IT 4 KIDS

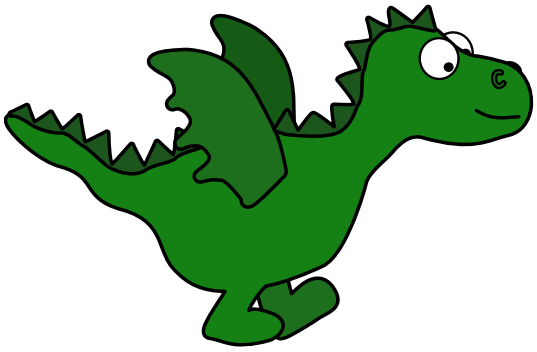
5. – 7. Klasse

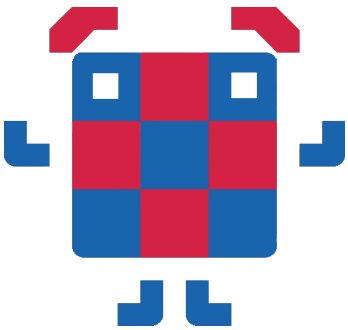
Alle Schulformen

Informatik erleben

– Teil 2 –

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** aufgearbeitet 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 Frusttrationstoleranz 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 gefordert und gefördert.

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 allgegenwertig.

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								
	Eingabe-Verarbeitung-Ausgabe	Standardalgorithmen	Variablen	Fehlersuche & Testen	Schleifen mit Bedingungen	Verschachtelte Verzweigungen	Struktogramm	Funktionen	Eigenes Spiel
Baden-Württemberg			X	X			X		X
Bayern	X		X	X	X	X	X	X	X
Berlin/Brandenburg			X					X	X
Hamburg	X		X		X	X		X	X
Hessen			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	
Rheinland-Pfalz			X	X	X	X			
Saarland			X	X	X	X	X		X
Sachsen					X	X	X		
Schleswig-Holstein	X	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. In Thüringen sind nur Themen des ersten Teils dieser Lernreihe im Bildungsplan verankert.

Inhaltsverzeichnis

EVA

45 Minuten
plugged

Seite 6

Standardalgorithmen

45 Minuten
(un)plugged

Seite 11

Variablen

90 Minuten
plugged

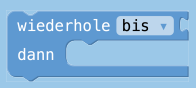
Seite 15

Fehlersuche & Testen

45 Minuten
plugged

Seite 24

Schleifen



45 Minuten
plugged

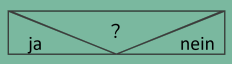
Seite 29

Verzweigungen

90 Minuten
plugged

Seite 33

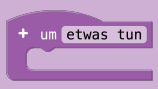
Struktogramm



45 Minuten
plugged

Seite 39

Funktionen



45 Minuten
plugged

Seite 43

Eigenes Spiel

90 Minuten
plugged

Seite 49

Material für 1 x 45 Minuten

Eingabe-Verarbeitung-Ausgabe

In dieser Stunde geht es um die Motivation, programmieren zu lernen. Dafür wird an bisherige Programmiererfahrungen angeknüpft. Die Schüler*innen aktivieren ihr Vorwissen und arbeiten einen Bezug zu ihrer Lebenswirklichkeit am Beispiel von Software im Alltag heraus.

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 6/7; Informationstechnologie, Anfangsunterricht), **Hamburg**[1] (Informatik, Jgs. 7; Naturwissenschaften – Gymnasium/Stadtteilschule, Jgs. 5/6), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... können Informationen sammeln, aufbereiten, bewerten und präsentieren.
- ... gehen beim Lernen strukturiert und systematisch vor, planen und organisiert eigene Arbeitsprozesse.
- ... setzen sich für Dinge ein, die ihnen wichtig sind, zeigen Einsatz und Initiative.
- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.
- ... verhalten sich in Konflikten angemessen, verstehen die Sichtweisen anderer und gehen darauf ein.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... beschreiben und analysieren digitale Anwendungen hinsichtlich der Wirkung von Algorithmen.
- ... festigen ihr Vorwissen der grafischen Programmierung durch erneutes Anwenden.

[1] In dieser Stunde kann die Erarbeitung mit Programmieraufgaben übersprungen und die frei gewordene Zeit zur Arbeitsphase hinzugefügt werden. Weitere Rechercheaufträge finden sich in der Stunde **Einführung in die Algorithmen** des ersten Teils der Lernreihe.

... leiten Definitionen von Fachbegriffen der Softwarentwicklung her, z.B. Software, Programmiersprache, Entwicklungsumgebung.

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Weitere verwendete Bausteine

Start:	**Wenn Start geklickt wurde**
Bewegung:	**Gehe ... Schritte**, **Drehe links/rechts um ... Grad**
Aussehen:	**Sage „ ... “ für ... Sekunden**
Schleifen:	**Wiederhole ... mal**, **Wiederhole fortlaufend**
Fühlen:	**berühre ich Farbe ...?**, **berühre ich ... ?**
Kontrolle:	**Wenn ... dann ...**, **Wenn ... dann ... sonst**

Vorbereitung

Bereite Dich auf die Stunde vor, indem Du Beispiele für Programmierung in Deinem Alltag sammelst. Du kannst Dich dabei von der Liste unter **Einstieg** inspirieren lassen. Falls Du im Unterricht mit einer digitalen Tafel arbeitest, kannst Du eine Collage mit den Ergebnissen Deines Brainstormings gestalten.
Wähle aus der Liste der Level unter **Erarbeitung** die Level aus, die Du mit Deiner Klasse in Teil 1 dieser Lernreihe gelöst hast. Vollziehe die Lösungen der Programmieraufgaben für Dich nach, damit Du auf die Fragen Deiner Schüler*innen gut eingehen kannst. Eine Erläuterung zu den Musterlösungen findest Du im Begleitmaterial unter **Lösungswege**.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Spaziergang durch den Alltag: Wo begegnen mir Programme und Software?	P	☐ (Digitaler) Ideenstern/Collage
10	Erarbeitung	Die Schüler*innen lösen bekannte Cubi-Level	EA/ PA	☐ Tablets/Laptops/PCs ☐ ggf. KV **QR-Codes EVA**
15	Arbeitsphase	Recherche zu Begriffen der Softwareentwicklung, Vorbereitung der Präsentation der Ergebnisse	GA	☐ Tablets/Laptops/PCs ☐ ggf. AB **Unsere Recherche**
10	Sicherung	Vortrag der Ergebnisse der Arbeitsphase, beantworten der Quizfragen	P	☐ ggf. Präsentationstechnik ☐ ggf. AB **Unsere Recherche**
5	Reflexion und Transfer	Motivation für das weitere Erlernen von Programmieren	P	

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

Einstieg

Steige in die Unterrichtsstunde mit einem Spaziergang durch den Alltag ein: Was haben die Schüler*innen heute schon genutzt, wo ein Programm dahinter steckte? Du kannst die Beispiele entweder während des Unterrichtsgesprächs an der (digitalen) Tafel mitschreiben oder schon im Vorfeld eine Collage mit Bildern aus Deinem Alltag vorbereiten und die Ideen der Schüler*innen ergänzen.
Markiert Programme und Software, die viele Schüler*innen nutzen. Beispiele für Programme und Software im Alltag sind:

- Apps: Wecker, Messengerdienste, Lernapps, Streaming von Musik/Podcasts, Nachrichten, Social Media, Spiele
- Geräte: Videospielkonsole, Elektrische Zahnbürste, Smartphone, Tablet, Laptop, Computer, digitale Uhr, Smartwatch, Bluetooth-Kopfhörer, Taschenrechner, digitale Tafel, Beamer, Auto (Navi, Musik, Fahrassistenten, automatisches Licht)
- Infrastruktur: Anzeigetafel von Straßenbahnen, Ampelschaltung, Parkhäuser, Straßenlaternen, Mobilfunkverbindungen, Energieversorgung, Überprüfung von Wasserqualität

- SmartHome: Sprachassistent, Licht über Bewegungsmelder, Rolladen, Lüftung, Heizung, Hausahltsgeräte wie Saugroboter und Kühlschränke, Unterhaltung

Betrachtet die Menge an Programmen und Software, die die Schüler*innen allein an diesem Tag genutzt haben. Verdeutliche die Relevanz, die dadurch das Handwerkszeug des Progammierens bei Erfindungen und der Weiterentwicklung von Technologien hat.

Erarbeitung

Gib den Schüler*innen die Gelegenheit, ihr Vorwissen zu blockbasierter Programmierung zu aktivieren. Teilt gemeinsam die Geräte aus, stellt sicher, dass sie mit dem Internet verbunden sind und öffnet den Cubi-Editor über editor.i4k.org. Gib den Arbeitsauftrag, eine Auswahl der folgenden, aus Teil 1 bekannten Cubi-Level erneut zu programmieren:

Cubi kennenlernen: Esuri und die Birne (Teil 1, Sequenzen)

Schleifen: Wellenreiten mit Natari, Nataris Kunststück (Teil 1, Schleifen)

Verzweigungen: Farben, Pilu trifft Entscheidungen (Teil 1, Verzweigungen)

Eine Lösungsanleitung findest Du im Lehrmaterial von Teil 1, bzw. im Begleitmaterial unter **Lösungswege**. Erinnere die Schüler*innen gegebenenfalls daran, dass sie die Level über **Menü** und **Öffnen** im Tab **Entwicklerreihe** finden.
Im Anschluss können sich die Schüler*innen kurz mit ihrem/ihrer Sitznachbar*in austauschen, was ihnen bei der Bearbeitung der Level leicht- und schwergefallen ist.

Arbeitsphase

In dieser Arbeitsphase geht es darum, dass die Schüler*innen ihr Vorwissen und ihren Wortschatz rund um das Thema **Programmieren** verschriftlichen und konkretisieren. Teile die Klasse in 4er- bis 6er-Gruppen für eine Gruppenrecherche ein.
Jede Gruppe bekommt einen Begriff zugeordnet, zu dem sie im Internet recherchieren soll. Hier findest Du die Begriffe sowie Empfehlungen, auf welchen Kindersuchmaschinen sich dazu Inhalte finden lassen:

- EVA-Prinzip (fragFINN, Serlo)
- Algorithmus und Programmieren (Helles Köpfchen, Klexikon, Serlo)
- Software (fragFINN, Klexikon)
- Programmiersprache (Klexikon, fragFINN, Serlo)
- Entwicklungsumgebung (fragFINN)
- Debugging (Helles Köpfchen)

Die Aufgabe der Schüler*innen ist es, eine Definition für ihren Begriff in eigenen Worten zu formulieren. Außerdem sollen sie ihnen bekannte Beispiele für ihren Begriff sammeln. Wo finden sie das Prinzip ihres Begriffs in Cubi wieder? Ihre Arbeitsergebnisse stellen die Schüler*innen auf einem Plakat, einer digitalen Präsentation oder Collage dar. Alternativ bereiten sie einen kurzen Vortrag ohne visuelle Unterstützung vor.
Als Vorbereitung für die Sicherungsphase bereiten die Schüler*innen bis zu drei Quizfragen für ihre Mitschüler*innen vor, die diese durch den Vortrag beantworten können.

Als Differenzierung nach unten kannst Du den Schüler*innen zu Beginn der Recherche das Arbeitsblatt **Unsere Recherche** mit an die Hand geben. Das Arbeitsblatt gibt ihnen Impulse, worauf sie bei ihrer Recherche achten können. Du findest es im Begleitmaterial.

Sicherung

Alle Gruppen stellen ihre Ergebnisse vor. Vor jedem Vortrag werden die Quizfragen vorgestellt. Du kannst die Antworten der Schüler*innen entweder einsammeln, selbst auswerten und in der nächsten Stunde austeilen oder die Quizfragen zur eigenständigen Kontrolle nutzen.

Reflexion und Transfer

Leite den Abschluss der Stunde mit der Frage ein, welche Motivation die Schüler*innen mitbringen, um Programmieren zu lernen. Beende die Stunde mit einem Ausblick auf die Themen dieser Unterrichtsreihe.

Standardalgorithmen

Unter dem Thema **Standardalgorithmen** ist eine Sammlung von häufig genutzten, erwiesenen Lösungsverfahren zusammengefasst. In dieser Unterrichtsstunde sind die Themen **Sortieren** und **Suchen** herausgegriffen, da diese Probleme den Schüler*innen aus ihrem Alltag bekannt sind.

Hinweis: *Die Übungsaufgaben in dieser Unterrichtsstunde können sowohl mit als auch ohne Tablet bearbeitet werden. Das hängt davon ab, ob die Kopiervorlage* ***Baumschule*** *ausgedruckt wird oder von den Schüler*innen als PDF am Tablet geöffnet wird.*

Anknüpfung an Bildungspläne

Schleswig-Holstein (Informatik, Jgs. 5 – 7)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... können Informationen sammeln, aufbereiten, bewerten und präsentieren.
- ... sind motiviert, Neues zu lernen und Dinge zu verstehen, strengen sich an, um sich zu verbessern.
- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.
- ... entwickeln eine eigene Meinung, treffen eigene Entscheidungen und vertreten diese gegenüber anderen.
- ... verhalten sich in Konflikten angemessen, verstehen die Sichtweisen anderer und gehen darauf ein.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... benennen einfache Standardalgorithmen.
- ... erläutern die Funktionsweise der Standardalgorithmen zum Sortieren und Suchen.
- ... leiten das Verfahren von verschiedenen Sortieralgorithmen her.

Vorbereitung

Die Kopiervorlagen **Baumschule 1 - 3** bestehen aus je 8 Seiten. Wir empfehlen daher, die Dateien nicht auszudrucken, sondern den Schüler*innen digital zur Verfügung zu stellen. Es müssen keine Anmerkungen in den Dokumenten gemacht werden. Alternativ können auch Vierergruppen gebildet werden, die sich eine Kopiervorlage teilen.
Schneide die Kopiervolage **Buchstaben** zurecht oder bitte die Schüler*innen eine Schere für die Stunde mitzubringen.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
10	Einstieg	Suchwettbewerb	P	□ KV **Baumschule 1-3** □ ggf. Tablets/Laptops/ PCs
10	Erarbeitung	Die Schüler*innen entwerfen ihre eigenen Sortier-Strategien	EA/ PA	□ KV **Buchstaben**
15	Arbeitsphase	Die Schüler*innen arbeiten das Prinzip von verschiedenen Sortieralgorithmen heraus	GA	□ AB **Sortieren durch Auswahl (Selection-sort)** □ AB **Sortieren durch Einfügen (Insertion-sort)** □ AB **Sortieren durch Aufsteigen (Bubble-sort)**
10	Präsentation und Reflexion	Diskussion über Vor- und Nachteile von verschiedenen Sortieralgorithmen, Übertrag auf den Alltag	P	

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

Einstieg

Eröffne die Stunde mit einem Wettbewerb. Trenne die Klasse in drei Gruppen. Teile den Gruppen je eine Version der Kopiervorlagen **Baumschule**[2] mit Datensätzen aus. Alle drei Versionen beinhalten die gleichen Informationen: Wo wurde im Herbst 2023 welcher Baum in Neuss gepflanzt? Der erste Datensatz ist nach dem **Straßennamen** sortiert, der zweite nach dem **Stadtteil** und der dritte nach der **Baumfamilie**. Kommentiere diesen

[2] Quelle: Stadt Neuss: *Baumpflanzung 2023*, gekürzt, Open NRW, Lizenz: CC-0 `https://open.nrw/dataset/stadt-neuss-baumpflanzungen-2023-ne` (zuletzt aufgerufen am 13.06.2024).

Unterschied gegenüber den Schüler*innen vorerst nicht. Sie sollen denken, dass alle die gleichen Chancen haben.
Die Gruppen spielen nun gegeneinander. Wer kann am schnellsten Deine Fragen mithilfe seines Datensatzes beantworten? Schreibe die Punkte an die Tafel. Welche Gruppe gewinnt?
Die Suchaufträge:

- Wie viele Bäume wurden in der Römerstraße gepflanzt? **10 Bäume**
- Wie viele verschiedene Baumarten wurden im Zedernweg gepflanzt? **4 Baumarten**
- Wie heißt der botanische Name für die Baumfamilie „Nuss“? **Juglans regia**
- Wie viele Bäume wurden im Stadtteil Weckhoven gepflanzt? **6 Bäume**
- In wie vielen Straßen wurden in Weckhoven Bäume gepflanzt? **In 4 Straßen**
- Was sind die auf ganze Zahlen gerundeten Koordinaten von dem Baum 23059? **51 und 7**
- Wie viele Eichen wurden gepflanzt? **19 Eichen**
- Wie heißt die Baumfamilie für „Carpinus betulus“? **Hainbuche**

Welche Strategien haben die Gruppen angewandt? Kommen die Schüler*innen darauf, dass sie unterschiedliche Sortierungen der Datensätzen hatten? Stelle eine Verbindung zum Programmieren her: Auch der Computer muss ständig in riesigen Datensätzen nach Informationen suchen. Da geht es am schnellsten, wenn er auf einen sortierten Datensatz zugreifen kann. Stelle damit das Thema der Stunde vor: **sortieren wie ein Computer**.

Erarbeitung

Formuliere den Arbeitsauftrag für die Schüler*innen. Sie schreiben in Einzelarbeit in eigenen Worten auf, wie sie einen Stapel von unsortierten Klassenarbeiten alphabetisch sortieren würden. Dabei sollten sie darauf achten, die Anweisungen genau zu formulieren. Als Hilfestellung erhalten sie die Kopiervorlage **Buchstaben**. Diese können sie sich selbstständig zurechtschneiden und ihr System testen, wenn sie den Stapel durchmischen. Im Anschluss tauschen sie die Anleitung mit ihren Sitznachbar*innen aus und testen deren Anleitung mit ihren Buchstaben-Kärtchen. Ist die Strategie verständlich? Führt sie zu dem richtigen Ergebnis?

Arbeitsphase

Nachdem die Schüler*innen ihre eigenen Strategien entwickelt haben, lernen sie bewährte Standardverfahren aus der Informatik kennen. Teile dazu Vierer- bis Fünfergruppen ein. Jede Gruppe bekommt einen Sortieralgorithmus zugeteilt und erhält von Dir das entsprechende Arbeitsblatt **Sortieren durch Auswählen (Selectionsort)**, **Sortieren durch Einfügen (Insertionsort)** oder **Sortieren durch Aufsteigen (Bubblesort)**. In ihrer Gruppe arbeiten die Schüler*innen das Prinzip des Algorithmus heraus.

Präsentation und Reflexion

Die einzelnen Gruppen stellen ihren Algorithmus der Klasse vor. Eine Gruppe von fünf Freiwilligen stellt sich zufällig nebeneinander. Die Aufgabe der vorstellenden Gruppe ist es nun, die Freiwilligen mit ihrem Sortieralgorithmus der Größe nach zu sortieren. Dabei kommentieren sie ihre Handlungen und erklären das Vorgehen des Algorithmus.
Schließe die Stunde mit einem Unterrichtsgespräch über Sortier- und Suchalgorithmen: Welcher Sortieralgorithmus war bei diesen Beispielen am schnellsten? In welchen Situationen muss ein Computer Daten sortieren? Wo müssen die Schüler*innen Dinge in ihrem Alltag sortieren? Welcher ist der favorisierte Sortieralgorithmus der Schüler*innen? Wie begründen sie ihre Auswahl? Wobei hilft das Wissen aus der heutigen Stunde im Alltag?
Beende den Unterricht mit einem kurzen Ausblick auf die nächste Stunde.

Material für 2 x 45 Minuten

Variablen

Der Einsatz von Variablen bringt neue Flexibilität in das Programmieren. Mithilfe von Variablen können Zahlen, Wörter oder Entscheidungen gespeichert und zu einem späteren Zeitpunkt erneut verwendet oder verändert werden. Variablen können auch Platzhalter sein, damit man mit ihnen rechnen kann, ohne, dass man ihren Wert schon kennt. Das braucht man, um z.B. den Punktestand von Spielenden zu vergleichen, der sich während des Spiels verändert.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 7; Informationstechnologie, Anfangsunterricht), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7), **Hessen** (Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Saarland** (IKT, S3), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **optional: Rheinland-Pfalz** (IPS, 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.
- ... 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.
- ... gehen beim Lernen strukturiert und systematisch vor, planen und organisieren eigene Arbeitsprozesse.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... identifizieren die Eigenschaften einer Variable.
- ... wenden das Variablenkonzept an: Initialisierung und Zuweisen und Auslesen von Werten.

... unterscheiden die Datentypen von Variablen nach: Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch).

Neue Bausteine

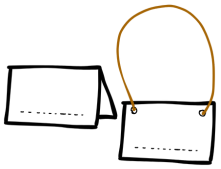
Variablen: **Setze ... auf ...**, **Verkleinere ... um**, **Erhöhe ... um**
Mathe: **Zuffallszahl von ... bis ...**

Weitere verwendete Bausteine

Start: **Wenn Taste ... gedrückt wurde**
Kontrolle: **Warte (bis)**
Schleifen: **Wiederhole bis**
Aussehen: **Wechsle zu Kostüm ...**, **Sage „... “**

Vorbereitung

Drucke die Kopiervorlage **Datentypen** aus und schneide sie entlang der vorgegebenen Linien zu. Sortiere sie für Dich als Übung zu den drei Datentypen Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch).
Mach Dich außerdem mit den Spielregeln des Würfelspiels vertraut. Entscheide Dich für die normale oder die Schnellspielvarainte. Bereite für die normale Variante die Kopiervorlage **Variablenreise** vor, indem Du die beiden Seiten mit den Zahlen- und Variablenschildern ausdruckst, zuschneidest und faltest. Dann können die Schüler*innen sie vor sich auf den Tisch stellen. Wenn Du eine Kordel zur Hand hast, können sie sich die Schüler*innen die Zettel auch wie ein Namensschild um den Hals hängen. Für die Schnellspielvariante brauchst Du nur die dritte Seite der Kopiervorlage **Variablenreise** ausdrucken und auf DIN-A5 zuschneiden.



Unterrichtsverlaufsplan – 1. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Comic zum Thema Variablen	P	☐ Präsentationstechnik ☐ Variablen-Comic
10	Erarbeitung	Zuordnungsspiel mit Datentypen	GA	☐ KV **Datentypen**
25	Arbeitsphase	Würfelspiel mit Variablen	GA	☐ Würfel ☐ Spielfiguren ☐ Spielplan ☐ KV **Variablenreise**
5	Sicherung	Wiederholung des Gelernten, Ausblick auf die nächste Unterrichtsstunde	P	

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

Einstieg

Eröffne die Unterrichtsstunde, indem Du das Thema der Stunde nennst: Variablen. Frage die Klasse, ob sie den Begriff schon einmal gehört hat und was er bedeutet. Lasse auch Vermutungen wie „Wenn etwas variabel ist, dann kann man es verändern." zu.
Lest gemeinsam den **Comic** zu Variablen. Zeige die Bilder nacheinander über eine digitale Tafel oder andere Präsentationstechnik. Lass die Schüler*innen die Texte vorlesen. Wie würden sie nach dem Lesen des Comics nun eine Variable beschreiben? Schreibe die wichtigsten Eigenschaften einer Variable gut sichtbar an die Tafel:

- Eine Variable ist ein Platzhalter/Speicher für einen bestimmten Wert.
- Der Wert einer Variable kann sich ändern.
- Eine Variable hat einen Namen.
- Eine Variable kann höchstens einen Wert gleichzeitig speichern, niemals zwei gleichzeitig.

Erarbeitung

Frage die Schüler*innen, was in dem Comic in einer Variable gespeichert wurde. Welche Art von Werten könnte man außer Zahlen noch in Variablen speichern und warum? Sammelt Beispiele wie die Folgenden an der Tafel:

- Benutzername und Passwort in einem Anmeldeformular.
- Ein ChatBot möchte sich die Antworten seines Gegenübers merken.

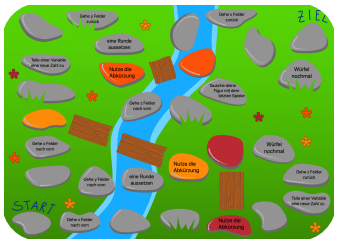
- Den Zustand einer Schranke im Parkhaus, z.B. ist sie geöffnet?

Sortiere die Beispiele grob nach Datentypen, ohne explizit auf sie einzugehen: Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch). Lasse in der Mitte Platz, um dort später das Thema der Stunde zu ergänzen.
Deute an, dass sich die Beispiele in verschiedene Kategorien oder, wie man in der Informatik sagt, **Datentypen** einteilen lassen, mit denen man unterschiedliche Sachen programmieren kann. Schreibe den Begriff **Datentyp** in die Mitte der Tafel.
Leite die folgende Klassenaktivität ein, indem Du sagst, dass die Klasse diese verschiedenen Datentypen herausfinden wird. Teile jedem Kind einen Teil der zugeschnittenen Kopiervorlage **Datentypen** aus. Fordere die Schüler*innen auf, sich in Gruppen zusammenzufinden, in denen alle den gleichen Datentyp, also die gleiche Art von Wert haben. Betone, dass die Aufgabe nur gelöst werden kann, wenn alle als ein Team zusammenarbeiten.
Die Klasse hat es geschafft, wenn sich jeweils alle Zahlen, alle Wörter und alle Wahrheitswerte (✓und ×) gefunden haben. Sprecht gemeinsam über die Werte, die schwer zuzuordnen waren. ***Hinweis:*** *Die Zettel* ***fünf*** *und* ***hundert*** *sind kniffelig. Auch wenn sie der Bedeutung nach eine Zahl sind, gehören sie trotzdem zu der Kategorie* ***Strings*** *(Wörter).* Wenn noch Zeit ist, sammel die Zettel wieder ein, mische sie und verteile sie neu an die Schüler*innen. Finden sie sich dieses Mal auch, ohne miteinander zu reden?
Bitte die Schüler*innen, wieder an ihren Platz zurückzukehren. Welche Datentypen haben die Schüler*innen in der Klassenaktivität kennengelernt? In welchen Gruppen haben sie sich zusammengefunden? Richte die Aufmerksamkeit der Schüler*innen wieder auf das Tafelbild. Wo würden sie ihre Zettel aus der Klassenaktivität einordnen? Lass die Schüler*innen nach vorne kommen und ihren Zettel aufhängen. Schreibe den Namen des Datentyps (Zahl, Wort, Wahrheitswert) zu den Beispielen dazu.
Ergänze ganz zum Schluss die Fachbegriffe aus der Informatik: Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch). Mache darauf aufmerksam, dass der Datentyp Boolean nur zwei Werte hat: wahr (✓) und falsch (×). Klärt Rückfragen zu den neuen Begriffen.

Datentypen

Integer: Englisch für „Ganzzahl“.
String: Englisch für „Schnur“, bezeichnet in der Informatik eine Zeichenkette, d.h. eine Aneinanderreihung von Buchstaben.
Boolean: Benannt nach dem englischen Mathematiker George Boole, dem Begründer der Aussagenlogik (s. **Verzweigungen** (Teil 1) und **Kopfgesteuerte Schleifen**).

Arbeitsphase



In dem folgenden Würfelspiel sollen die Schüler*innen das Konzept der Variable als Wertespeicher verinnerlichen. Stelle die Spielregeln vor und geht einzelne Aktionen beispielhaft durch.

- Zu Beginn: Verteilt die Sielfiguren und setzt sie auf **Start**. Jede Person erhält ein Schild mit einer Zahl und hängt es sich um oder stellt es vor sich auf. Die jüngste Person ist zuerst der Computer und stellt sich in die Mitte. Alle anderen Personen

stellen sich in einem Kreis außen herum. Der Computer schließt die Augen und dreht sich. Nach einer gewissen Zeit bleibt sie stehen und öffnet die Augen. Die Person, die sie anschaut, ist der Wert für die Variable **x** und erhält zusätzlich das Variablenschild für **x**. Der Vorgang wird für die anderen beiden Variablen wiederholt. Alle Personen setzen sich um das Spielfeld herum.

ODER

Schnellspielvariante ohne Variablenschilder: Die Werte der Variablen werden gewürfelt und auf der Kopiervorlage **Variablenreise** notiert. Bei dieser Variante kann auch in kleineren Gruppen gespielt werden.

- Spielzug: Es wird reihum gespielt. Jeder Zug besteht aus: würfeln, die Spielfigur um die gewürfelte Augenzahl bewegen und, wenn die Spielfigur auf einem Spielfeld mit einer Aktion gelandet ist, diese ausführen. ***Hinweis:*** *Nach Ausführen der Aktion ist der Zug beendet, auch wenn die Person auf ein neues Aktionsfeld kommt. Dieses wird nicht ausgeführt.*

- Aktionsfeld: **Gehe x/y/z Felder nach vorne**. Die Person, die am Zug ist, sucht die Person, die das Schild mit dem **x** umhängen hat bzw. schaut in der Schnellspielvariante auf der Kopiervorlage **Variablenreise** nach. Dessen Zahl gibt an, wie viele Schritte die Person, die am Zug ist, nach vorne ziehen darf.

- Aktionsfeld: **Teile einer Variable eine neue Zahl zu**. Teilt der Variable wie am Anfang des Spiels eine neue Zahl zu. Die Rolle des Computers darf nun die Person übernehmen, die auf das Aktionsfeld gekommen ist.

Teile die Klasse in 5er- bis 6er-Gruppen ein. Jede Gruppe bekommt einen Würfel, einen Spielplan, die Zahlenschilder, die Variablenschilder für x, y und z und für jede Person eine Spielfigur. Die Gruppen führen das Spiel selbstständig durch. Das Spiel endet, wenn die erste Person das Ziel erreicht hat. Ist noch etwas Zeit, kann um den zweiten und dritten Platz gespielt werden.

Sicherung

Fasst am Ende der Stunde alle neuen Informationen zusammen. Was ist eine Variable? Was ist der Unterschied zwischen dem Setzen und dem Auslesen einer Variable? Welche Datentypen gibt es? Frage die Schüler*innen, was ihnen bei dem Zuordnungsspiel zu den Datentypen und dem Würfelspiel zu Variablen allgemein leicht- und was ihnen schwergefallen ist.
Beende die Stunde, indem Du einen Ausblick auf die nächste Stunde gibst, in der die Schüler*innen Cubi-Level mit Variablen programmieren werden.

Vorbereitung

Vollziehe die Lösung der Cubi-Level nach, damit Du auf die Fragen Deiner Schüler*innen gut eingehen kannst. 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.
Präpariere die Deckblätter für die Stationen und die zugehörigen Tippkarten. Baue vor der Stunde die Stationen im Klassenraum auf.

Unterrichtsverlaufsplan – 2. Stunde

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Wiederholung der Definition von **Variablen**	P	
5	Erarbeitung	Einführung in Variablen in Cubi	P	☐ Präsentationstechnik ☐ Tablets/Laptops/PCs
25	Arbeitsphase	Stationenarbeit zu Cubi-Leveln	EA	☐ KV **Deckblätter** ☐ KV **Tippkarten** ☐ Präsentationstechnik ☐ Tablets/Laptops/PCs
10	Sicherung und Reflexion	Vorstellen der Programmierergebnisse, Beispiele für Programme mit Variablen, Nutzen und Limitationen von Datentypen	P	☐ Präsentationstechnik

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

Einstieg

Wiederhole zu Beginn der Stunde, was die Schüler*innen in der letzten Stunde gelernt haben. Was ist eine Variable und wofür wird sie verwendet?
Hinweis: *Eine Variable ist ein Speicher für einen Wert und gleichzeitig ein Platzhalter. Sie wird in der Programmierung verwendet, wenn ein bestimmter Wert mehrfach gebraucht wird oder er zum Zeitpunkt des Programmschreibens noch nicht bekannt ist.*
Was kann man mit einer Variable manchen?
Hinweis: *Man kann ihr einen beliebigen Wert zuweisen, ihren Wert basierend auf dem aktuellen Wert verändern, z.B.* ***verkleinere x um 2*** *oder ihren aktuellen Wert auslesen.*

Erarbeitung

Öffne das Level **Tiefsee** in Cubi über eine digitale Tafel oder ähnliche Präsentationstechnik. Gib den Schüler*innen kurz Zeit, um die vorhandenen Bausteine zu lesen. Lass Vermutungen anstellen, was passiert, wenn Du gleich das Level startest. Prüft die Vermutungen.
Lösung: *Es passiert zunächst nichts. Die Qualle kann jedoch mit den Pfeiltasten gesteuert werden.*

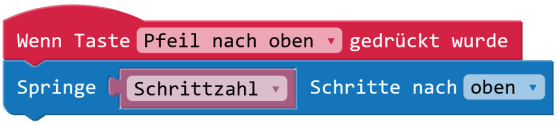
Levelvorlage:
`level.i4k.org/tiefsee_sek`

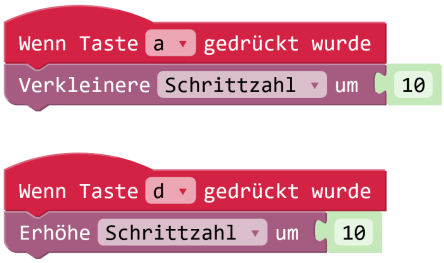
Levellösung:
`level.i4k.org/tiefsee_sek_lsg`

Erkläre, dass Du an diesem Level das **Erstellen** und **Verwenden** von Variablen zeigen wirst und stelle die Idee des Levels vor. Anstatt die Qualle bei jedem Tastendruck **100** Schritte springen zu lassen, sollte man die Anzahl der Schritte während des Levels ändern können. Dafür braucht man Variablenbausteine. Welche Farbe hat die Kategorie? Lass die Schüler*innen erneut Vermutungen anstellen, was die einzelnen Bausteine machen.

- Zeige Variable ... und Verstecke Variable ... bestimmen, ob die Variable auf der Bühne angezeigt wird oder nicht.
- Der Baustein Setze ... auf weist der Variable einen neuen Wert zu. Dabei spielt der bisherige Wert der Variable keine Rolle. Dies ist vergleichbar mit dem Würfelspiel aus der vergangenen Stunde.
- Die Bausteine Verkleinere ... um und Erhöhe ... um ändern den Wert der Variable. Das bedeutet, dass die Zahl, die an den Baustein angepuzzelt wird, zu dem aktuellen Wert addiert oder von ihm subtrahiert wird. Das Ergebnis ist der neue Wert der Variable.
- Der Variablenbaustein, der aussieht wie ein Zahlen-Baustein, kann zum Auslesen des Werts der Variable genutzt werden oder als Platzhalter zum Rechnen.

Zeige, wie man eine neue Variable erstellt, indem Du die Kategorie Variablen öffnest und oben auf **Variable erstellen** klickst. Es öffnet sich ein Dialogfenster. Gib hier den Namen der Variable ein: **Schrittzahl**.
Ziehe den Variablenbaustein **Schrittzahl** viermal in die Programmierfläche und setze ihn als Anzahl der Schritte in die Springe-Bausteine ein. Füge den **Wenn Start geklickt wurde**-Baustein hinzu und setze nach ihm die Variable **Schrittzahl** auf **100**. Trage verschiedene Werte ein und teste das Level jedes Mal.





Füge **Wenn Taste ... gedrückt wurde** als zwei neue **Startbausteine** zu dem Programm hinzu. Wähle bei dem einen die Taste **a** aus und bei dem anderen die Taste **d**. Hänge an den Baustein zur Taste **a** den Baustein **Verkleinere Schrittzahl um** an und wähle als Zahl den Wert **10**. Starte das Level erneut, steuere die Qualle über die Bühne und drücke zwischendurch immer mal wieder die Taste **a**. Fällt den Schüler*innen auf, dass die Schritte immer kleiner werden? Sehen sie, dass sich die Zahl hinter **Schrittzahl** auf der Bühne immer weiter verkleinert?
Frage die Schüler*innen, wie Du das Programm erweitern musst, damit man durch Drücken der Taste **d** die Schrittzahl wieder vergrößern kann.

Arbeitsphase

Leite in die Arbeitsphase über, in der die Schüler*innen in ihrem eigenen Tempo in Einzel- oder Partnerarbeit verschiedene Level in Cubi zu **Variablen** lösen können. Stelle die einzelnen Level und die zugehörigen Aufgabenstellungen kurz vor. Wiederhole mit den Schüler*innen, wie sie über das **Menü** ☰ bzw. mit dem QR-Code die einzelnen Level öffnen können. Verteile mit den Schüler*innen die Geräte und stelle sicher, dass alle mit dem Internet verbunden sind.

- **Tiefsee**: Wiederhole das Level, das ihr gemeinsam als Klasse gelöst habt.
- **Würfelspaß**: Programmiere mithilfe von Variablen einen Würfel. Du würfelst, indem Du Cubi anklickst. Er sagt Dir, welche Zahl du gewürfelt hast.
  🧩 **Knobelaufgabe:** Wenn Du eine 6 gewürfelt hast, sagt Cubi, dass Du gewonnen hast.
- **Feuer speien**: Programmiere mithilfe von Variablen einen Countdown. Bei einem Countdown zählt man rückwärts bis zum Start. Nach Ablauf des Countdowns speit der Drache Feuer.
  🧩 **Knobelaufgabe:** Auch der Hintergrund wechselt sein Kostüm, wenn der Countdown bei 0 angekommen ist.
- **Freizeitplanung**: Hier muss nicht programmiert werden, sondern es müssen nur Werte in bestehenden Bausteinen geändert werden. Dies ist eine kreative Aufgabe. Der Ablauf des Levels ist auf der Kopiervorlage **Aufgabenstellungen** erklärt.

Levelvorlage:
`level.i4k.org/wuerfelspass_sek`

Levellösung:
`level.i4k.org/wuerfelspass_sek_lsg`

Levelvorlage:
`level.i4k.org/feuer_speien_sek`

Levellösung:
`level.i4k.org/feuer_speien_sek_lsg`

Levelvorlage:
`level.i4k.org/freizeitplanung_sek`

Sicherung und Reflexion

Gib am Ende der Stunde einzelnen Schüler*innen die Möglichkeit, ihr Programm der Klasse vorzustellen. Frage sie und ihre Mitschüler*innen, was ihnen leicht- und was ihnen schwergefallen ist. Überlegt im Plenum, wann Variablen beim Programmieren verwendet werden:

- Man möchte den gleichen Wert mehrfach in einem Programm verwenden.
- Man möchte mit einem Wert arbeiten, der sich während des Programms ändert, z.B. die verbleibende Zeit bis zum Ende des Spiels.
- Man möchte einen Wert mit einem anderen Wert vergleichen, den man beim Schreiben des Programms noch nicht kennt, z.B. den Punktestand.

Wenn noch etwas Zeit ist, könnt ihr Beispiele für Programme mit Variablen sammeln wie Schrittzähler, Countdown, Geschwindigkeit bei einem Rennauto...
Wiederholt die drei verschiedenen Datentypen, die ihr für Variablen kennen gelernt habt: Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch). Warum kann es sinnvoll sein, Variablen je nach ihrem Datentyp unterschiedlich zu behandeln?

- Es gibt Aktionen, wie z.B. Division, die nur bei bestimmten Datentypen Sinn ergeben.
- Variablen von unterschiedlichen Datentypen lassen sich schwer vergleichen. Der Ausdruck **4 > 7** lässt sich leicht zu **falsch** auswerten. Auch der Ausdruck **Apfel < Zahnbürste** ist leicht zu **richtig** auswertbar, da man Wörter alphabetisch sortieren und damit auch vergleichen kann. Jedoch lässt sich der Ausdruck **bunt > 5** nicht sinnvoll auswerten und führt zu einer Fehlermeldung. In Cubi lässt sich daher diese Kombination aus Bausteinen auch nicht bilden.

Beende die Stunde mit einem Ausblick auf das nächste Thema.

IT 4 KIDS

Material für 1 x 45 Minuten

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 sie 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. 7; Informationstechnologie, Anfangsunterricht), **Hessen** (Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Saarland** (IKT, S3), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **optional: Rheinland-Pfalz** (IPS, Jgs. 5/6)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... gehen beim Lernen strukturiert und systematisch vor, planen und organisieren eigene Arbeitsprozesse.
- ... haben Vertrauen in die eigenen Fähigkeiten und glauben an die Wirksamkeit des eigenen Handelns.
- ... arbeiten ausdauernd und konzentriert, geben auch bei Schwierigkeiten nicht auf.
- ... können Informationen sammeln, aufbereiten, bewerten und präsentieren.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... testen Algorithmen unter wechselnden Ausgangsbedingungen.

... testen die Korrektheit und die Wirkungsweise eines Algorithmus systematisch.

... beurteilen Programme im Hinblick auf Korrektheit.

... erkunden die Ursachen von Programmfehlern systematisch.

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Vorwissen

Algorithmisches Verständnis, Variablen

Vorbereitung

Mach Dich mit den Beispielprogrammen aus der Arbeitsphase vertraut und löse die Quizfragen. Du findest das 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.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Unterrichtsgespräch zu bisherigem Umgang mit Programmierfehlern und Strategien zur Fehlersuche in Cubi-Leveln	P	
15	Erarbeitung	Funktionsweise eines Algorithmus in Pseudocode-Form analysieren durch das Bestimmen von Variablenwerten	EA	□ AB **Variablen raten**
15	Arbeitsphase	Spielregeln des Cubi-Levels **Wüstenjagd** herausfinden	EA/ PA	□ AB **Wüstenjagd** □ Tablets/Laptops/PCs □ ggf. KV **QR-Codes Fehlersuche und Testen 2**
10	Sicherung und Reflexion	Quiztime, Fehler als Teil des Programmierprozesses anerkennen	P	

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

Einstieg

Nenne zu Beginn der Stunde das Thema: **Fehlersuche**. Leite ein Unterrichtsgespräch an. Mögliche Gesprächseinstiege sind:

- Wie seid ihr bisher mit Fehlern in euren Programmen umgegangen?
- An welche Fehlersuche-Level erinnert ihr euch?
- Was passiert, wenn ein Programm einen Fehler hat?

Erarbeitung

Wiederhole zunächst mit den Schüler*innen die Besonderheiten eines Pseudocodes:

- Algorithmus/Programm auf dem Papier
- Unabhängig von der Programmiersprache
- Für Menschen verständlich, für die Maschine/den Computer jedoch nicht
- Pseudocode ist wie eine „Stichwortliste“ für ein Programm

Teile dann das Arbeitsblatt **Variablen raten** aus. Alternativ kannst Du es auch über eine digitale Tafel oder ähnliche Präsentationstechnik zeigen und ihr löst die folgende Aufgabe im Plenum.
Kläre Verständnisfragen zu der Notation des Pseudocodes auf dem Arbeitsblatt **Variablen raten**. Formuliere im Anschluss den Arbeitsauftrag: Die Schüler*innen sollen herausfinden, welche Werte die Variablen bei den gegebenen Anfangswerten am Ende des Programms haben.

Besonders schnelle Schüler*innen können durch systematisches Ausprobieren oder scharfes Hinschauen herausfinden, welche Aufgabe dieser Algorithmus löst.

Vergleicht die Lösungen entweder in Kleingruppen oder im Plenum.

Arbeitsphase

Formuliere den Arbeitsauftrag für die kommende Arbeitsphase: Die Schüler*innen sollen durch systematisches Ausprobieren und eigene Analysemethoden herausfinden, welche Aufgabe das Cubi-Level **Wüstenjagd** erfüllt. Sie sollen auf dem Arbeitsblatt **Wüstenjagd** die Spielregeln und erkannte Abhängigkeiten eintragen. Haben sie das Grundprinzip erkannt, ermutige sie, Ausnahmen oder Grenzfälle zu identifizieren.
In dem Level hat sich außerdem ein Fehler eingeschlichen. Die Figur Cubi verhält sich nicht wie erwartet und läuft nach rechts, obwohl die Pfeiltaste nach links gedrückt wurde und andersherum. Was müssen die Schüler*innen in dem Programm ändern?
Teile das Arbeitsblatt **Wüstenjagd** aus. Baue außerdem mit den Schüler*innen gemeinsam die Tablets, Laptops oder Computer auf und stell sicher, dass jedes Gerät mit dem Internet verbunden ist. Lass sie den Cubi-Editor über `editor.i4k.org` öffnen und stell sicher, dass alle zu dem Level **Wüstenjagd** finden.

Levelvorlage:
`level.i4k.org/wuestenjagd_sek`

Levellösung:
`level.i4k.org/wuestenjagd_sek_lsg`

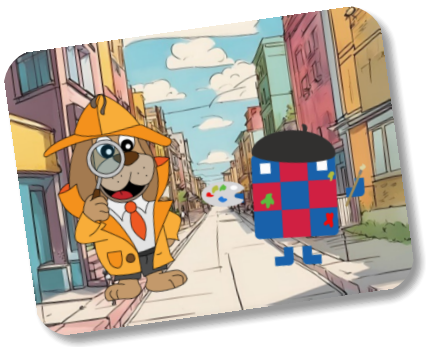
Sicherung und Reflexion

Öffne das Level **Wüstenjagd** über eine digitale Tafel oder ähnliche Präsentationsmöglichkeit. Löst zunächst den Programmierfehler von der Figur **Cubi** auf. Die Steuerung nach links und rechts ist vertauscht. Bitte ein Kind, den Fehler zu erklären. Woran hat es ihn erkannt? Wie kann der Fehler behoben werden?
Danach ist es Zeit für ein Quiz! Wer hat das Cubi-Level **Wüstenjagd** ganz genau unter die Lupe genommen und alle Regeln herausgefunden? Ziehe die Bühne so groß, damit die Programmierfläche verdeckt wird und die Schüler*innen nicht lünkern können. Bei Unsicherheiten könnt ihr die Situation der Frage in Cubi nachstellen.

- Lässt sich Cubi in alle Richtung steuern? **Ja**

- Prallt Cubi immer vom Rand ab? **Nein, wenn man ganz häufig in eine Richtung klickt, verschwindet die Figur hinter dem Rand.**
- In welche Richtung dreht sich Cubi, wenn er den Rand berührt hat? **Zu der Wasserflasche**
- Wann hast du das Level gewonnen? **Wenn die Variable Gefunden größer als fünf ist, bzw. Cubi die Wasserflasche mindestens sechs Mal berührt hat.**
- Was passiert mit der Wasserflasche, wenn sie berührt wird? **Sie verschwindet und taucht fünf Sekunden später wieder auf.**
- Was passiert sonst noch im Level, wenn Cubi die Wasserflasche berührt? **Die Variable Gefunden wird um eins hochgezählt, die Variable Zeit um zehn und Cubi sagt: „Juhu!“**
- Wann ändert Cubi sein Kostüm? **Wenn die Wasserflasche angeklickt wurde oder, wenn er angeklickt wurde und eine zufällig generierte Zahl kleiner als drei ist. Die Wahrscheinlichkeit beträgt hierbei $\frac{1}{5} = 20\%$.**
- In welchem Muster bewegt sich die Wasserflasche? **Sie dreht sich in eine zufällige Richtung und geht dann bis zum Rand.**
- Wann erscheint ein Meteorit? **Wenn das Level verloren wurde.**
- Wie viele Sekunden hast du am Anfang Zeit, um die Wasserflasche zu fangen? **10 Sekunden**

Betone am Ende der Stunde noch einmal, dass selbst die erfahrensten Programmierer*innen Fehler beim Schreiben ihres Codes machen und dass es wichtiger ist, einen Fehler identifizieren zu können, als sie gar nicht erst zu machen. Beende die Stunde mit einem Ausblick auf das Thema der nächsten Stunde.



Material für 1 x 45 Minuten

Schleifen mit Bedingungen

Bisher haben die Schüler*innen Anweisungen entweder unendlich oft wiederholt (**Wiederhole fortlaufend**) oder eine bestimmte Anzahl an Wiederholungen im Vorfeld bestimmt (**Wiederhole ... mal**). Diese Art von Programmen reagieren nicht auf ihre Umwelt und Benutzer-Interaktionen haben keinen Einfluss auf den Ablauf oder das Ergebnis des Programms. Dies wird durch Schleifen mit Bedingungen möglich. Die Schüler*innen implementieren Programme, bei denen die Figuren aufeinander oder auf den Benutzer reagieren können.
Hinweis: *Alle Schleifen, die in Cubi verwendet werden, sind kopfgesteuert. Das bedeutet, dass erst die Bedingung der Schleife geprüft wird und dann die Bausteine ausgeführt werden. So kann es vorkommen, dass die Bausteine innerhalb der Schleife bei einem Programmlauf gar nicht ausgeführt werden, wenn die Bedingung schon beim ersten Durchlauf nicht erfüllt ist.*

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 6; Informationstechnologie, Anfangsunterricht; Natur und Technik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Saarland** (IKT, S3), **Sachsen** (Informatik, Jgs. 8), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **optional: Rheinland-Pfalz** (IPS, 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.
- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... werten verknüpfte Aussagen in Bezug auf ihren Wahrheitsgehalt aus.
- ... interpretieren und verwenden kopfgesteuerte Schleifen.
- ... negieren verknüpfte Aussagen.

Neue Bausteine

Schleifen: **Wiederhole bis**, **Wiederhole solange**
Fühlen: **berühre ich ... ?**

Weitere verwendete Bausteine

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

Vorbereitung

Löse das Cubi-Level **Verbrecherjagd**. Du findest es, indem Du oben links im **Menü** ☰ auf **Öffnen** 🗀 gehst und sie anschließend unter dem Tab **Entdeckerreihe** auswählst oder die QR-Codes weiter unten scannst.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
10	Einstieg	Spiel: **Wahrheitssuche**	P	
10	Erarbeitung	Aussagen verknüpfen und negieren	EA/ PA	☐ AB **Detektivsuche**
20	Arbeitsphase	Cubi-Level **Verbrecher-jagd**	EA	☐ Tablets/Laptops/PCs ☐ ggf. KV **QR-Codes Kopfgesteuerte Schleifen**
5	Reflexion	Vorstellen der Program-mierergebnisse	P	

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

Einstieg

Starte direkt in das Thema, indem Du die folgenden Sätze einzeln vorliest. Die Schüler*innen drücken aus, ob die Sätze wahr oder falsch sind. Dies kann durch Meldung sein, Farbkärtchen, aufstehen/hinsetzen oder etwas mobiler, sofern der Platz vorhanden ist, indem man zur Türseite oder zur Fensterseite läuft.

1. Es regnet.
2. $14 \div 7 = 2$
3. Wir haben gerade Kunstunterricht.

4. Gelb ist meine Lieblingsfarbe.
5. Heute ist ...tag.
6. Ich bin mit dem Fahrrad zur Schule gekommen.
7. Ich bin heute zu Fuß oder mit einem Verkehrsmittel zur Schule gekommen.
8. Unser/e Lehrer*in heißt ...
9. 9 + 2 ist nicht 11.
10. Walnüsse sind lecker.

Bei welchen Sätzen war sich die Klasse einig? Bei welchen nicht? Bei welchen Sätzen hätte auch jemand von außen die Schüler*innen zuordnen können? Was haben die Sätze jeweils gemein? Arbeite mit der Klasse Unterschiede und Gemeinsamkeiten der einzelnen Sätze heraus, z.B.:

- Allgemeingültige Aussagen werden von allen Menschen immer gleich eingeordnet: 2, 9.
- Aussagen, die ihren Wahrheitswert abhängig von der Situation ändern können, aber eindeutig beobachtbar sind: 1, 3, 5, 6, 7, 8.
- Meinungen können nur die Personen selbst äußern: 4, 10.

Greife die mittlere Kategorie heraus und stelle sie als Thema der Stunde vor: Aussagen, die abhängig von der Situation mal wahr und mal falsch, aber eindeutig beobachtbar sind. Diese Art von Aussagen werden die Schüler*innen in Cubi selber umsetzen.

Erarbeitung

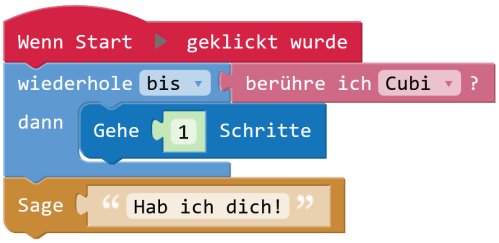
Gib den Schüler*innen Zeit zu üben und Sicherheit zu gewinnen, bevor es an die Arbeit mit Cubi geht. Teile dazu das Arbeitsblatt **Detektivsuche** aus. Gehe mit den Schüler*innen die einzelnen Aufgaben durch und kläre Verständnisfragen.
Die Schüler*innen bearbeiten das Arbeitsblatt **Detektivsuche** in Einzelarbeit. Sobald sie fertig sind, können sie es mit ihren Sitznachbar*innen vergleichen.

Arbeitsphase 

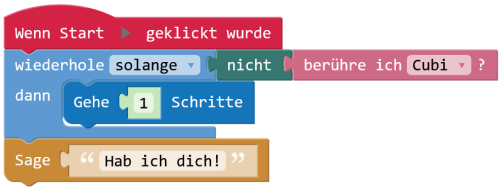
Leite in die nächste Unterrichtsphase über, öffne den Cubi-Editor und zeige ihn über die digitale Tafel oder ähnliche Präsentationstechnik. Rufe das Level **Verbrecherjagd** auf.
Welche Kategorien sind neu für die Schüler*innen oder welche haben sie dieses Schuljahr noch nicht verwendet? Lenke die Aufmerksamkeit auf die **Fühlen**-Kategorie und schaut euch den Baustein in dieser Kategorie an. Erschließt euch aufgrund des Namens, was der Baustein macht. Konstruiere auf der Bühne nacheinander die beiden Situationen von dem Arbeitsblatt **Detektivsuche** und frage die Schüler*innen, welche der **Fühlenbausteine** nun wahr und welche falsch sind.
Zeige nach den **Fühlenbausteinen** einen weiteren neuen Baustein: die **Wiederhole bis**-Schleife. ***Hinweis:*** *Du findest sie als **Wiederhole solange**-Schleife in der Werkzeugpalette. Ziehe sie auf die Programmierfläche und klicke auf das helle Feld. Es öffnet sich eine Auswahl und Du kannst zwischen **bis** und **solange** wechseln.*

Levelvorlage:
`level.i4k.org/verbrecherjagd_sek`

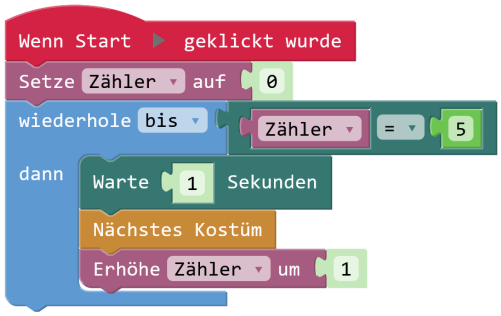
Levellösung:
`level.i4k.org/verbrecherjagd_sek_lsg`



Wie könnte die Wiederhole bis-Schleife mit den Fühlenbausteinen zusammenpassen und was ist der Unterschied zwischen der Wiederhole bis- und der Wiederhole solange-Schleife? Programmiert die erste Aufgabe im Plenum: **Wenn Start geklickt wurde**, **geht** der Detektivhund **einen Schritt**, **bis** er **Cubi berührt**. Danach **sagt** er, dass er Cubi gefangen hat.
Bau dann mit den Schüler*innen gemeinsam die Tablets, Laptops oder Computer auf und stell sicher, dass jedes Gerät mit dem Internet verbunden ist. Lass die Schüler*innen den Cubi-Editor über `editor.i4k.org` öffnen und stell sicher, dass alle zu dem Level **Verbrecherjagd** finden.



Formuliere den Arbeitsauftrag für diese Arbeitsphase: Zunächst programmieren die Schüler*innen die erste Aufgabe für sich nach. Dann ersetzen sie die Wiederhole bis-Schleife durch eine Wiederhole solange-Schleife. Was müssen sie an ihrem Programm ändern, um das gleiche Ergebnis zu erzielen wie in der Aufgabe davor? ***Lösung:*** *Sie müssen einen **nicht**-Baustein zwischen die Schleife und den Fühlenbaustein einfügen.*



Als letztes programmieren sie Cubi. Er will nicht gefangen werden und verkleidet sich deshalb fünf Mal, um nicht aufzufallen. Dafür wird eine neue Variable erstellt, welche in der Wiederhole bis-Schleife nach einer Sekunde **warten** und einem **Kostümwechsel** um eins hochgezählt wird. Was passiert, wenn das Erhöhen der Variable vergessen wird? ***Lösung:*** *Das Programm läuft für immer weiter.*

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.
Beende die Stunde mit einem Ausblick auf das nächste Thema: **Verzweigungen**. Auch hier werden die Bausteine aus der Kategorie Fühlen eine große Rolle spielen.

Material für 90 Minuten

Verzweigungen und Entscheidungsbäume

Bisher haben die Schüler*innen nur Anweisungen und keine Kontrollstrukturen wie **Wenn dann**-Bausteine oder **Schleifen** in andere Kontrollstrukturen verschachtelt. Mit der tieferen Verschachtelung von **Wenn dann**-Bausteinen werden die Programme schwieriger nachzuvollziehen, aber man kann mit ihnen komplexere Sachverhalte implementieren und Abhängigkeiten elegant darstellen.
Hinweis: *Diese Doppelstunde ist zweigeteilt. Für die erste Stunde werden keine Tablets, Laptops oder PCs für die Schüler*innen benötigt. Erst in der zweiten Stunde wird programmiert.*

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 6; Informationstechnologie, Anfangsunterricht), **Hamburg** (Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Saarland** (IKT, S3), **Sachsen** (Informatik, Jgs. 8), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7), **optional: Rheinland-Pfalz** (IPS, 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...

- ... stellen den algorithmischen Grundbaustein **Verzweigung** grafisch dar und entwerfen einen eigenen Entscheidungsbaum.
- ... entwerfen Algorithmen mithilfe des Grundbausteins **Verzweigung**.
- ... implementieren Algorithmen unter der Nutzung von Rechen- und Vergleichsoperatoren.

... implementieren Programme mit einer Verschachtelungstiefe von mindestens drei Ebenen.

Neue Bausteine

Start: **Wenn ich angegklickt wurde**
Kontrolle: Verschachtelungen von **Wenn ... dann ... sonst**-Bausteinen

Weitere verwendete Bausteine

Kontrolle: **Wenn ... dann**
Variablen: **Variablen** erstellen und nutzen
Aussehen: **Wechsle zu Kostüm ...**

Vorbereitung

Bereite das Tafelbild für die Erarbeitungsphase vor.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Wiederholung zu Verzweigungen	P	
30	Erarbeitung	Entscheidungsbaum: kennenlernen, Pfad finden und einzeichnen, eigenen Entscheidungsbaum entwerfen	EA/PA	☐ Tafelbild verschachtelten **Wenn dann**-Bausteinen ☐ AB **Kleiderschrank** ☐ AB **Hund Cali**
45	Arbeitsphase	Entscheidungsbaum für Cubi-Level **Wettervorhersage** nachvollziehen, Cubi-Level **Wettervorhersage** lösen	P/EA	☐ AB **Wettervorhersage** ☐ Tablets/Laptops/PCs ☐ Levelcodes ☐ ggf. KV **QR-Codes Verzeigungen und Entscheidungsbäume**
10	Reflexion	Vorstellen der Programmierergebnisse	P	

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

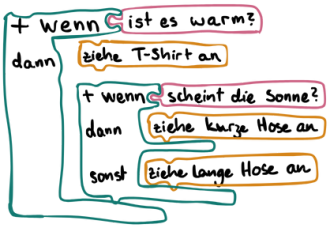
Einstieg

Eröffne die Stunde, indem Du das Thema nennst: **verschachtelte Verzweigungen**. Lass die Schüler*innen ihr Vorwissen zu Verzweigungen aktivieren, indem Du sie fragst, was sie zu Verzweigungen noch wissen. Vielleicht kann jemand den **Wenn dann**-Baustein an die Tafel malen? Weise darauf hin, dass in den **dann**-Teil später noch etwas reingemalt werden wird und dementsprechend viel Platz eingeplant werden sollte.

Zeige das Ziel der Stunde, indem Du in den **Wenn dann**-Baustein einen weiteren **Wenn dann**-Baustein einzeichnest und andeutest, dass man diesen Schritt sehr häufig wiederholen kann. Nenne den Fachbegriff für dieses Konzept: **Verschachtelung**. Die Schüler*innen können sich das wie eine Matrjoschka vorstellen oder ein Scherzpaket, wo immer eine kleinere Version drin ist, wenn man die äußere Verpackung aufmacht.

Erarbeitung

Damit man bei den ganzen **Wenn dann**-Bausteinen nicht den Überblick verliert, gibt es eine bildliche Darstellungsform von verschachtelten **Wenn dann**-Bausteinen: **Entscheidungsbäume**.



Teile das Arbeitsblatt **Kleiderschrank** aus. Schaut euch den Entscheidungsbaum **Kleiderschrank** an und arbeitet gemeinsam die Verbindung zu verschachtelten **Wenn dann**-Bausteinen heraus. Hier kann es helfen, immer die gleiche Formulierung zu benutzten: „**Wenn** es warm ist, **dann** ziehe ich ein T-Shirt an. **Wenn** es warm ist und die Sonne scheint, **dann** ziehe ich eine kurze Hose an. **Wenn** es warm ist und die Sonne **nicht** scheint, **dann** ziehe ich eine lange Hose an.“ Zeichne dieses Beispiel in Pseudocode und angedeuteten Bausteinen an die Tafel.

Entlasse die Klasse in Einzel- bzw. Partnerarbeit. Die Schüler*innen durchlaufen den Kleidung-Entscheidungsbaum und markieren den Pfad in dem Baum. Anschließend ziehen sie die abgebildeten Figuren entsprechend des eingezeichneten Pfads an. Der Kreativität bei der Gestaltung der einzelnen Kleidungsstücke ist keine Grenze gesetzt, die Auswahl ist jedoch vorgegeben. Bei ausreichender Zeit können die Schüler*innen sich auch einen neuen Pfad ausdenken.

Wer fertig ist, kann sich bei Dir das zweite Arbeitsblatt **Hund Cali** abholen. Gib den Hinweis, dass jetzt der Spieß umgedreht wird und die Schüler*innen einen eigenen Entscheidungsbaum erstellen.

Wichtig: Eine textuelle Beschreibung ist nicht immer eindeutig. Es sind also verschiedene Varianten zulässig. Dennoch sollte genau überprüft werden, ob in dem Graphen keine Pfade möglich sind, die in dem Text explizit ausgeschlossen wurden.

Schnelle Schüler*innen können sich ihre eigene Geschichte ausdenken (Aufgabe 3). Welche verschiedenen Schritte und Handlungen gibt es? Wie hängen sie zusammen? Welche Bedingungen müssen erfüllt sein, damit welche Handlung ausgeführt werden soll? Wie könnte ein Entscheidungsgraph für diese Situation aussehen?

Was ist der Unterschied zwischen einem Baum und einem Graph?

Ein Baum ist ein besonderer Graph. Er ist zykelfrei. Das bedeutet, dass man keinen Pfad findet, der einen Kreis bildet. Außerdem sind die Pfade in ein Baum eindeutig. Das bedeutet, dass es zwischen zwei Knoten nur einen Pfad gibt und nicht zwei verschiedene.

Arbeitsphase

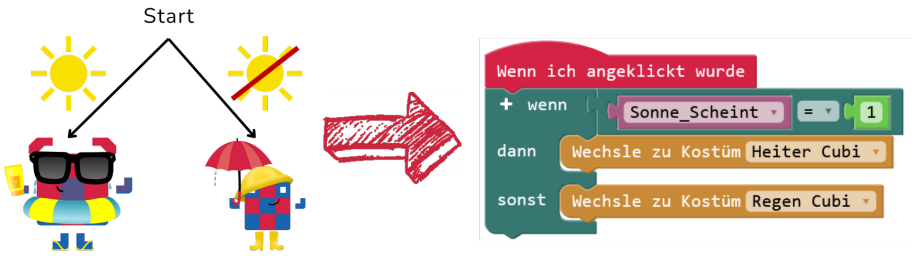
Wenn alle Schüler*innen mit dem ersten Arbeitsblatt **Kleiderschrank** fertig sind, teile das Arbeitsblatt **Wettervorhersage** aus und erkläre, dass es eine Vorbereitung und Hilfestellung für das Programmier-Level ist.
Bau mit den Schüler*innen gemeinsam die Tablets, Laptops oder Computer auf und stell sicher, dass jedes Gerät mit dem Internet verbunden ist. Lass die Schüler*inenn den Cubi-Editor über `editor.i4k.org` öffnen und stell sicher, dass alle zu dem Level **Wettervorhersage** finden.

Levelvorlage:
`level.i4k.org/wettervorhersage_sek`

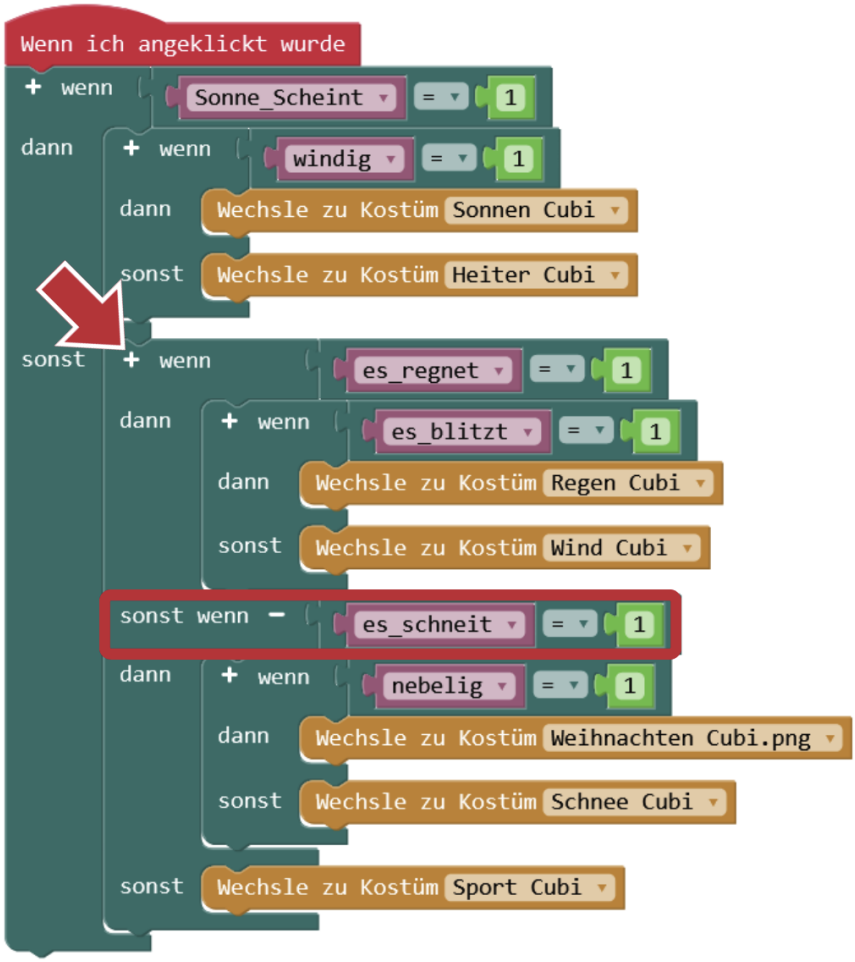
Levellösung:
`level.i4k.org/wettervorhersage_sek_lsg`

Wiederhole mit den Schüler*innen, wie sie die Kostümbibliothek öffnen und zeige ihnen, dass sie in der Kategorie **Personen** eine Unterkategorie mit Kostümen für **Cubi** finden. Aus dieser Auswahl können sie für die Bearbeitung von Aufgabe 1 auf dem Arbeitsblatt **Wettervorhersage** wählen. Gib den Schüler*innen Zeit, diese Aufgabe zu bearbeiten.
Setzt den ersten Teil des Entscheidungsbaumes gemeinsam in Cubi um. Stelle dafür zunächst das Spielprinzip vor. Nach Start ▸ des Levels können die Wettersymbole oben an der Bühne durch Klicken an- und abgewählt werden. Wenn man eine Wetterlage ausgewählt hat, kann man auf Cubi klicken und er zieht sich entsprechend für das Wetter an. Dieser Vorgang kann beliebig häufig wiederholt werden.
Eine zweifache Verzweigung wird in einen **Wenn dann sonst**-Baustein übertragen. Die Abfrage, ob eine Wetterlage auftritt oder nicht, ist etwas kniffliger. Für jedes Wetter gibt es eine **Variable**, z.B. **Sonne_Scheint** oder **neblig**. Wenn diese Variable den Wert **1** hat, tritt dieses Wetter auf. Wenn die Variable den Wert **0** hat, dann tritt es nicht auf. Die ausgemalte Cubi-Figur entspricht dem Baustein **Wechlse zu Kostüm ...**

Nehmt euch Zeit, die Verbindung von Abzweigungen im Entscheidungsbaum und dem **Wenn dann sonst**-Baustein zu erkunden. Ziehe die Schüler*innen in die Lösungsfindung mit ein und probiere ihre Vorschläge aus. Wenn etwas nicht geklappt hat, überlegt gemeinsam, was ihr programmiert habt. Erinnere an die Stunden zu Fehlersuche: „Fehler“ sind in der Programmierung Ausgangspunkt für neues Wissen. Durch Ideen, die nicht direkt zum Ziel führen, lernt ihr sehr viel mehr als durch eine fehlerfreie Programmierung beim ersten Versuch!



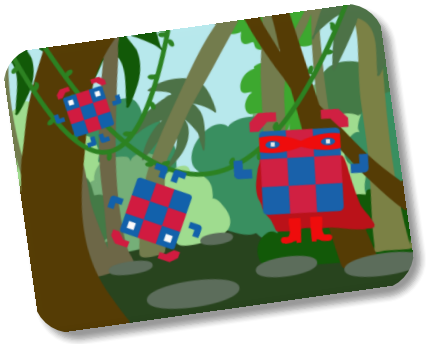
Bei so einem komplexen Level ist es wichtig, immer wieder zu testen, ob sich das Level wirklich so verhält, wie erwartet. Es gibt zwei Besonderheiten in dem Level: Zum einen werden das erste Mal mehrere **Wenn dann**-Bausteine ineinander verschachtelt. Zum anderen gibt es in dem Entscheidungsbaum im rechten Pfad eine Kreuzung mit drei Pfaden. Dafür braucht man einen **Wenn dann sonst**-Baustein mit einem zusätzlichen **sonst wenn**-Arm. Diesen kann man hinzufügen, indem man auf das Plus links oben neben dem **wenn** klickt.



Entlasse die Schüler*innen in Einzel- bzw. Partnerarbeit und lass sie den Rest des Entscheidungsbaumes implementieren. Wenn sie damit fertig sind, können sie die zweite Aufgabe auf dem Arbeitsblatt **Wettervorhersage** bearbeiten: Welche Wetterlagen haben keinen Pfad im Entscheidungsbaum? Was passiert im Level, wenn eine dieser Kombinationen von Wetterlagen ausgewählt werden? ***Lösung:*** *Cubi zieht sich nicht um.*

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.
Beende die Stunde mit einem Ausblick auf das nächste Thema.



IT 4 KIDS

Material für 1 x 45 Minuten

Struktogramm

Struktogramme helfen, die Struktur auch eines tief verschachtelten Programms schnell zu erfassen. Sie lassen sich wie ein Entscheidungsbaum lesen. Ihre Inhalte und Anweisungen erinnern an Pseudocode. Struktogramme sind somit eine weitere visuelle Darstellungsform von Algorithmen.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 5/7), **Hessen** (Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Saarland** (IKT, S1 – S2), **Sachsen** (Informatik, Jgs. 8)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.
- ... entwickeln eine eigene Meinung, treffen eigene Entscheidungen und vertreten diese gegenüber anderen.
- ... arbeiten gut mit anderen zusammen, übernehmen Aufgaben und Verantwortung in Gruppen.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... stellen einen Algorithmus grafisch dar.
- ... überführen Handlungsvorschriften in ein Struktogramm.
- ... bewerten einen als Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität.
- ... übertragen ein Struktogramm in eine grafische Programmiersprache.

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Weitere verwendete Bausteine

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

Vorbereitung

Mache Dich mit den Spielregeln von **Der gemeinen 1** und dem Cubi-Level **Expedition** vertraut. Du findest es, indem Du oben links im **Menü** ☰ auf **Öffnen** 🗀 gehst und sie anschließend unter dem Tab **Entdeckerreihe** auswählst oder die QR-Codes weiter unten scannst.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Rückblick auf das Thema **Pseudocode**	P	
15	Erarbeitung	Vorstellung von Struktogrammen als neue Darstellungsform von Algorithmen	PA, P	☐ AB **Die gemeine 1** ☐ Würfel
20	Arbeitsphase	Übertragung eines Struktogramms in das Cubi-Level **Expedition**	EA/ PA	☐ AB **Expedition mit Cubi** ☐ Laptops/Tablets/PCs ☐ ggf. KV **QR-Codes Struktogramm**
5	Präsentation und Reflexion	Vielfalt von Darstellungsformen	P	

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

Einstieg

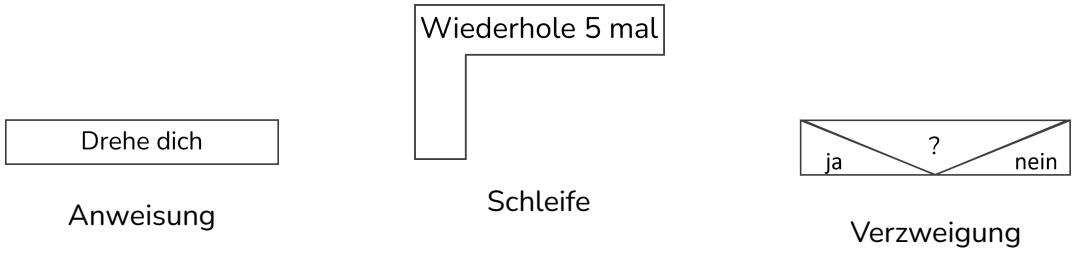
Eröffne die Stunde, indem Du das Thema der Stunde nennst: **eine neue Darstellungsform von Algorithmen kennenlernen**. Lass die Schüler*innen ihr Vorwissen aktivieren, indem Du sie fragst, welche Darstellungsformen von Algorithmen sie schon kennen:

- Grafische Programmierung
- Programmcode mit Stift und Papier, z.B. Pseudocode
- evtl. textbasierte Programmiersprachen

Greife den Pseudocode aus den genannten Beispielen heraus. In Teil 1 wurde Pseudocode eingeführt als eine Art Skizze, bevor man das eigentliche Programm schreibt. Es ist das gleiche Prinzip wie eine Stichwortliste, die sich Journalist*innen bei Recherchearbeiten machen, aus denen sie hinterher einen fertigen Artikel formulieren. In welchen Situationen ist er hilfreich? Wie unterstützt er Programmierer*innen?

Erarbeitung

In dieser Stunde soll es um eine etwas andere Darstellungsform von Algorithmen gehen: das Struktogramm. Teile das Arbeitsblatt **Die gemeine 1** aus. Lass die Schüler*innen etwas über die Spielregeln knobeln und das Spiel ausprobieren. Tragt die Spielregeln anschließend im Plenum zusammen.
Arbeitet die verschiedenen Elemente eines Struktogramms heraus. Wie werden Anweisungen dargestellt? Wie wird eine Schleife visualisiert? Wie wird eine Verzweigung abgebildet?



Anweisung | Schleife | Verzweigung

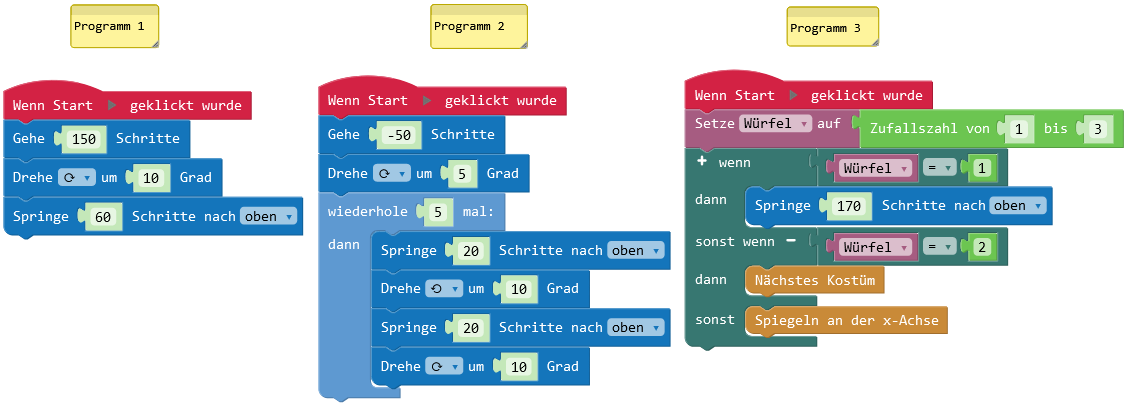
Um eine Verknüpfung der Struktogrammelemente und der Programmiererfahrung in Cubi herzustellen, können die Schüler*innen die Kontrollelemente des Struktogramms in den Farben der Programmierkonzepte in Cubi anmalen: Schleifen und Verzweigungen. Die anderen Anweisungen lassen sich nicht eindeutig zuordnen, es finden sich Anweisungen aus den Kategorien Mathe (würfeln, addieren), Variablen (Werte speichern) und Aussehen (gewonnen) etc.

Arbeitsphase

Um sich weiter mit der Darstellungsform von Struktogrammen auseinanderzusetzen, übertragen die Schüler*innen kurze Struktogramme in das Cubi-Level **Expedition**. Teile dazu das Arbeitsblatt **Expedition mit Cubi** aus und gehe mit den Schüler*innen die einzelnen Aufgaben durch. Gib ihnen dann Zeit, das Arbeitsblatt **Expedition mit Cubi** in Einzelarbeit zu bearbeiten.

Levelvorlage:
`level.i4k.org/expedition_sek`

Levellösung:
`level.i4k.org/expedition_sek_lsg`



Wenn Schüler*innen schneller fertig sind, können sie in Cubi ein eigenes kurzes Programm entwerfen, welches sie im Anschluss in ein Struktogramm übertragen. Dann bilden sie Zweiergruppen, tauschen ihre Struktogramme aus und programmieren das jeweilige Struktogramm in Cubi nach.

Präsentation und Reflexion

Vergleicht die Lösungen miteinander. Gib einzelnen Schüler*innen die Möglichkeit, ihre eigenen Programme und die zugehörigen Struktogramme der Klasse vorzustellen. Frage sie und ihre Mitschüler*innen, was ihnen leicht- und was ihnen schwergefallen ist.
Sammel zur Reflexion über die Stunde mit den Schüler*innen Vor- und Nachteile der neu gelernten Darstellungsform von Algorithmen.

- Für welche Zwecke ist sie nützlich? Ein Vorteil von Struktogrammen ist, dass sie den logischen Ablauf eines Algorithmus übersichtlich und klar strukturiert darstellen. Sie eignen sich besonders, um einfache bis mittelkomplexe Algorithmen visuell zu verdeutlichen und die einzelnen Schritte nachvollziehbar zu machen.
- Für welche Situationen sind andere bekannte Darstellungsformen für Algorithmen passender? Andere Darstellungsformen wie Entscheidungsbäume sind besser geeignet, wenn komplexe Entscheidungen oder Verzweigungen im Algorithmus visualisiert werden sollen.

Erörtert, warum es verschiedene Darstellungsformen für Algorithmen gibt. Leitet her, dass Visualisierungen hilfreich für das Verständnis sind, auch bei logischen Zusammenhängen.
Beende die Stunde mit einem Ausblick auf das nächste Thema.

Material für 1 x 45 Minuten

Funktionen

Die Königsdisziplin im Programmieren ist nicht nur, funktionierenden Code zu schreiben, sondern auch einen, der verständlich, wartbar und vor allem effizient ist. In dieser Stunde geht es um Funktionen. Durch sie wird es möglich, eine Abfolge von Bausteinen an verschiedenen Stellen im Programm wiederzuverwenden. Das erhöht die **Wartbarkeit** des Codes, weil Änderungen leichter eingefügt werden können. Außerdem kann so geschriebener Code wiederverwendet werden und der Schreibprozess wird dadurch kürzer.

Anknüpfung an Bildungspläne

Bayern (Informatik, Jgs. 7), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Nordrhein-Westfalen** (Informatik, Jgs. 5/6), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7)

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... sind motiviert, Neues zu lernen und Dinge zu verstehen, strengen sich an, um sich zu verbessern.
- ... kennen und nutzen unterschiedliche Wege, um Probleme zu lösen.

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

- ... strukturieren Handlungsabläufe in logische Teileinheiten.
- ... analysieren eine algorithmische Problemstellung, um Teilprobleme zu identifizieren.
- ... implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung.

Neue Bausteine

Funktionen:	Funktionen mit und ohne Parameter
Aussehen:	Setze Größe auf ... %, Nächstes Kostüm
Bewegung:	Springe zu x:... y:...

Weitere verwendete Bausteine

Schleifen: **Wiederhole ... mal**
Kontrolle: **Warte ... Sekunden**

Vorbereitung

Mache Dich mit den Definitionen von Funktionen vertraut und suche die Muster in dem Cubi-Level **Tanz**. 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.
Vollziehe den Unterschied zwischen Schleifen und Funktionen nach. **Weitergedacht:** Informiere Dich über die Bedeutung von Modularisierung in professionellen Softwareprojekten, z.B. mit dem Suchwort **Modulare Softwareentwicklung**.

Unterrichtsverlaufsplan

Zeit	Phase	Unterrichtsschritte	SF	Material
5	Einstieg	Definition von Modulen und Funktionen	P	
15	Erarbeitung	Mustersuche in Programmen, Funktionen in Cubi-Level **Tanz** programmieren	P, EA/ PA	☐ Präsentationstechnik ☐ Tablets/Laptops/PCs
15	Arbeitsphase	Cubi-Level **Vieleck**	EA/ PA	☐ Tablets/Laptops/PCs ☐ KV **Hilfekarten Vieleck** ☐ ggf. KV **QR-Codes Funktionen**
10	Präsentation und Reflexion	Vorstellen der Programmierergebnisse, Ausflug in professionelle Softwareprojekte	P	☐ Präsentationstechnik

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

Einstieg

Eröffne die Stunde, indem Du mit den Schüler*innen das folgende Gedankenexperiment durchspielst: Stellt euch vor, ihr wollt zusammen eine Gemüsepfanne kochen. Jeder von euch bekommt eine bestimmte Aufgabe: Zwiebeln hacken, verschiedene Gemüsesorten klein schneiden, Reis kochen, Gemüse anbraten, würzen. Worauf muss geachtet werden?

- Das Gemüse sollte ungefähr gleich groß geschnitten werden, damit es gleichzeitig gar ist.
- Wenn jemand die Aufgabe **Zwiebeln hacken** übernimmt, sollte die Zwiebel am Ende der Aufgabe auch gehackt sein und nicht nur der Stängel abgeschnitten sein.
- Die Reihenfolge der Arbeitsschritte sollte eingehalten werden.
- Wenn jemand das Gemüse würzen soll, muss er/sie wissen, welche Gewürze schon verwendet wurden.

Stelle den Bezug zur Informatik her und erkläre die folgenden Sachverhalte. An großen Softwareprojekten, wie z.B. Google, WhatsApp oder TikTok, arbeiten hunderte Entwickler*innen an einem Projekt. Der Quellcode kann mehrere Millionen Zeilen haben. Um nicht den Überblick zu verlieren, müssen sich die Teams gut absprechen, wer welche Funktionalitäten implementiert und genau definieren, wie andere mit ihrem Stück Programm weiterarbeiten können – genau wie in eurem Kochprojekt.
Ein wichtiges Werkzeug hierfür sind **Funktionen**: eine Reihe von Anweisungen, die man einmal schreiben muss und dann an mehreren Stellen im Programm verwenden kann.

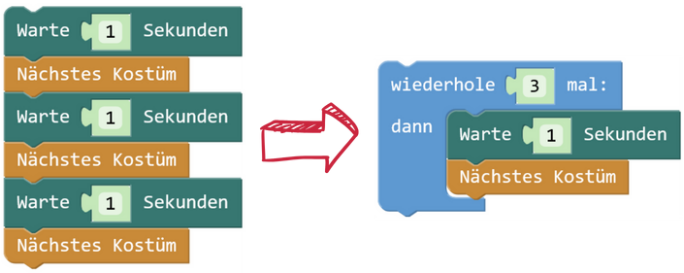
Erarbeitung

Leite dazu über, dass ihr euch nun **Funktionen** in Cubi anschaut. Öffne das Cubi-Level **Tanz** über eine digitale Tafel oder ähnliche Präsentationstechnik. Wie bewegt sich die Tänzerin? Welche Muster erkennen die Schüler*innen? Wie könnte man das Programm verkürzen? An welchen Stellen hilft eine **Schleife** und an welchen nicht? Warum? Verkürzt das Programm mit drei **Wiederhole 3 mal**-Schleifen.

Levelvorlage:
`level.i4k.org/tanz_sek`

Levellösung:
`level.i4k.org/tanz_sek_lsg`

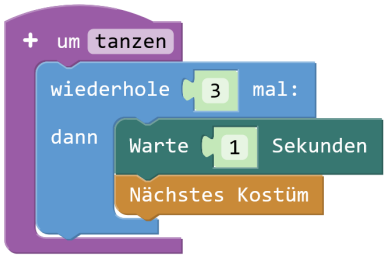
Lösung: *Die Bausteine **Warte … Sekunden** und **Nächstes Kostüm**, die direkt untereinanderstehen, können mit einer **Wiederhole … mal**-Schleife zusammengefasst werden.*



Weiter kann das Programm **Tanz** mit **Schleifen** nicht verkürzt werden, obwohl die Bausteine **Wiederhole 3 mal**, **Warte 1 Sekunde** und **Nächstes Kostüm** dreimal exakt gleich

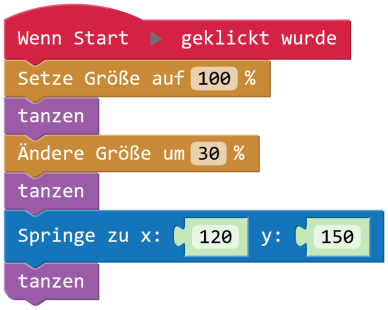
in dem Programm vorkommen. Das liegt daran, dass die Bausteine zwischen den Schleifen unterschiedlich sind und nicht in die Wiederhole ... mal-Schleife hineingezogen werden können.

Was eine Schleife hier nicht lösen kann, kann eine Funktion. Ziehe den Funktionsbaustein Um etwas tun in die Programmierfläche. Er steht für sich allein und wird nicht mit den bisherigen Bausteinen verbunden.

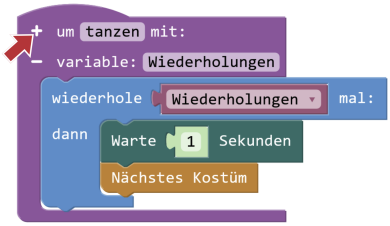


Ändere den Namen der Funktion von **etwas tun** zu **tanzen**, indem Du auf das helle Feld klickst. Löse eine der Schleifen aus dem großen Programm und ziehe sie in den Funktionsbaustein **tanzen**.

Öffne erneut die Kategorie Funktionen. Hier befindet sich nun ein neuer Befehlsbaustein mit dem Namen eurer Funktion tanzen. Ziehe ihn an die Stelle im Programm, wo vorher die Schleife war. Drücke auf **Start** ▶ und beobachtet, wie die Tänzerin genau das Gleiche macht wie vorher. Tausche nun auch die anderen Schleifen gegen den neuen Funktionsbaustein aus. Auch jetzt tanzt die Tänzerin so wie vorher, jedoch ist das Programm deutlich kürzer und einfacher zu lesen.



Baut gemeinsam die Geräte auf. Gib den Schüler*innen etwas Zeit, das Programm und das Erstellen der Funktionen für sich selbst zu wiederholen.



Richte anschließend die Aufmerksamkeit der Schüler*innen wieder nach vorne und stelle eine Besonderheit von Funktionen vor: **Parameter** oder auch **Übergabewerte** genannt. Sie werden wie eine Variable in der Funktion benutzt und bekommen ihren Wert beim Aufruf der Funktion. Füge einen Parameter zu der Tanz-Funktion hinzu, indem Du auf das **Plus** im Funktionskopf drückst. Ändere den Namen des Parameters von **x** zu **Wiederholungen**. Öffne die Kategorie Variablen und ziehe den Variablenbaustein Wiederholungen in das Zahlenfeld der Wiederhole ... mal-Schleife. Füge als letztes einen Zahlen-Baustein aus der Kategorie Mathe in den Funktionsaufruf und puzzle ihn an den Parameter **Wiederholungen**. Trage verschiedene Werte für die Wiederholungen ein und starte das Programm, z.B. **3** oder **5**.

Gib den Schüler*innen etwas Zeit, das Hinzufügen von Parametern für sich selbst zu üben. Schnelle Schüler*innen können die **Wartezeit** als zweiten Parameter hinzufügen und in den Warte-Baustein integrieren.

Was beobachten die Schüler*innen? Welche Vorteile sehen sie in der Verwendung von Parametern? Die Tänzerin kann bei jedem Funktionsaufruf unterschiedlich viele Schritte tanzen, ohne, dass der Wert in der Schleife verändert werden muss.

Deklaration und Aufruf

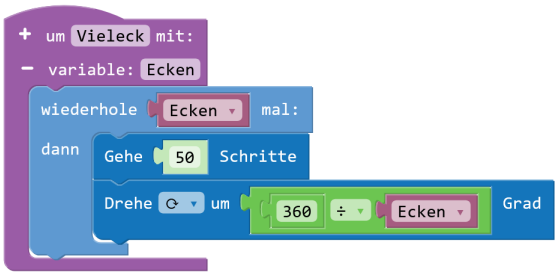
Eine Funktion zu definieren oder auch zu deklarieren macht man genau einmal in einem Programm pro Funktion. Danach kann man sie beliebig oft aufrufen und ausführen.

Arbeitsphase

Formuliere den Arbeitsauftrag für die Arbeitsphase: Die Schüler*innen öffnen das Level **Salto** und schreiben eine Funktion, welche als Parameter eine Zahl nimmt und dann ein Vieleck zeichnet, das so viele Ecken hat, wie durch den Parameter vorgegeben werden.

Levelvorlage:
`level.i4k.org/salto_sek`

Levellösung:
`level.i4k.org/salto_sek_lsg`



Die Schüler*innen erstellen eine neue Funktion und geben ihr den Namen Vieleck. Sie fügen über das **Plus** einen neuen Parameter hinzu und benennen ihn als **Ecken** oder **Kanten**. Die Funktion Vieleck besteht aus einer Wiederhole ... mal-Schleife und einem Gehe- und einem Drehe-Baustein. Durch das Hinzufügen des Parameters wurde eine neue Variable mit dem Namen **Ecken** erstellt. Diese Variable Ecken gibt die Anzahl der Wiederholungen für die Schleife an. Außerdem ergibt sich durch die Division des Vollwinkels 360° durch den Parameter Ecken die Gradzahl für die Drehung.

Als Hilfestellung kannst Du einzelnen Schüler*innen auf eigenen Wunsch Tippkarten austeilen. Als Knobelaufgabe können die Schüler*innen einen zweiten Parameter **Länge** hinzufügen und die zugehörige Variable Länge in den Gehe-Baustein ziehen. Wer möchte, kann auch die Länge abhängig von der Anzahl der Ecken berechnen.

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.
Leite ein Unterrichtsgespräch an, in dem die Schüler*innen Vermutungen anstellen können, wofür Funktionen in großen Softwareprojekten genutzt werden können. Eine Auswahl an Einsatzmöglichkeiten ist:

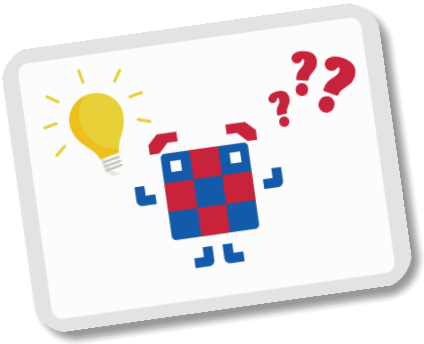
- Mehrere Programmierer*innen können an einem Programm gleichzeitig schreiben, weil jeder seine eigenen Funktionen schreibt und diese hinterher zusammengefügt werden.
- Es geht schneller, wenn man Programmcode nicht doppelt schreiben muss, sondern schon geschriebenen Programmcode wiederverwenden kann.

- Wenn später etwas an der Funktion geändert werden muss, muss man es nur an einer Stelle ändern und nicht an allen.

  Weitergedacht: Was passiert, wenn man etwas an den Parametern einer Funktion im Nachhinein ändert? Dann ist Vorsicht geboten, denn das ist eine häufige Fehlerquelle.

- Beim Programmieren kann man auch Funktionen nutzen, die man nicht selbst geschrieben hat, z.B. sind die Bausteine in Cubi alle Funktionen. Im Hintergrund wird Code in einer anderen Programmiersprach aufgerufen, welcher dann dazu führt, dass auf der Bühne etwas passiert.

Beende die Stunde mit einem Ausblick auf das nächste Thema.



Material für 90 Minuten

Eigenes Spiel

In diesem Abschluss des zweiten Teils der Unterrichtsreihe 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. Damit sie nicht die einzigen sind, die ihre Implementierung nachvollziehen können, fertigen sie als letzten Schritt eine Dokumentation an. Eine Dokumentation ist wie eine Bedienungsanleitung für ein Programm. Anders als die Kommentare, welche Teil des Quellcodes sind, bildet die Dokumentation ein eigenes, leicht zugängliches Dokument oder Wiki bei größeren Programmierprojekten.

Anknüpfung an Bildungspläne

Baden-Württemberg (Aufbaukurs Informatik, Jgs. 7), **Bayern** (Informatik, Jgs. 7; Informationstechnologie, Anfangsunterricht), **Berlin/Brandenburg** (Wahlpflichtfach Informatik, Jgs. 7), **Hamburg** (Informatik, Jgs. 7), **Hessen** (Wahlfach Informatik, Jgs. 7), **Mecklenburg-Vorpommern** (Informatik und Medienbildung, Jgs. 6), **Niedersachsen** (Informatik, Jgs. 5 – 7), **Saarland** (IKT, S3), **Schleswig-Holstein** (Informatik, Jgs. 5 – 7)

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.
- ... dokumentieren gemeinsam ihren Arbeitsprozess und ihre Ergebnisse auch mithilfe digitaler Werkzeuge.

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

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	
35	Arbeitsphase I	Implementierung der eigenen Level	EA/ PA/ GA	☐ Tablets/Laptops/PCs ☐ AB **Meine Level** ☐ ggf. KV **QR-Codes Eigenes Spiel 2**
20	Arbeitsphase II	Dokumentation zum eigenen Level	EA/ PA/ GA	☐ AB **Unsere Dokumentation** ☐ Tablets/Laptops/PCs
10	Präsentation und Reflexion	Vorstellen der Level, Thematisierung von Herausforderungen und Erkenntnissen	P	☐ Tablets/Laptops/PCs
5	Abschluss	Schlussrunde zum Thema **Programmieren**	P	

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 über das **Menü**, **Öffnen** und den Tab **Entwicklerreihe** die Vorlage **Eigenes Spiel 2** 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 Schüler*innen, wie sie an das Projekt drangehen wollen. Erinnere sie daran, dass es wichtig ist, ein Projekt im Vorfeld zu planen und zu skizzieren, bevor wild drauflosprogrammiert wird. Folgende Planungswerkzeuge können die Schüler*innen verwenden:

- Einen Pseudocode aufschreiben.
- Ein Struktogramm entwerfen.
- 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.

Levelvorlage:
`level.i4k.org/eigenes_spiel2_sek`

Erarbeitung

Die Schüler*innen planen ihr Projekt zunächst mit Stift und Papier. Dafür erstellen sie Stichpunkte, Skizzen, Struktogramme und Pseudocodes für ihr Level.

Arbeitsphase I

Die Schüler*innen implementieren ihre Levelkonzepte in Cubi. Bau mit den Schüler*innen gemeinsam die Tablets, Laptops oder Computer auf und stell sicher, dass jedes Gerät mit dem Internet verbunden ist. Lass sie den Cubi-Editor über `editor.i4k.org` öffnen und stelle sicher, dass alle zu der Vorlage **Eigenes Spiel 2** finden.
Erinnere die Schüler*innen daran, dass man sich in Cubi mithilfe der Kommentarfunktion Notizen machen kann, die das Programm nicht beeinflussen. Dafür müssen sie über die rechte Maustaste auf dem Computer oder durch langes Drücken auf die Programmierfläche am Tablet ein Menü öffnen und **Kommentar hinzufügen** wählen.
Gehe durch die Klasse und hilf bei Fragen und Problemen. Mache die Schüler*innen bei Bedarf darauf aufmerksam, dass sie ihr Levelkonzept während der Umsetzung noch anpassen können.

Am Ende ist es ganz wichtig, dass sie ihr Level online speichern, damit sie es in der nächsten Stunde weiter bearbeiten können. 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.
Um ihren alten Stand später wieder zu öffnen, kicken sie im **Menü** auf **Öffnen** und geben in das Feld rechts ihr Codewort ein. Dann klicken sie auf **Level mit Codewort laden**. Nun öffnet sich ihr Level. Wenn sie weiter programmieren, müssen sie am Ende ihr Level erneut abspeichern und sich das neue Codewort notieren.

Arbeitsphase II

Als letzten Schritt der Implementierung sollen die Schüler*innen eine Dokumentation zu ihrem Level erstellen. Motiviere die Aufgabe, indem Du erklärst, dass eine Dokumentation nützlich ist, damit andere ein selbstgeschriebenes Level spielen können, ohne dass die/der Entwickler*in es persönlich erklären muss. Auch hilft es den Schüler*innen selbst, nicht den Überblick über ihr Programm und dess Funktionalitäten zu verlieren.
Für einen einfachen Erstellungsprozess der Dokumentation teile das Arbeitsblatt **Unsere Dokumentation** aus. Gib den Schüler*innen Zeit, das Arbeitsblatt in Ruhe auszufüllen. Wenn genügend Zeit ist, können sie ihr Level und ihre Dokumentation einer/m Mitschüler*in zeigen und auf Verständlichkeit und Vollständigkeit prüfen lassen.

Präsentation und Reflexion 

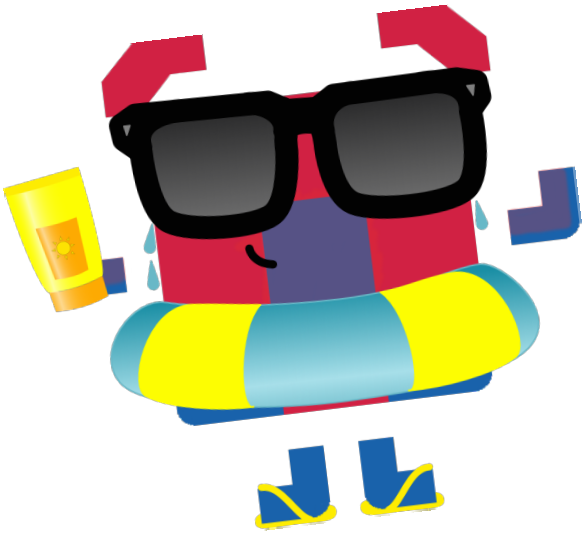
Für die Präsentation der Ergebnisse kann eine kleine Ausstellung im Klassenraum veranstaltet werden. Dazu werden die Dokumentationen und die Planungsnotizen ausgelegt sowie die Level auf den Tablets oder Computern geöffnet. Lass die Schüler*innen frei durch die Klasse laufen, die Level zeigen, betrachten 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? Wie hat die Dokumentation ihnen bei der Vorbereitung der Präsentation geholfen?

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 2 der Lernreihe** geschafft! Was eine tolle Leistung!



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

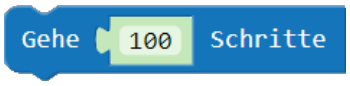
Baustein-Lexikon

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.

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



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.

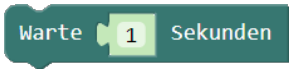
Mithilfe des **Springe zu x: ... y: ...** kann eine Koordinate festgelegt werden, zu der die Figur springen soll. Die Koordinaten können unterhalb der Bühne abgelesen werden.

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.

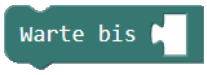
Kontrolle

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.



Gelangt ein Programm zu einem **Warte bis**-Baustein, dann bleibt es hier solange stehen, bis die Bedingungen erfüllt ist. Andere Programmteile der Figur, die ihren eigenen **Startbaustein** haben, werden hierdurch nicht unterbrochen. Erst wenn die Bedingung erfüllt 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.

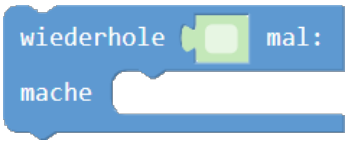
Mathe

Mit diesem Baustein kann ein Bereich definiert werden, aus dem eine zufällige Zahl gewürfelt wird. Die Zahlen können beliebig hoch sein, die zweite Zahl muss allerdings größer als die erste Zahl sein.

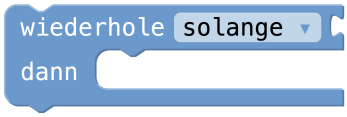
Variablen

Über die Kategorie **Variablen** können Variablen erstellt, umbenannt und gelöscht werden. Variablen können oben links im Vorschaufenster angezeigt oder wieder versteckt werden. **Setze *Variable* auf** ändert den Wert der Variable auf den angegebenen Wert. Mit **Erhöhe *Variable* um** und **Verringere *Variable* um** kann der Wert der Variable um ein festes Intervall positiv oder negativ geändert werden kann.

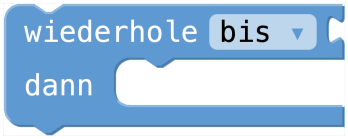
Schleifen



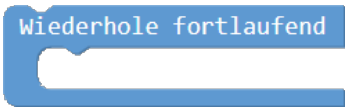
Mit der **Wiederhole ... mal**-Schleife können Bausteine, die in die Schleife eingefügt werden, wiederholt werden. Klicke auf die Zahl, um die Anzahl der Wiederholungen zu ändern.



Die **Wiederhole solange**-Schleife ist äquivalent zu **Wiederhole bis nicht**. Bei jedem Durchlauf wird geprüft, ob die Schleifenbedingung erfüllt ist. Solange die Bedingung erfüllt ist, werden die Bausteine in der Schleife ein weiteres Mal wiederholt. Ist die Bedingung nicht mehr erfüllt, wird die Schleife verlassen und die Bausteine nach der Schleife ausgeführt.



An die **Wiederhole bis**-Schleife kann ein Baustein aus den Kategorien **Fühlen** oder **Kontrolle** als Schleifenbedingung angepuzzelt werden. Bei jedem Durchlauf wird geprüft, ob diese Schleifenbedingung erfüllt ist. Bis die Bedingung erfüllt ist, werden die Bausteine in der Schleife ein weiteres Mal wiederholt. Sobald die Bedingung erfüllt ist, wird die Schleife verlassen und die Bausteine nach der Schleife ausgeführt.



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

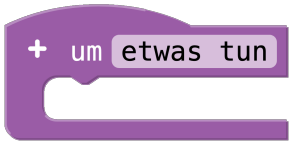
Mithilfe dieses Bausteins wechselt die entsprechende Figur das Kostüm und ändert so ihr Aussehen. Die Kostüme sind unterhalb der Bühne zu sehen. Hier wird zu dem Kostüm gewechselt, welches in dem Feld ausgewählt wurde.

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.

Mithilfe dieses Bausteins ändert die entsprechende Figur ihre Größe. Der Wert kann geändert werden. Je größer der Wert, desto größer wird die Figur.

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.

Funktionen



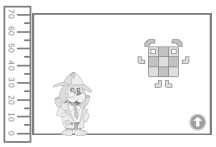
Mit Hilfe der Bausteine der Kategorie **Funktionen** kann man seine eigenen Bausteine erstellen. Dafür schreibt man ein Mini-Programm in den Baustein **Um etwas tun** und ändert den Namen in eine eigene Überschrift. Immer wenn der eigene Baustein ausgeführt wird, also in einem Programm unter einem **Startbaustein** steht, werden genau die Befehle ausgeführt, die man bei **Um etwas tun** eingegeben hat. Das ist ins Besondere dann hilfreich, wenn eine bestimmte Abfolge von Bausteinen mehrfach an unterschiedlichen Stellen eines Programms vorkommt.

Name: _____ Datum: _____

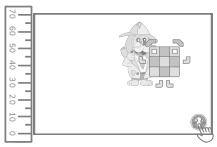
Detektivsuche

Aufgabe 1: Betrachte die beiden Bilder. Du bist der Detektivhund. Welche Aussagen sind wahr (✓) und welche sind falsch (✗)? Markiere sie.

1. Berühre ich Cubi?
2. Berühre ich den Rand?
3. Ist die Entfernung zum Rand oben kleiner als 20?
4. Ist die Entfernung zum Rand oben größer als 30?
5. Ist die Taste *Pfeil nach oben* gedrückt?



1. Berühre ich Cubi?
2. Berühre ich den Rand?
3. Ist die Entfernung zum Rand oben kleiner als 20?
4. Ist die Entfernung zum Rand oben größer als 70?
5. Ist die Taste *Pfeil nach oben* gedrückt?



Aufgabe 2: Aussagen und Bedingungen können mit **und** und **oder** verknüpft werden. Betrachte das **obere** Bild. Markiere auch die folgenden Aussagen als wahr (✓) oder falsch (✗).

1. Berühre ich den Rand und berühre ich Cubi?
2. Ist die Entfernung zum Rand oben kleiner als 100 und größer als 10?
3. B
4. B

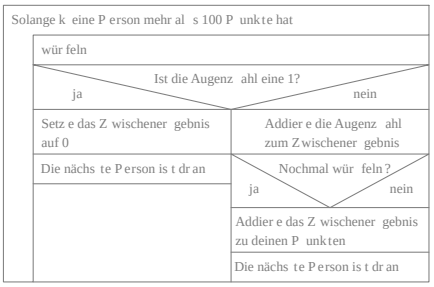
Auf

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

Nam

Die Gemeine 1

Aufgabe 1: Schaut euch das Struktogramm für einen Zug des Spiels **Die Gemeine 1** an. Welche Spielregeln könnt ihr daraus ableiten?



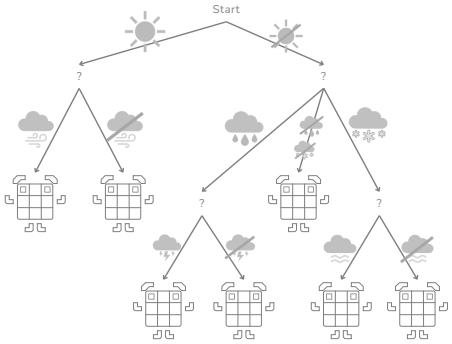
Aufgabe 2: Es spielen zwei Personen gegeneinander. Erweitere das Struktugramm um die Überprüfung, wer von den beiden gewonnen hat.

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/

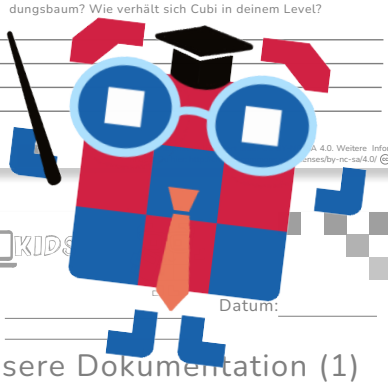
Name: _____ Datum: _____

Wettervorhersage

Aufgabe 1: Ziehe Cubis passende Kleidung an. Schau in der Kostümbibliothek nach, welche Varianten es von Cubi gibt.



Aufgabe 2: Welche Wetterkombinationen haben keinen Pfad im Entscheidungsbaum? Wie verhält sich Cubi in deinem Level?



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/

Namen: _____ Datum: _____

Unsere Dokumentation (1)

Level: _____

Aufgabe 1: Beschreibt die Geschichte eures Levels.

Jahreszahl: _____

Ort: _____

Handlung: _____

Szene aus dem Level

Aufgabe 2: Charakterisiert die Figuren eures Levels

Name: _____

Alter: _____ Steuerbar: ☐ ja ☐ nein

Fähigkeiten: _____

Aufgabe: _____

Name: _____

Alter: _____ Steuerbar: ☐ ja ☐ nein

Fähigkeiten: _____

Aufgabe: _____

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? Neben einer Modulreihe zu den Themen **Sequenzen**, Schleifen und **Verzweigungen** mit der Zielgruppe **Orimarstufe** bieten wir 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/teil2