



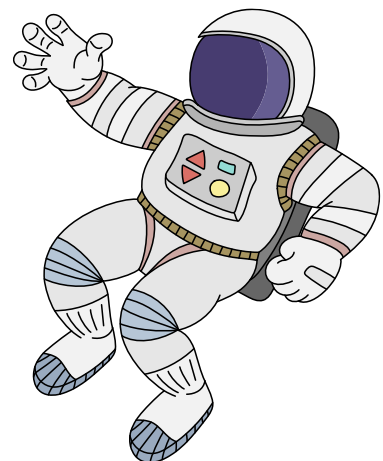
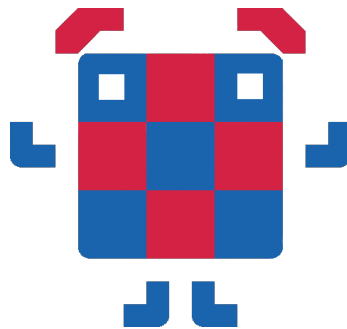
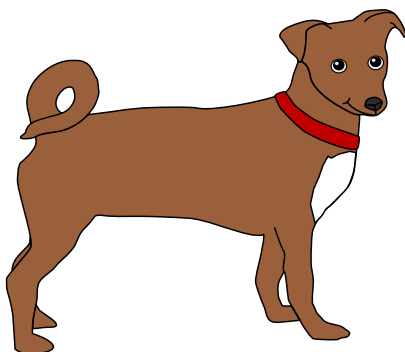
5. – 7. Klasse

Teil 1

Informatik erleben

– Fehlersuche –

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



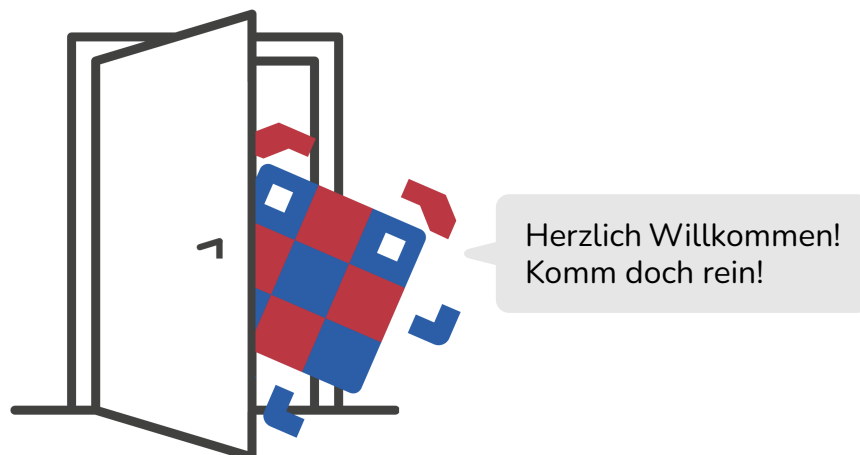
Herzlich Willkommen zu unserer Lernreihe

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

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

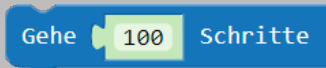
Inhaltsverzeichnis

Algorithmen



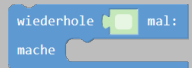
90 Minuten
unplugged

Sequenzen



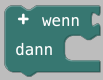
90 Minuten
plugged

Schleifen




45 Minuten
plugged

Verzweigungen



45 Minuten
plugged

Pseudocode



45 Minuten
unplugged

Fehlersuche & Testen




45 Minuten
plugged


Seite 6

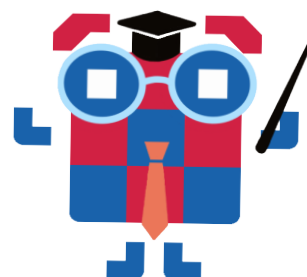
Eigenes Spiel



90 Minuten
plugged

 = plugged

 = unplugged





Fehlersuche und Testen

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

Anknüpfung an Bildungspläne

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

Überfachliche Kompetenzen

Die Schülerinnen und Schüler...

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

Fachliche Kompetenzen

Die Schülerinnen und Schüler...

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

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

Neue Bausteine

Es werden keine neuen Bausteine eingeführt.

Weitere verwendete Bausteine

Schleifen: **Wiederhole fortlaufend**

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

Vorbereitung

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

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

Algorithmus 1 :

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

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

Unterrichtsverlaufsplan

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

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

Einstieg

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

Betone, dass Fehler in Programmen etwas ganz Alltägliches sind. Programmierer*innen verbringen viel Zeit damit, Fehler in Programmen zu suchen und zu beheben. Sogar in der Raumfahrt kommen Programmierfehler vor. Ein bekanntes Beispiel ist das der ersten **Ariane 5**-Rakete, die kurz nach dem Start abgestürzt und explodiert ist, weil es einen Fehler im Programm gab, den zuvor keiner gesehen hatte. Mehr Informationen findest Du im Artikel *Darum explodierte die Ariane 5-Rakete bei ihrem Jungfernflug* von David Beck des SWR¹. Zeige den Schüler*innen das Beispiel mithilfe des YouTube-Videos *Vor 20 Jahren: Der Ariane-5-Absturz | SPIEGEL TV*². Weise die Schüler*innen darauf hin, dass 1996, als die Ariane 5 abgestürzt ist und das Video gedreht wurde, die Geräte, an denen Programmier*innen gearbeitet haben, ganz anders aussahen als heute. Sprecht kurz darüber, dass sogar absolute Programmier-Expert*innen, die Raketen ins Weltall schicken, nicht immun gegen Programmierfehler sind. Was hätte den Absturz der Rakete verhindern können? Durch ausführliche Tests hätte auffallen können, dass das Programm der Ariane 4 für das Nachfolgemodell nicht geeignet ist.

¹Quelle: Beck: *Darum explodierte die Ariane 5-Rakete bei ihrem Jungfernflug*. SWR Wissen: <https://www.swr.de/wissen/beruehmte-technik-fails-start-der-ariane-5-100.html> (zuletzt aufgerufen am 13.06.2024).

²Quelle: DER SPIEGEL: *Vor 20 Jahren: Der Ariane-5-Absturz | SPIEGEL TV*. YouTube: <https://www.youtube.com/watch?v=fd0x2K8ZCJs> (zuletzt aufgerufen am 03.06.2024).

Erarbeitung

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



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

Passt den Pseudocode dem Text an, sodass die Lösung aus Algorithmus 2 entsteht.



Algorithmus 2 :

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


```

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

Arbeitsphase

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

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

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



Levelvorlage:

`level.i4k.org/astronautenflug_sek`

Sicherung und Reflexion

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

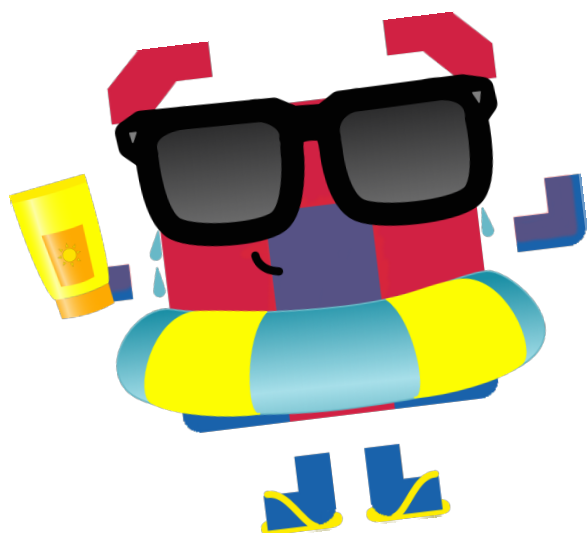
- Was macht der gelbe Astronaut Anton, wenn er die weiße Astronautin Astrid berührt? – **Er bleibt stehen und sagt: „Hallo!“**
- Was macht Astrid, wenn sie Anton berührt? – **Sie wechselt ganz schnell zwischen den Kostümen hin und her. Das liegt daran, dass das Programm immer wieder überprüft, ob Astrid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

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

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

Geschafft!

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



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

Baustein-Lexikon

Start

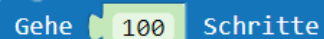
Der erste Baustein eines Blocks aus mehreren Bausteinen ist immer ein **Starbaustein**. Ein Programm einer Figur kann beliebig viele **Startbausteine** haben. **Startbausteine** zeichnen sich durch die Rundung am oberen Teil aus. Diese sagt aus, dass **Startbausteine** nicht an andere Bausteine angehängt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



Wenn Start ▶ geklickt

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

Bewegung



Gehe 100 Schritte

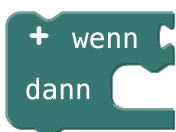
Der Baustein **Gehe ... Schritte** bewegt die Figur die entsprechende Anzahl an Pixel in die aktuelle Richtung der Figur. Im Normalfall ist dies bei Programmstart nach rechts.



Drehe rechts um 90 Grad

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

Kontrolle



+ wenn dann

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

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



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

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



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

Fühlen

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



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

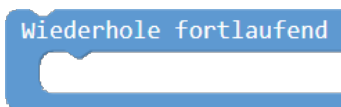


Mit diesem Baustein kann die Figur reagieren, wenn sie etwas berührt. Durch Klicken auf den kleinen Pfeil kann ausgewählt werden, ob die Figur auf den Rand oder eine andere Figur reagieren soll, wenn es noch weitere Figuren in dem Level gibt.

Schleifen



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



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

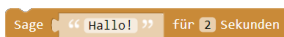
Aussehen



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



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



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

KV Kunstgalerie



Schreibe eine Anleitung für diese Tür.



Schreibe eine Anleitung für diese Brille.



Schreibe eine Anleitung für



Schreibe eine Anleitung für diesen Menschen.



Schreibe eine Anleitung für diese Sonne.



Schreibe eine Anleitung für

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

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

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

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

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

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

Damit ist ihr Job für den Raketenstart getan.



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

Namen: _____

Datum: _____

Sequenz schreiben und malen

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

Aufgabe: Befolge die Anleitung deines Partners als gelbliche Bilder?

Das habe ich gemalt: _____

Namen: _____ Datum: _____

Eigenes Level 1

Worum geht es in dem Level?

Thema: _____

Ort: _____

Geschichte: _____

Um wen geht es in dem Level?

Name: _____ Name: _____

Aufgabe: _____ Aufgabe: _____

Steuerbar:

Nein

Ja, und zwar so: _____

Steuerbar:

Nein

Ja, und zwar so: _____

Aussehen:

Aussehen:

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

Weitere Angebote von IT4Kids

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

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

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

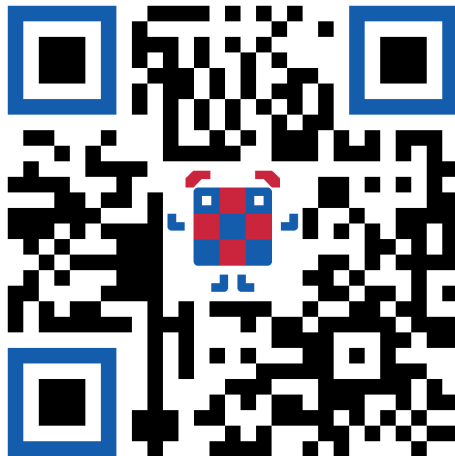
Wir freuen uns auf Dich! 😊



Hilf uns, besser zu werden!

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

Wir freuen uns über Dein Feedback:



feedback.i4k.org/lk/sek1/teil1