken protokollieren sie auf den Arbeitsblättern **Eigenes Level**.

Arbeitsphase

In der Arbeitsphase implementieren die Schüler*innen ihre Levelkonzepte in Cubi. Zeige den Schüler*innen, dass man sich in Cubi mithilfe der Kommentarfunktion Notizen machen kann, die das Programm nicht beeinflussen. Die Kinder können dafür über die rechte Maustaste auf dem Computer oder durch langes Drücken auf die Programmierfläche am Tablet ein Menü öffnen und **Kommentar hinzufügen** wählen.




Für die Umsetzung ihrer Ideen nutzen die Schüler*innen die Levelvorlage **Eigenes Spiel**. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Stelle sicher, dass alle Schüler*innen über **Menü** , **Öffnen**  und den Tab **Entwicklerreihe** zu der Levelvorlage finden. Arbeiten sie mit Tablets, können die Kinder den QR-Code einscannen, der sie direkt zum Level führt. Eine Kopiervorlage für den QR-Code findest Du im Begleitmaterial. Gehe durch die Klasse und hilf bei Fragen und Problemen. Mache die Kinder bei Bedarf darauf aufmerksam, dass sie ihr Levelkonzept während der Umsetzung noch anpassen können.



Levelvorlage:

level.i4k.org/eigenes_spiel1_sek

Reflexion

Beende die Unterrichtsstunde, indem Du den Schüler*innen zunächst versicherst, dass sie in der kommenden Stunde mehr Zeit für die Umsetzung ihres Levels haben werden. Dafür ist es allerdings ganz wichtig, dass sie ihr Level online speichern, bevor sie es schließen. Dafür müssen sie im **Menü**  auf **Speichern**  gehen. Nun öffnet sich ein Fenster, in dem sie auf **Online speichern**  klicken müssen. Dadurch kriegen sie ein Codewort, das aus einer Farbe, einer Zahl und einem Tier besteht. Zeige diesen Weg über die digitale Tafel oder andere Präsentationstechnik und achte darauf, dass sich alle Schüler*innen ihr Codewort notieren. Dafür eignet sich die Kopiervorlage **Meine Level**. Mache die Schüler*innen in diesem Kontext darauf aufmerksam, dass das Codewort wie ein Passwort zu ihrem Level ist. Kennen die Kinder die Funktion von Passwörtern? Mithilfe von Passwörtern wird sichergestellt, dass nur bestimmte Leute Zugriff auf die Dateien haben, die durch das Passwort geschützt werden. Für welche Plattformen nutzen die Schüler*innen bereits Passwörter? Wollen sie ihr eigenen Level mit jemandem teilen? An wen würden sie das Codewort rausgeben?

Vorbereitung

Bereite die Stunde vor, indem Du sicherstellst, genug Tablets für die Klasse zur Verfügung oder Zugang zum Computerraum zu haben.

Unterrichtsverlaufsplan – 2. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Rückblick	Aktivierung des Vorwissens aus der vergangenen Stunde	P	
25	Arbeitsphase	Implementierung der eigenen Level	EA/ PA/ GA	<input type="checkbox"/> Codewörter der Level <input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Ausgefüllte ABs Eigenes Level
10	Präsentation und Reflexion	Vorstellen der Level, Thematisierung von Herausforderungen und Erkenntnissen	P	<input type="checkbox"/> Tablets/Laptops/PCs <input type="checkbox"/> Ausgefüllte ABs Eigenes Level
5	Abschluss	Schlussrunde zum Thema Programmieren	P	

EA = Einzelarbeit, GA = Gruppenarbeit, PA = Partnerarbeit, P = Plenum,
S = Sitzkreis, SF = Sozialform

Rückblick

Eröffne die Unterrichtsstunde, indem Du das Vorwissen der Schüler*innen aktivierst und sie fragst, was ihr in der vergangenen Stunde gemacht habt. Dort haben die Kinder eigene Levelideen entwickelt und begonnen, diese in Cubi zu implementieren. Welche Planungswerkzeuge haben die Schüler*innen genutzt, um ihr Level vor der Implementierung zu skizzieren? In dieser Stunde werdet ihr die Level fertig umsetzen und in der Klasse präsentieren. Erwähne die Schüler*innen gegebenenfalls an ihre Vorarbeit auf den Arbeitsblättern **Eigenes Level**.

Arbeitsphase

In der Arbeitsphase arbeiten die Schüler*innen weiter an der Implementierung ihrer Level und stellen diese weitestgehend fertig. Dazu benötigen sie Tablets, Laptops oder Computer, die eine Verbindung zum Internet haben. Sie öffnen den Cubi-Editor über editor.i4k.org. Öffne ebenfalls Cubi über die digitale Tafel oder andere Präsentationstechnik. Zeige den Schüler*innen, wie sie ihr Level mithilfe des Codeworts öffnen können, das sie sich zum Abschluss der vorherigen Stunde notiert haben. Gehe dafür in

das **Menü** ☰ oben links und klicke dort auf **Öffnen** 📁 . Nun können die Schüler*innen in dem Feld oben rechts ihr Codewort eingeben und das Level mittels des Knopfs **Level mit Codewort laden** öffnen.

Präsentation und Reflexion

Für die Präsentation der Ergebnisse kann eine kleine Ausstellung im Klassenraum veranstaltet werden. Dazu werden die Arbeitsblätter **Eigenes Level** sowie die Skizzen, Stichpunkte und Pseudocodes aus der Planungsphase ausgelegt sowie die Level auf den Tablets oder Computern geöffnet. Lass die Schüler*innen frei durch die Klasse laufen, die Level zeigen, betrachten und ausprobieren.

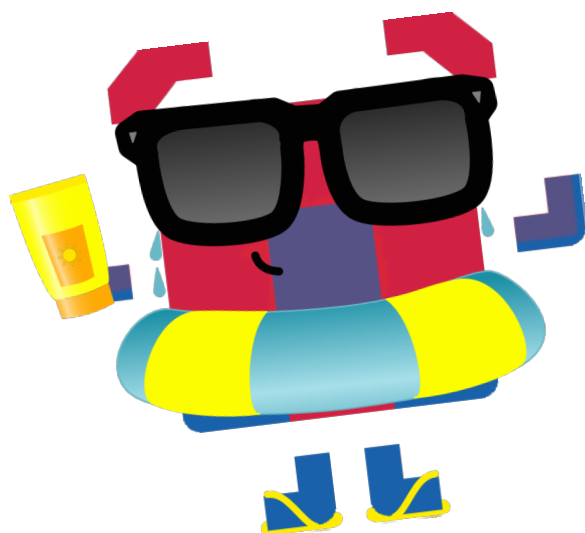
Dann werden im Plenum die Erfahrungen und Herausforderungen der Schüler*innen besprochen. Dabei kann auf Schwierigkeiten beim Planen des Levels und dessen Umsetzung eingegangen werden. Konnten die Schüler*innen ihre Level so umsetzen, wie sie es sich gedacht haben? Welche Anpassungen mussten gemacht werden? Gab es manchmal Uneinigkeiten zwischen den Gruppenmitgliedern? Wie sind die Schüler*innen damit umgegangen?

Abschluss

Werft nochmal einen Blick auf eure Programmierreise. Ihr habt sehr viel gelernt – Wahnsinn! 🤖 Was waren Momente, die euch besonders positiv oder negativ im Kopf geblieben sind? Gibt es Stereotypen, die ihr vor dieser Reise gegenüber Programmierung oder Informatik hattet? Wie ist jetzt eure Perspektive darauf? Wie war es für die Schüler*innen zum Schluss ihre neuen Kenntnisse anzuwenden und mit ihnen kreativ zu werden?

Geschafft!

Großartig, Du hast es durch **Teil 1 der Lernreihe** geschafft! Was eine tolle Leistung!



Jetzt kannst Du Dich zurücklehnen, während Deine Klasse fleißig programmiert.

Baustein-Lexikon

Start

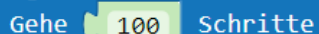
Der erste Baustein eines Blocks aus mehreren Bausteinen ist immer ein **Starbaustein**. Ein Programm einer Figur kann beliebig viele **Startbausteine** haben. **Startbausteine** zeichnen sich durch die Rundung am oberen Teil aus. Diese sagt aus, dass **Startbausteine** nicht an andere Bausteine angehängt werden können.



Wenn Start ▶ geklickt

Der **Startbaustein Wenn Start geklickt wurde** ist der erste Baustein, den die Schüler*innen kennenlernen. Nachfolgende Bausteine werden nacheinander ausgeführt, unmittelbar nachdem das Level gestartet ▶ wurde.

Bewegung



Gehe 100 Schritte

Der Baustein **Gehe ... Schritte** bewegt die Figur die entsprechende Anzahl an Pixel in die aktuelle Richtung der Figur. Im Normalfall ist dies bei Programmstart nach rechts.



Drehe rechts um 90 Grad

Mit dem Baustein **Drehe rechts/links um ... Grad** dreht sich die Figur in die ausgewählte Richtung um die entsprechende Gradzahl. In den ersten Leveln brauchen die Schüler*innen nur den rechten Winkel.

Kontrolle



+ wenn dann

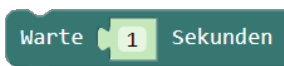
Der Baustein **Wenn dann** leitet eine Verzweigung ein. Oben an das **wenn** wird eine Bedingung angepuzzelt. Diese ist entweder **wahr** oder **falsch**. Wenn die Bedingung wahr ist, werden die Bausteine, die neben dem **dann** stehen ausgeführt. Ist die Bedingung falsch, also nicht erfüllt, werden die Bausteine bei **dann** übersprungen und nicht ausgeführt. Drückt man oben links auf dem Baustein auf das weiße Plus, wird ein neuer Verzweigungsarm hinzugefügt, an den eine

weitere Bedingung angepuzzelt werden kann. Diese wird jedoch nur überprüft, wenn die erste Bedingung falsch war.



Der Baustein **Wenn dann sonst** leitet eine Verzweigung ein. Oben an das **wenn** wird eine Bedingung angepuzzelt. Diese ist entweder **wahr** oder **falsch**. Wenn die Bedingung wahr ist, werden die Bausteine, die neben dem **dann** stehen ausgeführt. Die Bausteine hinter **sonst** werden übersprungen. Ist die Bedingung falsch, also nicht erfüllt, ist es genau andersherum und die Bausteine bei **dann** werden übersprungen und an ihrer Stelle werden die Bausteine, die hinter **sonst** stehen, ausgeführt.

Drückt man oben links auf dem Baustein auf das weiße Plus, wird ein neuer Verzweigungsarm hinzugefügt, an den eine weitere Bedingung angepuzzelt werden kann. Diese wird jedoch nur überprüft, wenn alle vorherigen Bedingungen falsch waren.



Gelangt ein Programm zu einem **Warte**-Baustein, dann bleibt es hier für die Anzahl der eingegebenen Sekunden stehen. Andere Programmteile der Figur, die ihren eigenen **Startbaustein** haben, werden hierdurch nicht unterbrochen. Erst wenn die Zeit um ist, wird der nächste Baustein ausgeführt.

Fühlen

Bausteine der Kategorie **Fühlen** werden als Bedingungen in **Verzweigungen** oder **Schleifen mit Bedingungen** angepuzzelt. Das Programm prüft, ob die Bedingung **wahr** oder **falsch** ist. Ist die Bedingung wahr, werden die Bausteine in der **Verzweigung** oder **Schleife** ausgeführt.



Mit diesem Baustein wird geprüft, ob die Figur eine bestimmte Farbe berührt. Durch Klicken auf das Farbfeld kann die Farbe geändert werden, die geprüft wird.

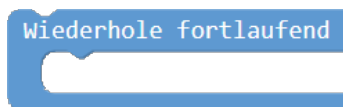


Mit diesem Baustein kann die Figur reagieren, wenn sie etwas berührt. Durch Klicken auf den kleinen Pfeil kann ausgewählt werden, ob die Figur auf den Rand oder eine andere Figur reagieren soll, wenn es noch weitere Figuren in dem Level gibt.

Schleifen



Mit der **Wiederhole ... mal**-Schleife können Bausteine, die in die Schleife eingefügt werden, wiederholt werden. Klicke auf die Zahl, um die Anzahl der Wiederholungen zu ändern.



Alle Bausteine, die in die **Wiederhole fortlaufend**-Schleife eingefügt werden, werden endlos lange wiederholt. Das Level stoppt erst, wenn man es über die Pause-Taste unterbricht oder über den Zurücksetzen-Button stoppt. Unter dieser Schleife können keine Bausteine angehängt werden, weil diese nie ausgeführt würden.

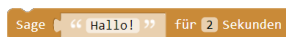
Aussehen



Mithilfe dieses Bausteins wechselt die entsprechende Figur das Kostüm und ändert so ihr Aussehen. Die Kostüme sind unterhalb der Bühne zu sehen. Hier wird zu dem Kostüm gewechselt, welches in dem Feld ausgewählt wurde.



Mithilfe dieses Bausteins wechselt die entsprechende Figur das Kostüm und ändert so ihr Aussehen. Die Kostüme sind unterhalb der Bühne zu sehen. Hier wird zu dem Kostüm gewechselt, welches unter der Bühne nach dem aktuellen Kostüm abgebildet ist.



Wenn der **Sage**-Baustein ausgeführt wird, erscheint für die Figur eine Sprechblase. Dort wird der Text angezeigt, welcher in das Textfeld geschrieben wurde. Nach der angegebenen Zeit verschwindet die Sprechblase und der nächste Baustein wird ausgeführt.

KV Kunstgalerie

IT 4 KIDS

Schreibe eine Anleitung für diese Tür.



Schreibe eine Anleitung für diese Brille.



Schreibe eine Anleitung für

Schreibe eine Anleitung für diesen Menschen.



Schreibe eine Anleitung für diese Sonne.



Schreibe eine Anleitung für

Alle Arbeitsblätter und Kopiervorlagen zu diesem Modul findest Du auf der Webseite von **IT 4 KIDS**: material.i4k.org/ab

Astrid ist Astronautin. Heute steht ihr großer Raketenstart an. Endlich kann der Versuch starten!

Bevor sie die Rakete betritt, schlüpf sie in ihren Raumanzug. Dann geht sie ins Cockpit und schnallt sich an.

Damit Astrid weiß, dass sie die Rakete starten kann, muss ein grünes Licht aufleuchten. Deshalb fragt sie, ob alles startklar ist.

Wenn das grüne Licht nicht erscheint, wartet Astrid zwei Minuten und fragt dann nochmal nach. Das macht sie so lange, bis sie grünes Licht bekommt und starten kann.

Bei grünem Licht atmet sie nochmal tief durch und startet dann die Rakete.

Damit ist ihr Job für den Raketenstart getan.



Dieses Material ist lizenziert unter CC BY-NC-SA 4.0. Weitere Informationen findest Du hier: <http://creativecommons.org/licenses/by-nc-sa/4.0/>

IT 4 KIDS

Namen: _____

Aufgabe: Schreibe eine Anleitung zu deinem Bild. Sei so genau wie möglich, damit ein anderes Kind das Bild malen kann. Das Bild ist gelblich, lass es niemanden sehen!

Stopp! Bevor es weitergeht, tausche das Arbeitsblatt mit deinem Partner. Arbeitet auf der rechten Seite weiter! →

Sequenz schreiben und malen

Dieses Material ist lizenziert unter CC BY-NC-SA 4.0. Weitere Informationen findest Du hier: <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Dieses Material ist lizenziert unter CC BY-NC-SA 4.0. Weitere Informationen findest Du hier: <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Dar habe ich gemalt: _____

Aufgabe: Befolge die Anleitung deines Partners, um das gelbe Bild zu malen.

Datum: _____

IT 4 KIDS

Namen: _____ Datum: _____

Eigenes Level 1

Worum geht es in dem Level?
Thema: _____
Ort: _____
Geschichte: _____

Um wen geht es in dem Level?
Name: _____ Name: _____
Aufgabe: _____ Aufgabe: _____

Steuerbar:
 Nein
 Ja, und zwar so: _____

Aussehen:
 Nein
 Ja, und zwar so: _____

Aussehen: _____ Aussehen: _____

Dieses Material ist lizenziert unter CC BY-NC-SA 4.0. Weitere Informationen findest Du hier: <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Weitere Angebote von IT4Kids

Du willst weiter mit IT4Kids arbeiten? Mach gerne mit **Teil 2 der Levelreihe** weiter! Hier kannst Du gemeinsam mit Deinen Schüler*innen tiefer in die Welt der Algorithmen eintauchen und lernst unter anderem **Variablen**, **verschachtelte Verzweigungen** und **Funktionen** kennen.

Außerdem bieten wir neben einer Modulreihe zu den Themen **Sequenzen**, **Schleifen** und **Verzweigungen** mit der Zielgruppe **Primarstufe** auch Fortbildungen für Dich und Dein Kollegium an. Hier lernen wir gemeinsam den Cubi-Editor kennen, sammeln grundlegende Programmiererfahrungen und planen eine erste Unterrichtsstunde mit Cubi speziell für Deine Klasse. Nach der Fortbildung kannst Du direkt am nächsten Tag eine Stunde Programmierung mit Deiner Klasse ausprobieren, weil wir alles gemeinsam in der Fortbildung vorbereitet haben. Alle aktuellen Informationen zu unserem Fortbildungsangebot findest du auf unserer Webseite unter www.i4k.org/fortbildung. 😊

Du bist noch unschlüssig? Dann schau Dich gerne auf unserer Webseite www.i4k.org um, stöbere durch unser Material und lerne uns ein bisschen besser kennen. Wenn auf dem Weg Fragen aufkommen oder Du mit uns ins Gespräch über die Materialien kommen möchtest, dann melde Dich gerne per E-Mail unter info@it-for-kids.org bei uns oder nimm über unsere Webseite www.i4k.org/kontakt mit uns Kontakt auf. 💬

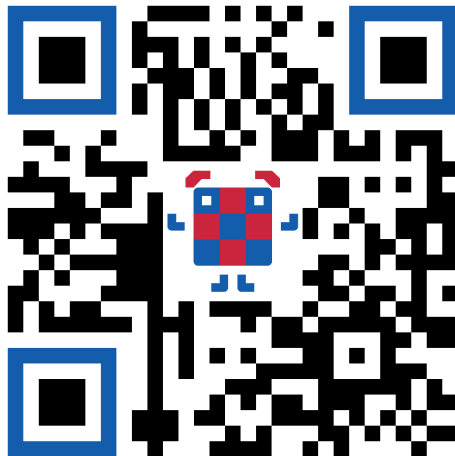
Wir freuen uns auf Dich! 😊



Hilf uns, besser zu werden!

Dir sind Fehler in dem Material aufgefallen?
Du hast Verbesserungsvorschläge?
Du möchtest mehr zum Einstieg in die Programmierung?

Wir freuen uns über Dein Feedback:



feedback.i4k.org/lk/sek1/teil1