



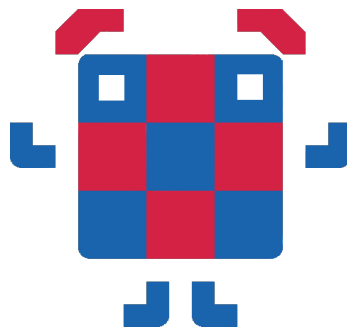
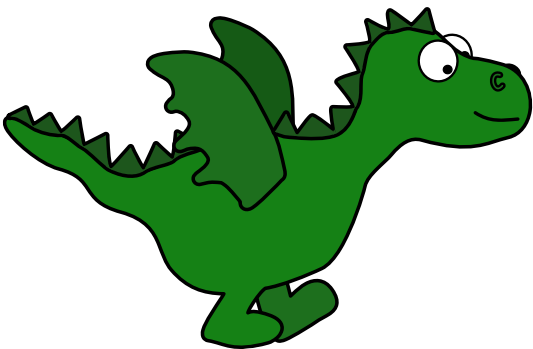
5. – 7. Klasse

Teil 2

Informatik erleben

– Variablen –

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



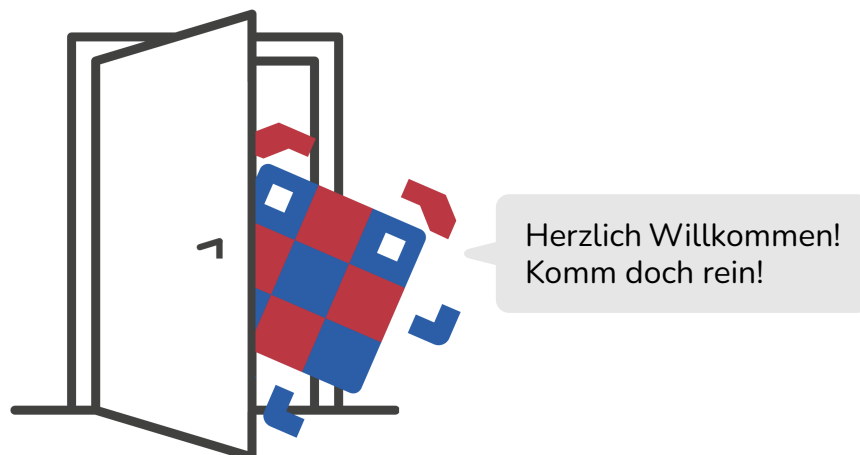
Herzlich Willkommen zu unserer Lernreihe

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

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

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

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



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



Informatik als Fachunterricht in der Sekundarstufe I

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

Schon gewusst?

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

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

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

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

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

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

Verankerungen von Inhalten zu Algorithmen in Bildungsplänen

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


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

Bundesland	Unterrichtseinheit								
	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

Standardalgorithmen



45 Minuten
(un)plugged

Variablen



90 Minuten
plugged

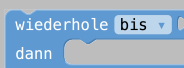
Seite 6

Fehlersuche & Testen



45 Minuten
plugged

Schleifen



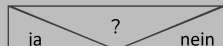
45 Minuten
plugged

Verzweigungen



90 Minuten
plugged

Struktogramm



45 Minuten
plugged

Funktionen



45 Minuten
plugged

Eigenes Spiel



90 Minuten
plugged



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)**

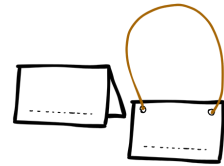
Schleifen: **Wiederhole bis**

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 Schnellspielvariante. Bereite für die normale Variante die Kopiervorlage **Variablenreise** vor, indem Du die beiden Seiten mit den Zahlen- und Variablenschildern ausdrückst, 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	<input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> Variablen-Comic
10	Erarbeitung	Zuordnungsspiel mit Datentypen	GA	<input type="checkbox"/> KV Datentypen
25	Arbeitsphase	Würfelspiel mit Variablen	GA	<input type="checkbox"/> Würfel <input type="checkbox"/> Spielfiguren <input type="checkbox"/> Spielplan <input type="checkbox"/> 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 knifflig. 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 Spielfiguren 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	<input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> Tablets/Laptops/PCs
25	Arbeitsphase	Stationenarbeit zu Cubi-Leveln	EA	<input type="checkbox"/> KV Deckblätter <input type="checkbox"/> KV Tippkarten <input type="checkbox"/> Präsentationstechnik <input type="checkbox"/> Tablets/Laptops/PCs
10	Sicherung und Reflexion	Vorstellen der Programmiererergebnisse, Beispiele für Programme mit Variablen, Nutzen und Limitationen von Datentypen	P	<input type="checkbox"/> 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.

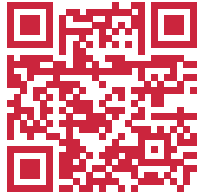
Was kann man mit einer Variable machen?

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



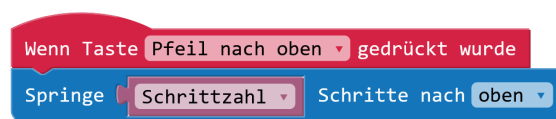
Levellösung:
level.i4k.org/tiefsee_sek_lsg

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 **Sprünge**-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

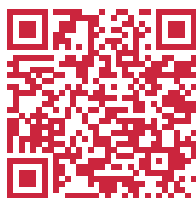
Schüler*innen auf, dass die Schritte immer kleiner werden? Sehen sie, dass sich die Zahl hinter **Schrittzahl** auf der Bühne immer weiter verkleinert?

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.



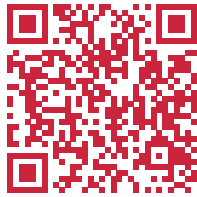
Levelvorlage:

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...

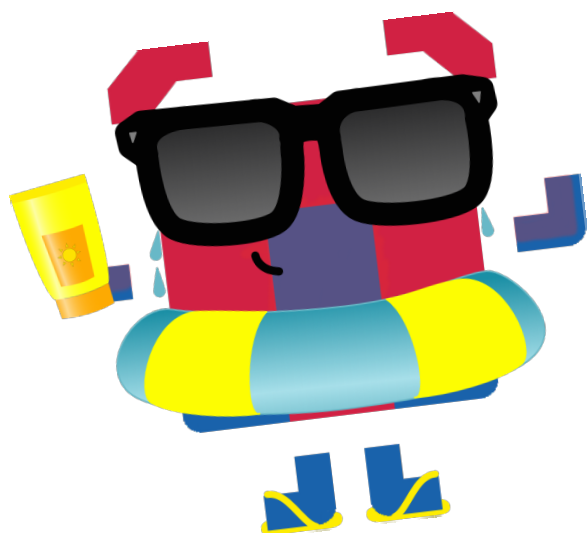
Wiederholt die drei verschiedenen Datentypen, die ihr für Variablen kennen gelernt habt: Integer (Ganzzahl), String (Wort) und Boolean (wahr/falsch). Warum kann es sinnvoll sein, Variablen je nach ihrem Datentyp unterschiedlich zu behandeln?

- 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.

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

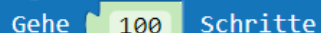
Der erste Baustein eines Blocks aus mehreren Bausteinen ist immer ein **Starbaustein**. Ein Programm einer Figur kann beliebig viele **Startbausteine** haben. **Startbausteine** zeichnen sich durch die Rundung am oberen Teil aus. Diese sagt aus, dass **Startbausteine** nicht an andere Bausteine angehängt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



Wenn Start ▶ geklickt

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

Bewegung



Gehe 100 Schritte

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



Springe zu x: 0 y: 0

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.



Drehe rechts um 90 Grad

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

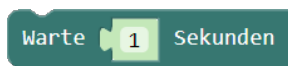
Kontrolle



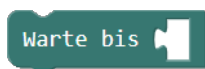
Der Baustein **Wenn dann** leitet eine Verzweigung ein. Oben an das **wenn** wird eine Bedingung angepuzzelt. Diese ist entweder **wahr** oder **falsch**. Wenn die Bedingung wahr ist, werden die Bausteine, die neben dem **dann** stehen ausgeführt. Ist die Bedingung falsch, also nicht erfüllt, werden 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.



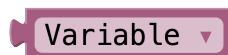
Mit diesem Baustein kann die Figur reagieren, wenn sie etwas berührt. Durch Klicken auf den kleinen Pfeil kann ausgewählt werden, ob die Figur auf den Rand oder eine andere Figur reagieren soll, wenn es noch weitere Figuren in dem Level gibt.

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

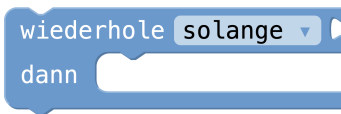


Über die Kategorie **Variablen** können Variablen erstellt, umbenannt und gelöscht werden. Variablen können oben links im Vorschauenfenster angezeigt oder wieder versteckt werden. **Setze Variable auf** ändert den Wert der Variable auf den angegebenen Wert. Mit **Erhöhe Variable um** und **Verringere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

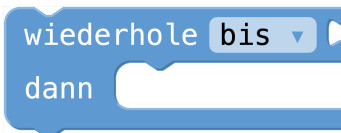
Schleifen



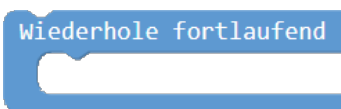
Mit der **Wiederhole ... mal**-Schleife können Bausteine, die in die Schleife eingefügt werden, wiederholt werden. Klicke auf die Zahl, um die Anzahl der Wiederholungen zu ändern.



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



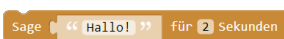
Mithilfe dieses Bausteins wechselt die entsprechende Figur das Kostüm und ändert so ihr Aussehen. Die Kostüme sind unterhalb der Bühne zu sehen. Hier wird zu dem Kostüm gewechselt, welches in dem Feld ausgewählt wurde.



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.

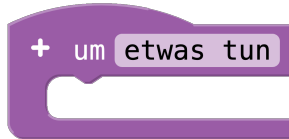


Mithilfe dieses Bausteins ändert die entsprechende Figur ihre Größe. Der Wert kann geändert werden. Je größer der Wert, desto größer wird die Figur.



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.

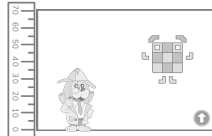


Name: _____ Datum: _____

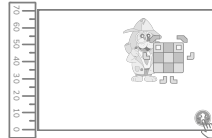
Detektivsuche

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

- Berühre ich Cubi?
- Berühre ich den Rand?
- Ist die Entfernung zum Rand oben kleiner als 20?
- Ist die Entfernung zum Rand oben größer als 30?
- Ist die Taste Pfeil nach oben gedrückt?



- Berühre ich Cubi?
- Berühre ich den Rand?
- Ist die Entfernung zum Rand oben kleiner als 20?
- Ist die Entfernung zum Rand oben größer als 70?
- Ist die Taste Pfeil nach oben gedrückt?



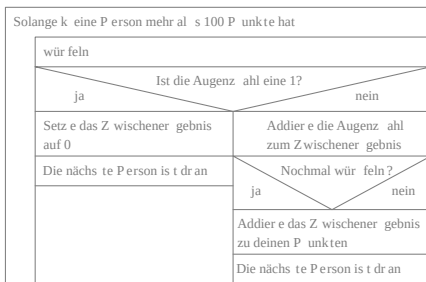
Aufgabe 2: Aussagen und Bedingungen können mit **und** oder **oder** verknüpft werden. Betrachte das obere Bild. Markiere auch die folgenden Aussagen als wahr (✓) oder falsch (x).

- Berühre ich den Rand und berühre ich Cubi?
- Ist die Entfernung zum Rand oben kleiner als 100 und größer als 10?
- Ist die Entfernung zum Rand oben kleiner als 100 oder größer als 10?
- Ist die Entfernung zum Rand oben kleiner als 100 oder größer als 100?

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

Die Gemeine 1

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



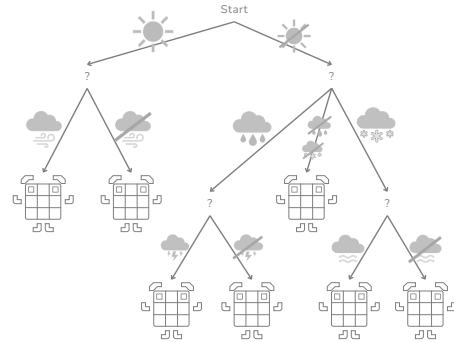
Aufgabe 2: Es spielen zwei Personen gegeneinander. Erweitere das Struktogramm um die Überprüfung, wer von den beiden gewonnen hat.



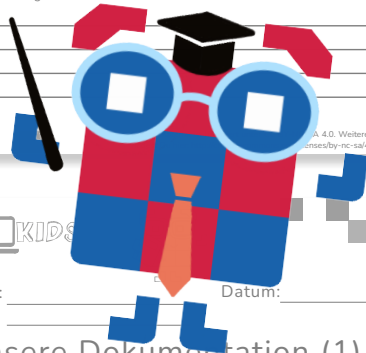
Name: _____ Datum: _____

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?



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: _____

Weitere Angebote von IT4Kids

Du willst weiter mit IT4Kids arbeiten? Neben einer Modulreihe zu den Themen **Sequenzen**, **Schleifen** und **Verzweigungen** mit der Zielgruppe **Orimiarstufe** 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 ungeschlüssig? 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. 💬

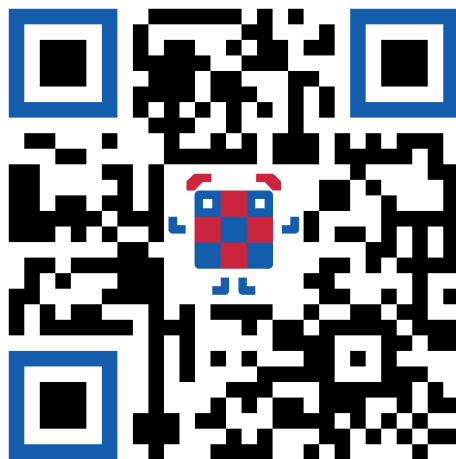
Wir freuen uns auf Dich! 😊



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