

Grafische Programmiersprachen

Kerstin Reese
kerstin.reese@uni-saarland.de



UNIVERSITÄT
DES
SAARLANDES

SIC Saarland Informatics
Campus

Vorstellungsrunde

Wer sind wir?

Wer sind Sie?

InfoLab Saar



Veranstaltungen
für Studierende



Teilnahme an
Wettbewerben



Kurse für Kinder
und Jugendliche



Neigungsgruppen
an Grundschulen



Besuche von
Schüler*innen



Präsenz an
Messen und
Festen



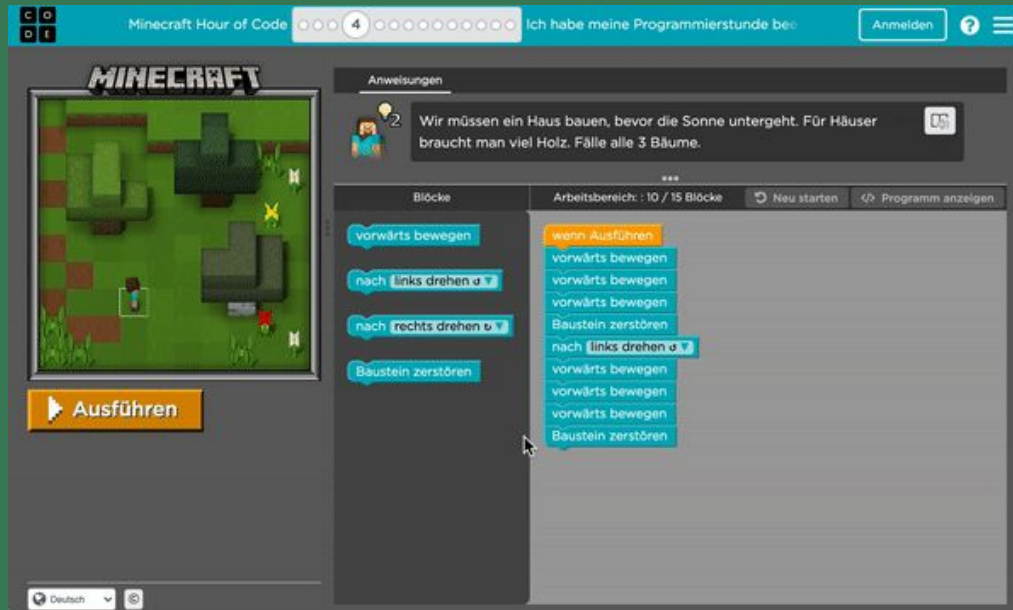
Lehrkräfte
Fortbildungen

Studien-
information

HOUR
OF
CODE

Hour of Code

Hour of Code – Minecraft “Abenteurer”



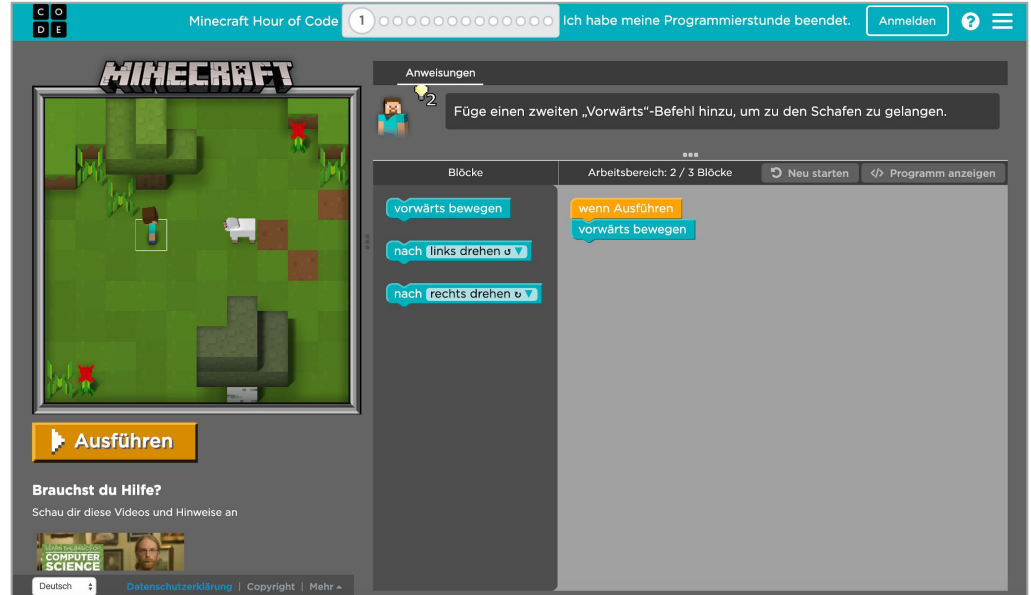
Ausprobieren:
studio.code.org/s/mc/lessons/1/levels/1



Hour of Code

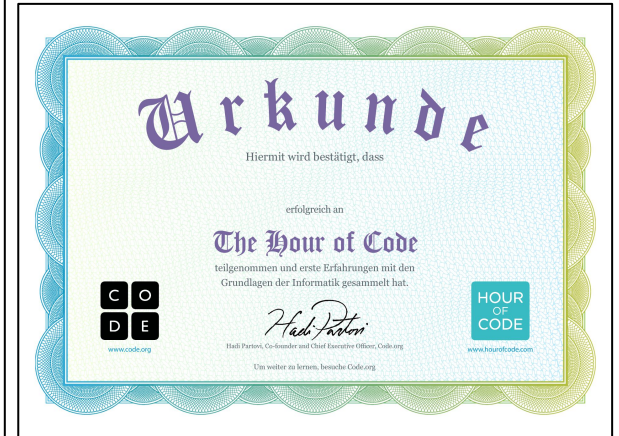


- Während der CS Education Week - jedes Jahr im Dezember
- Verfügbar in über 45 Sprachen
- Jährlich mehr als 800 Mio. SchülerInnen
- Motivierende Charaktere, z.B. Eiskönigin, Star Wars, Minecraft, ...
- code.org/minecraft



- Musterlösung Minecraft Abenteuer:
[infolab.cs.uni-saarland.de/wp-content/uploads/2023/04/Lernwerkstatt- Hour-of-Code-Stunde-des-Programmierens .pdf](https://infolab.cs.uni-saarland.de/wp-content/uploads/2023/04/Lernwerkstatt-Hour-of-Code-Stunde-des-Programmierens.pdf)

Urkunde



informatikdidaktik.cs.uni-saarland.de/hour-of-code-angebote/

Algorithmische Grundbausteine

Algorithmus

Definition:

Ein Algorithmus ist eine **Verarbeitungsvorschrift**, die aus einer endlichen Folge von eindeutig ausführbaren Anweisungen besteht, mit der man eine Vielzahl gleichartiger Aufgaben lösen kann.

Ein Algorithmus gibt an, wie **Eingabegrößen** schrittweise in **Ausgabegrößen** umgewandelt werden.

Beispiele:

- Kochrezept
- Addition mit den Fingern
- Anleitung zum Falten eines Papierflugzeugs
- Bauanleitungen, z.B. Lego
- Computerprogramm
- ...



Algorithmische Grundbausteine

Beispiel aus Level 1:



Algorithmische Grundbausteine

Sequenz (Reihenfolge):

Jede Aneinanderreihung von Befehlen ist eine Sequenz.

Beispiel aus Level 1:



Algorithmische Grundbausteine

Sequenz (Reihenfolge):

Jede Aneinanderreihung von Befehlen ist eine Sequenz.

Beispiel aus Level 1:



Beispiel aus Level 5:



Algorithmische Grundbausteine

Sequenz (Reihenfolge):

Jede Aneinanderreihung von Befehlen ist eine Sequenz.

Beispiel aus Level 1:



Schleife:

In einer Schleife werden einer oder mehrere Befehle wiederholt ausgeführt.

Beispiel aus Level 5:



Algorithmische Grundbausteine

Sequenz (Reihenfolge):

Jede Aneinanderreihung von Befehlen ist eine Sequenz.

Beispiel aus Level 1:



Schleife:

In einer Schleife werden einer oder mehrere Befehle wiederholt ausgeführt.

Beispiel aus Level 5:



Beispiel aus Level 11:



Algorithmische Grundbausteine

Sequenz (Reihenfolge):

Jede Aneinanderreihung von Befehlen ist eine Sequenz.

Beispiel aus Level 1:



Schleife:

In einer Schleife werden einer oder mehrere Befehle wiederholt ausgeführt.

Beispiel aus Level 5:



Bedingung:

Nach der Überprüfung einer Bedingung werden einer oder mehrere Befehle nur dann ausgeführt, wenn die Bedingung zutrifft.

Beispiel aus Level 11:



Algorithmische Grundbausteine

- ▶ Reihenfolge (Sequenz)
- ▶ Wiederholung (Schleife)
- ▶ Bedingungen (Entscheidungen)
- ▶ Variablen (Platzhalter)
- ▶ EVA-Prinzip
(Eingabe - Verarbeitung - Ausgabe)

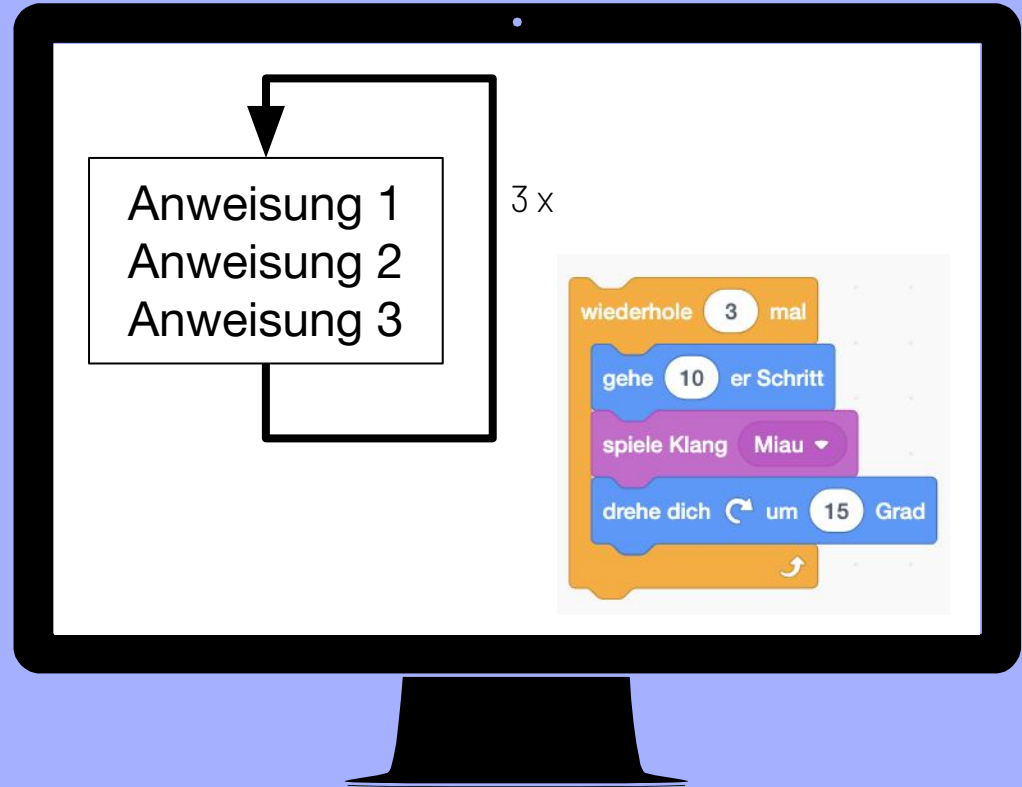


Reihenfolge Sequenz

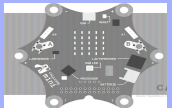
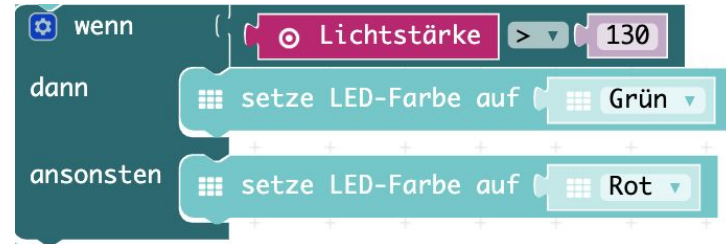
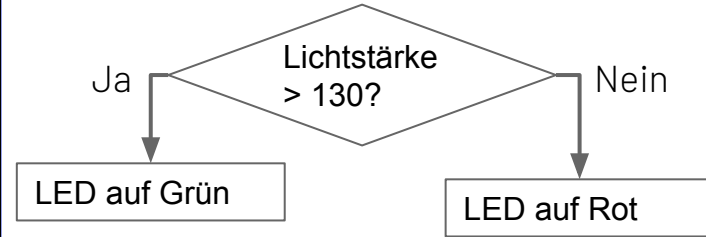
Anweisung 1
Anweisung 2
Anweisung 3



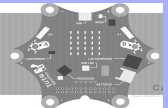
Wiederholung Schleife



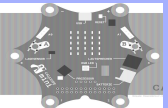
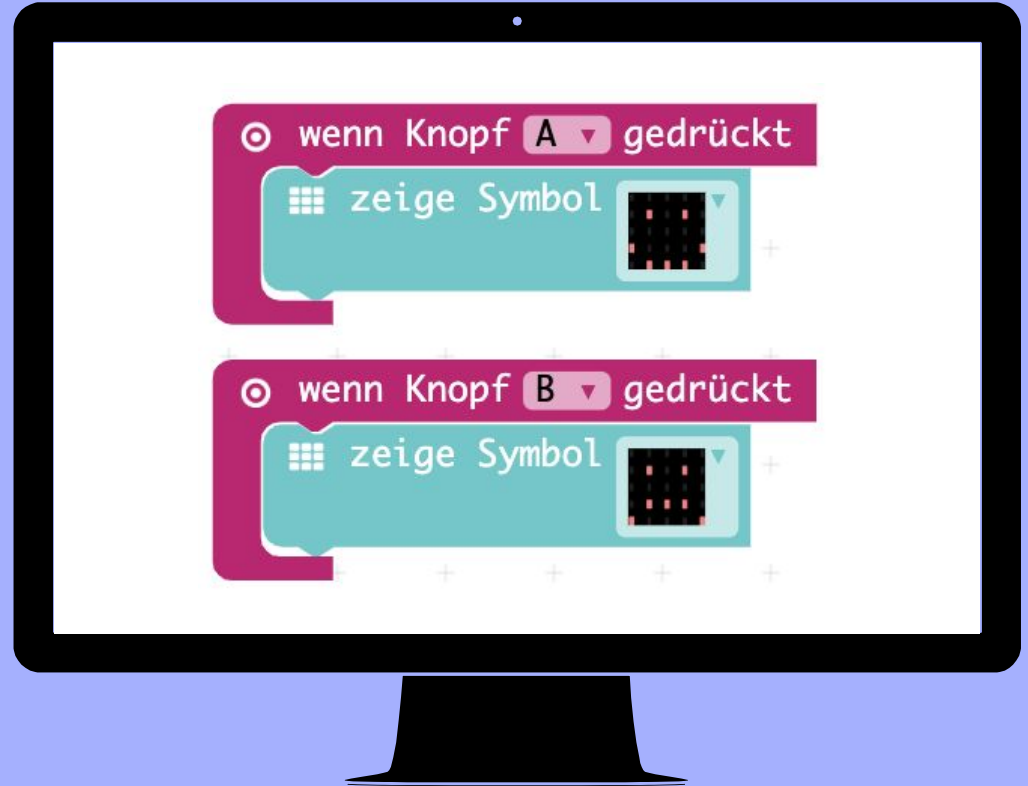
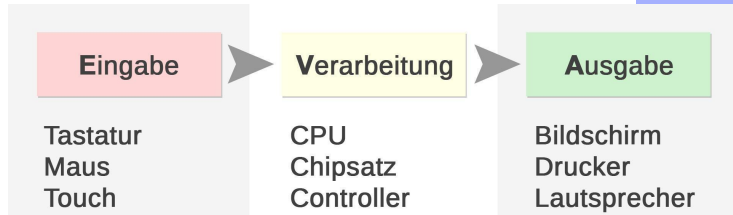
Bedingungen Entscheidungen



Variablen Platzhalter



EVA-Prinzip



Grafische Programmiersprache - ChatGPT

Was ist eine grafische Programmiersprache?

Eine **grafische Programmiersprache** (auch *visuelle Programmiersprache* genannt) ist eine Art von Programmiersprache, bei der Programme nicht (nur) durch Text geschrieben, sondern durch das **Zusammenstellen von grafischen Elementen** erstellt werden. Statt Codezeilen tippt man also z. B. Blöcke, Symbole oder Diagramme zusammen, die bestimmte Funktionen oder Abläufe darstellen.

Merkmale grafischer Programmiersprachen:

- **Drag-and-Drop-Oberfläche:** Man arbeitet mit einer Benutzeroberfläche, bei der man Blöcke oder Symbole per Maus zusammenfügt.
- **Klarere Visualisierung:** Logik und Abläufe sind oft leichter zu erkennen, weil sie visuell dargestellt sind.
- **Weniger Syntaxfehler:** Da man nicht selbst tippt, passieren weniger Tipp- oder Formatierungsfehler.
- **Ideal für Einsteiger:** Besonders im Bildungsbereich (z. B. für Kinder oder Anfänger) sind grafische Sprachen sehr beliebt.

Bekannte Beispiele:

- **Scratch:** Eine weit verbreitete grafische Sprache für Kinder. Man programmiert, indem man bunte Blöcke zusammensteckt.
- **Blockly:** Ein von Google entwickeltes Framework, auf dem auch viele Lernplattformen aufbauen.
- **LabVIEW:** Wird oft im technischen und wissenschaftlichen Bereich verwendet, z. B. zur Steuerung von Messgeräten.

Vergleich mit textbasierten Sprachen:

Grafische Sprache	Textbasierte Sprache
Einfacher Einstieg	Höhere Lernkurve
Visuelle Darstellung	Reiner Text
Weniger flexibel	Sehr flexibel und mächtig
Begrenzter Umfang	Für große Projekte besser

Wenn du magst, kann ich dir auch ein Beispiel zeigen – oder dir helfen, selbst etwas mit einer grafischen Sprache zu bauen! 😊

Grafische Programmiersprachen

Ich kann ...

- ❑ beschreiben, was eine grafische Programmiersprache auszeichnet.
- ❑ Vor- und Nachteile von grafischen Programmiersprachen aufzählen.
- ❑ Beispiele für grafische Programmiersprachen nennen.
- ❑ algorithmische Grundbausteine aufzählen und sie in einer grafischen Programmiersprache schreiben.
- ❑ einfache Algorithmen in einer grafischen Programmiersprache umsetzen.



Informatik an saarländischen Schulen

Lehrplan Informatik in Klasse 7



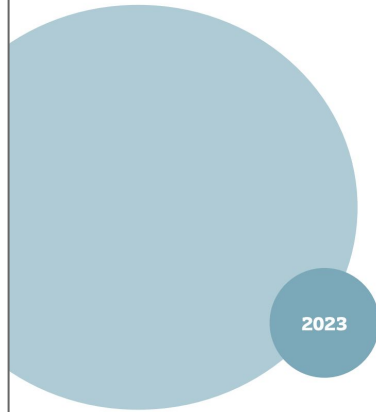
Themenfelder Klassenstufe 7	Informatik
Information und Daten	ca. 25 %
Grundlagen der Codierung	
Informatiksysteme	ca. 25 %
Computer und das Internet	
Algorithmen und Programmierung	ca. 35 %
Entwurf und blockbasierte Implementierung	
Künstliche Intelligenz	ca. 15 %
Spiele und Künstliche Intelligenz	

Kontrollstrukturen

- Anweisungssequenzen
 - ein- und zweiseitige Verzweigung
 - Schleifen
 - implementieren Algorithmen in einer blockbasierten Programmierumgebung,
 - verwenden Anweisungssequenzen,
 - verwenden ein- und zweiseitige Verzweigungen,
 - verwenden kopfgesteuerte Schleifen.
- Zur Vermeidung syntaktischer Schwierigkeiten ist zur Implementierung behandelter Algorithmen eine *blockbasierte* Programmierumgebung zu verwenden.

Lehrplan Informatik in Klasse 8

Informatik
Lehrplan
Gemeinschaftsschule
Klassenstufe 8
Redaktionell geänderte Fassung (April 2025)



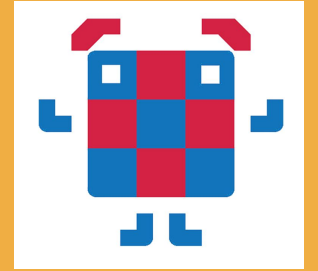
2023

Ministerium für
Bildung und Kultur
SAARLAND

Themenfelder Klassenstufe 8	Informatik
IT-Sicherheit	ca. 30 %
Datensicherheit und klassische Kryptologie	
Algorithmen und Programmierung	ca. 35 %
Modellieren und Implementieren	
Information und Daten	ca. 20 %
Daten darstellen und verarbeiten (Tabellenkalkulation)	
Künstliche Intelligenz	ca. 15 %
Einführung in das Maschinelle Lernen	

Vorschläge und Hinweise

- Hinsichtlich der Programmiersprache besteht die Möglichkeit, entweder die in Klassenstufe 7 eingeführte blockbasierte Programmiersprache weiterhin zu nutzen, oder zu einer textuellen Programmiersprache (wie etwa *Python*) zu wechseln. Auf eine detaillierte Beschäftigung mit Besonderheiten der Programmiersprache sollte verzichtet werden.



Tutorials: IT4Kids

Tutorials: IT4Kids

Die  neue Cubi-Version ist da

IT4KIDS

Grundschule ▾ Weiterführende Schule

Spielerisch programmieren lernen.

Chancengleichheit stärken.
Zukunftskompetenzen fördern.





















Fortbildungen **Lernsoftware** **Vor Ort**

it-for-kids.org

I KIDS

Calliope Computer in meinem Alltag Daten und Netze **IT4Kids - Cubi** Scratch

Level 0 Level 1 Level 2 Level 3

 Wenn Start gedrückt wurde	 Gehe 10 Schritte	 Drehe 90 Grad	 Gehe 10 Schritte Drehe 90 Grad	 Gehe 10 Schritte Drehe 90 Grad
1.1 Sequenzen - Einführung IT4Kids - Cubi	1.2 Sequenzen - Figuren bewegen IT4Kids - Cubi	1.3 Sequenzen - Figuren drehen IT4Kids - Cubi	1.4 Sequenzen - Code-Detektiv IT4Kids - Cubi	1.5 Sequenzen - Abschlussquiz IT4Kids - Cubi
 Wenn Start gedrückt wurde	 Wiederhole 1 mal dann	 Wiederhole fortlaufend	 Wiederhole 1 mal dann	 Wiederhole 1 mal dann
2.1 Schleifen - Einführung IT4Kids - Cubi	2.2 Schleifen - Anweisungen wiederholen IT4Kids - Cubi	2.3 Schleifen - Anweisungen endlos wiederholen IT4Kids - Cubi	2.4 Schleifen - Code-Detektiv IT4Kids - Cubi	2.5 Schleifen - Abschlussquiz IT4Kids - Cubi
 Wenn Start gedrückt wurde	 berühre ich 1 Taste	 und	 wenn dann	 wenn dann wenn dann
3.1 Verzweigungen - Einführung IT4Kids - Cubi	3.2 Bedingungen - Aussagen überprüfen IT4Kids - Cubi	3.3 Bedingungen - Aussagen verknüpfen IT4Kids - Cubi	3.4 Verzweigungen - Wenn-Dann-Logik IT4Kids - Cubi	3.5 Verzweigungen - Verschachtelte Verzweigungen IT4Kids - Cubi
 wenn dann	 wenn dann	 Wenn Start gedrückt wurde	 "Katze"	 Cubi chicken and rabbit
3.6 Verzweigungen - Code-Detektiv IT4Kids - Cubi	3.7 Verzweigungen - Abschlussquiz IT4Kids - Cubi	4.1 Variablen - Einführung IT4Kids - Cubi	4.2 Variablen - Datentypen IT4Kids - Cubi	4.3 Variablen - Variablenreise IT4Kids - Cubi

inf-schule.de/kids/kategorie:cubi



Bundesweite Informatikwettbewerbe

Bundesweite Informatikwettbewerbe

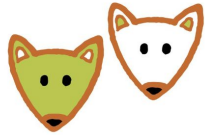


bwinf.de



Informatik Biber

- Klasse 3 - 13
- Ohne Programmieren
- Knobelaufgaben
- 1 Online-Test
- Jedes Jahr im November



Jugendwettbewerb Informatik

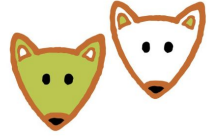
- Klasse 5 (3) - 13
- **Grafische Programmiersprache** (Blockly)
- 2 Online-Runden und 1 Test mit 2 freien Programmieraufgaben
 - Probewettbewerb jeweils vor der Runde
 - 1. Online-Test 23.02. - 08.03.2026
 - 2. Online-Test 16.03. - 03.05.2026
 - 3. Runde: ab 1. September



Bundeswettbewerb Informatik

- Nur freie Programmieraufgaben
- Abgabe der 1. Runde bis Ende November
- Insgesamt 3 Runden innerhalb eines Jahres

Jugendwettbewerb Informatik - 1. Runde



jwinf.de

Übungsaufgaben!

Jwinf > Wettbewerb > Jugendwettbewerb Informatik > Jugendwettbewerb Informatik 2019 Runde 1 (Jahrgangsstufen 3–6) > Die Aufgaben > Blumen einsammeln

Blumen einsammeln

AUFGABENSTELLUNG

Version☆☆☆ Version☆☆☆☆ Version☆☆☆☆☆

Programmiere den Roboter:
Der Roboter soll so gehen, dass er zur Blume gelangt.
Sobald er da ist, pflückt er sie von alleine.

Noch 2 von 10 Bausteinen verfügbar.

drehe nach links Roboter-Programm

drehe nach rechts

gehe vorwärts

|< |> > >> >>> >|

Ausführen und bewerten





Scratch

Wo ist der Scratch-Editor?

scratch.mit.edu



Erstelle Geschichten,
Spiele und
Animationen.
Teile sie mit anderen
auf der ganzen Welt.

Beginne mit dem Erstellen

Werde Scratcher

Schau dir das Video an

Über Scratch

Für Eltern

Für Lehrkräfte

Vorgestellte Projekte

Raindrop!

Office Maze

Fall of the farm

Universal Lab

-> Anmelden
(Fragen beantworten)
Sprache einstellen am Ende
der Seite:

Die beliebtesten Projekte

Vector Test

Real video Player v0.

ColorLess

3000

monoton

Über Scratch

Gemeinschaft

Rechtliche Hinweise

Scratch-Familie

Deutsch

Die Startseite einer Klasse



Scratch **Entwickeln** Entdecke Ideen Über Scratch Suche infolab-saar

2024-03-08 Arbeitslehre

Klassenseite | Erstellt vor 1 Stunde, 21 Minuten
United States

Über diese Klasse

Beschreibe diese Klasse. Dies wird öffentlich sichtbar, vermeide also bitte identifizierende Informationen, zum Beispiel Klarnamen.


Woran wir gerade arbeiten

Beschreibe was diese Klasse macht

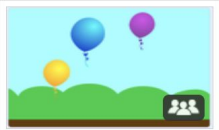
Letzte Aktivität

- Arbeitslehre46 wurde Kurator von [Labyrinth-Spiele](#)
vor 1 Stunde, 11 Minuten
- Arbeitslehre45 wurde Kurator von [Fangspiele](#)
vor 1 Stunde, 11 Minuten
- Arbeitslehre53 wurde Kurator von [Animationen](#)
vor 1 Stunde, 11 Minuten
- Arbeitslehre53 wurde Kurator von [Ballon-Spiele](#)
vor 1 Stunde, 11 Minuten

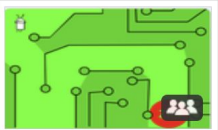
Klassenstudios (4) [Alle anzeigen](#)




Animationen



Ballon-Spiele



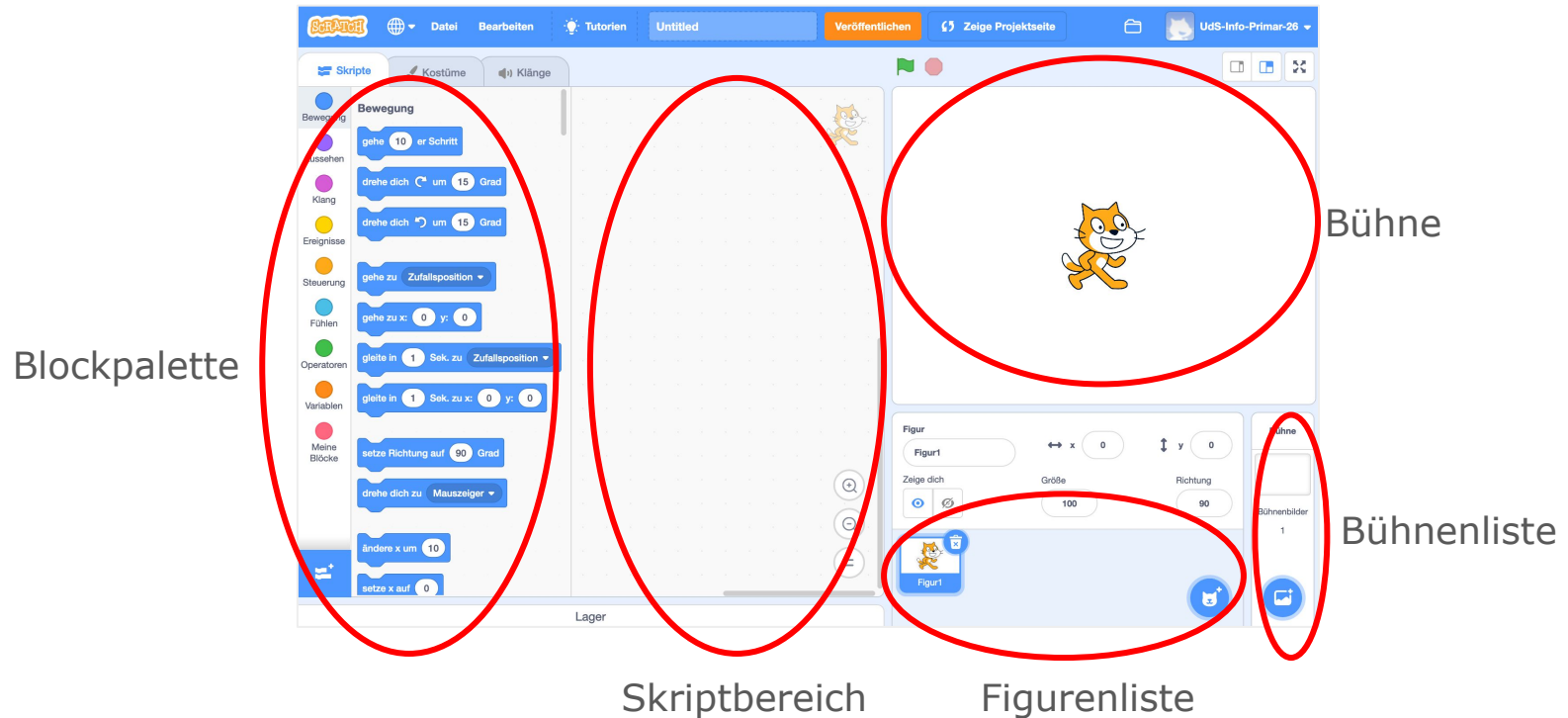
Labyrinth-Spiele



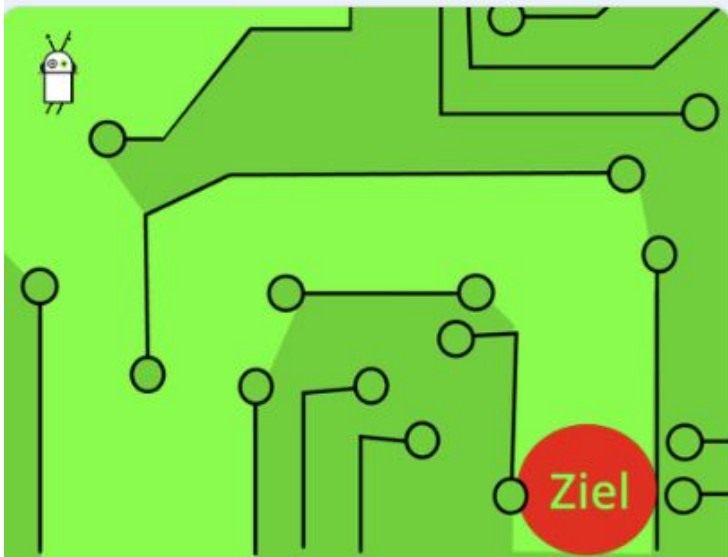
Fangspiele

-> Entwickeln

Der Scratch-Editor und seine Bestandteile

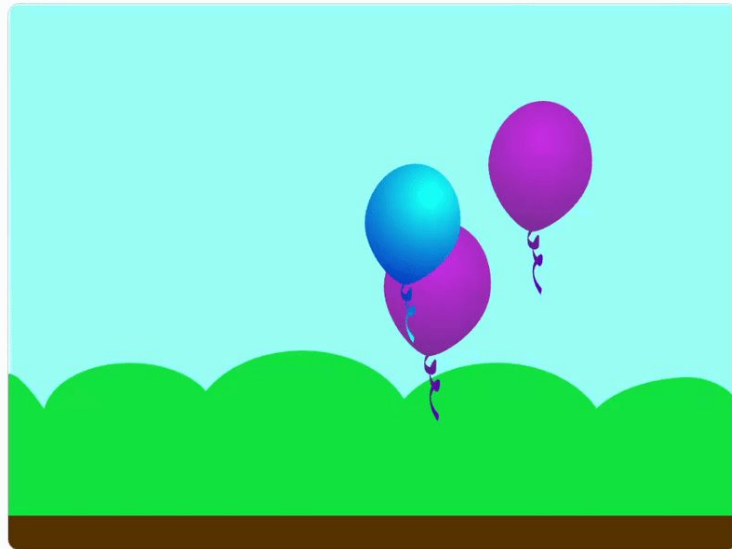


Labyrinth-Spiel



Braucht Tastatur

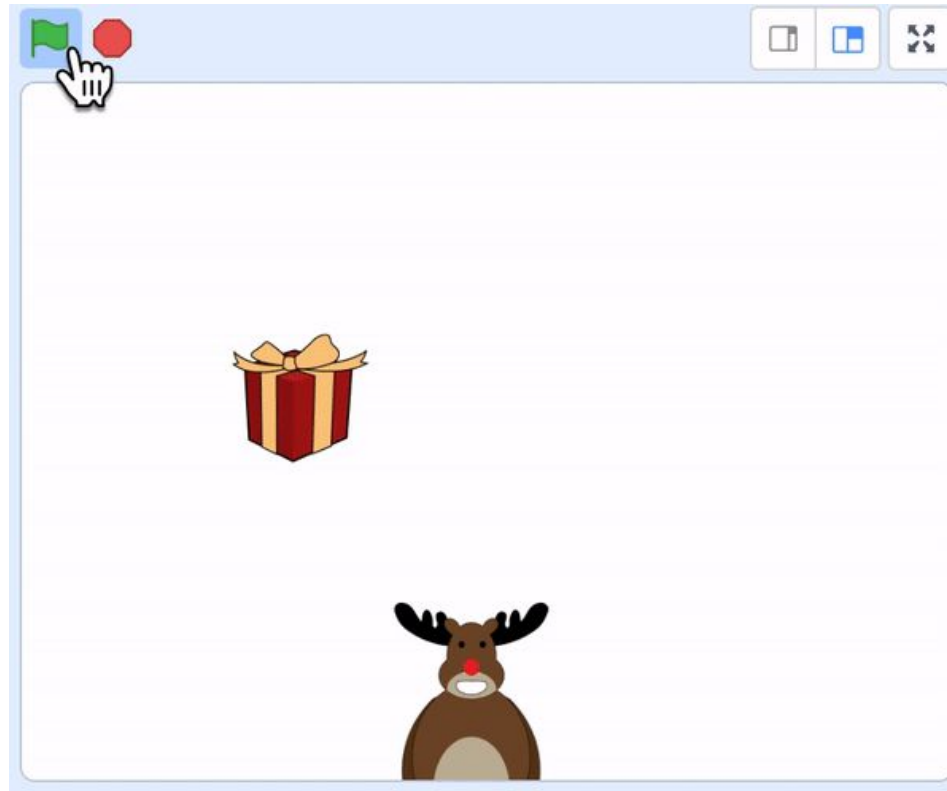
Ballons platzen lassen



Spielbar mit Touch-Screen

coderdojo-saar.de/2026/03/10/coderdojo-im-maerz-2026/

Fangspiel-Spiel - mit Anleitung



LEGO Spike Prime



Erste Schritte mit SPIKE™ Prime

6 Erste-Schritte-Übungen erleichtern den Einstieg in SPIKE Prime!

ERSTE SCHRITTE

Zuletzt verwendete Projekte

- Neues Projekt
- Projekt 5 vor einem Monat
- Projekt 3 vor einem Monat
- Breakdancer vor 3 Monaten
- Projekt 2 vor 3 Monaten
- Projekt 4 vor 4 Monaten

Lerneinheiten

Bauanleitungen

Startseite Erste Schritte Lerneinheiten Bauen Meine Projekte Hilfe

Motoren

- lasse Umdrehungen
- gehe auf kürzestem Wege
- starte Motor
- stoppe Motor
- stelle Geschwindigkeit auf 75
- Position
- Geschwindigkeit

Bewegung

- bewege dich 10 Umdrehungen
- starte Bewegung
- bewege dich nach rechts: 30 10
- starte Bewegung nach rechts: 30
- halte an

SPIKE™ Prime

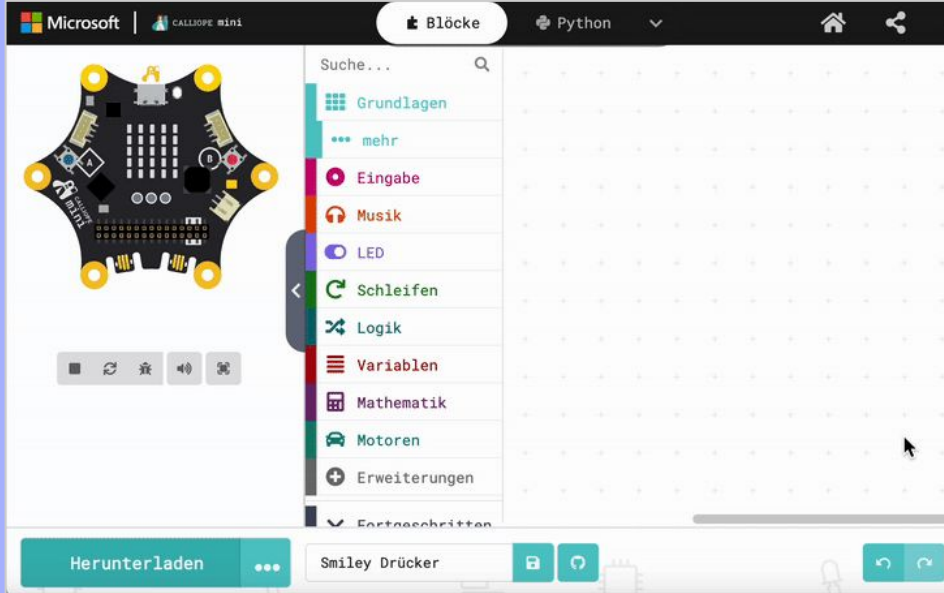
Projekt 6

Wenn Programm startet

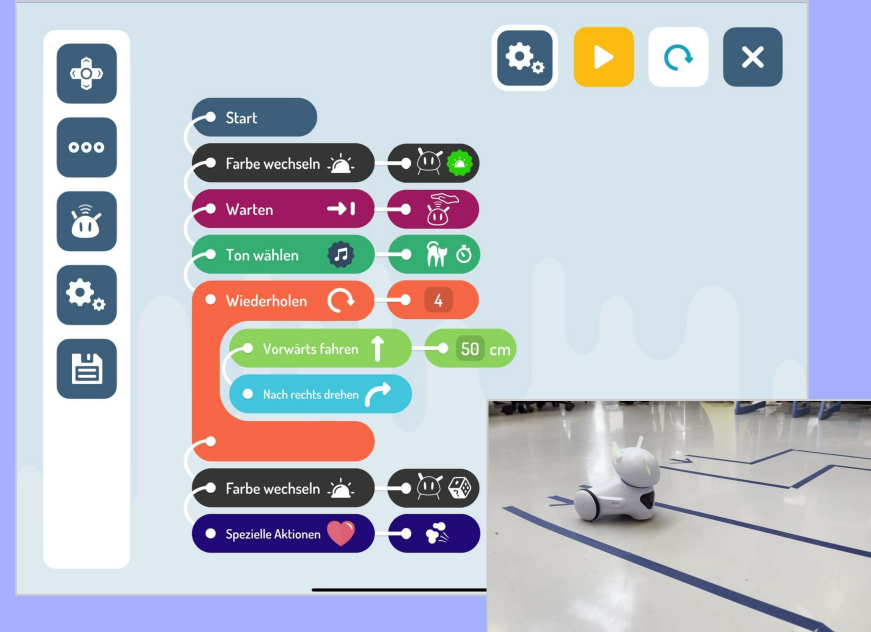
Projekt 6

Verbinden

Hardware programmieren

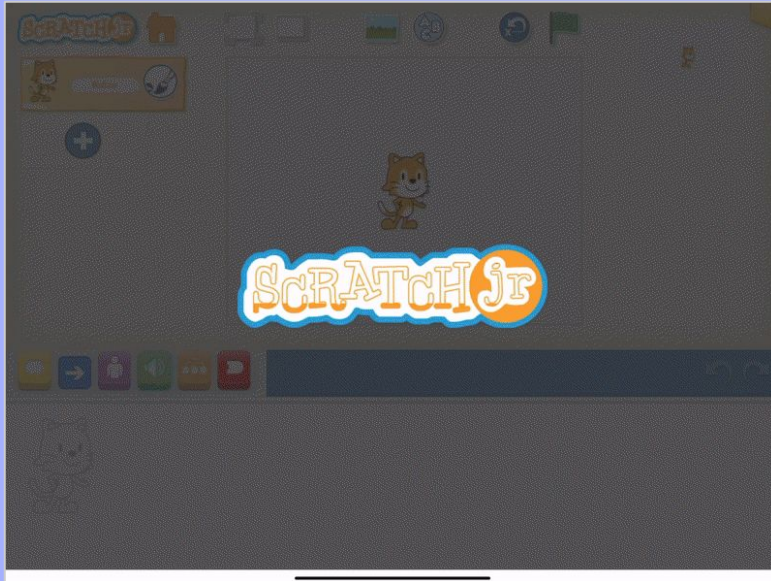


MakeCode für Calliope mini
makecode.calliope.cc



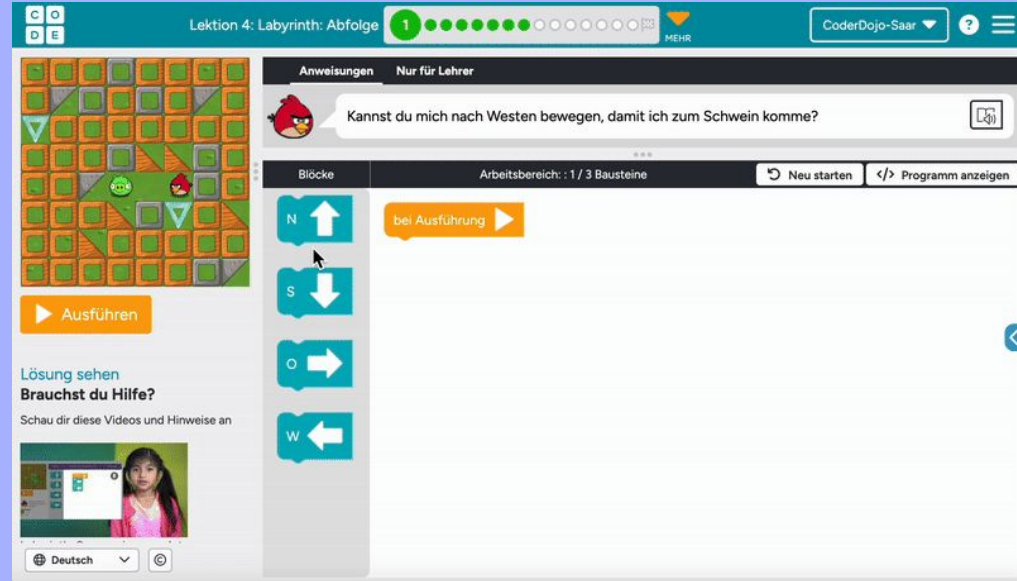
Photon Blocks für Photon Roboter
photon.education/de/apps/

Programmieren ohne Lesekompetenz



ScratchJr

www.scratchjr.org



Code.org

studio.code.org/s/course1/lessons/4/levels/1

Forschung zu grafischen Programmiersprachen

Session: Computing in Schools I

ITICSE '20, June 15–19, 2020, Trondheim, Norway

Common Bugs in Scratch Programs

Christoph Frädri^{ch} Florian Obermüller Nina Körber Ute Heuer Gordon Fraser
 fraedric@fim.uni- obermuel@fim.uni- koerber@fim.uni- ute.heuer@uni- gordon.fraser@uni-
 passau.de passau.de passau.de passau.de passau.de
 University of Passau University of Pa University of Pa University of Pa University of Pa
 Passau, Germany Passau, Germa

ABSTRACT

Bugs in SCRATCH programs can spoil the fun and success. Many common bugs are the result of bad code. In this paper we present a collection of patterns that typically hint at bugs in SCRATCH programs. We present a tool called LITTERBOX which can automatically detect and evaluate how frequently these patterns occur in their consequences usually are. While fixing bugs is part of learning, the possibility to identify the bugs provides the potential to support learners.

CCS CONCEPTS

• Social and professional topics → Software education; K-12 education; • Software and its Visual languages.

KEYWORDS

Scratch, Block-based programming, Code quality




Willkommen in XLogoOnline

1 **Mini**
Blockbasiert, ohne Zahlen

2 **Midi**
Blockbasiert, mit Parametern

3 **Maxi**
Programmieren in Logo

4 **Mega**
Programmieren in Python

Christoph Frädri^{ch}, Florian Obermüller, Nina Körber, Ute Heuer, and Gordon Fraser. 2020. **Common Bugs in Scratch Programs**. In Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (Trondheim, Norway) (ITICSE '20). 89–95. <https://doi.org/10.1145/3341525.3387389>

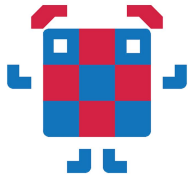
xlogo.inf.ethz.ch/release/latest/#/

Grafische Programmiersprachen

Ich kann ...

- ❑ beschreiben, was eine grafische Programmiersprache auszeichnet.
- ❑ Vor- und Nachteile von grafischen Programmiersprachen aufzählen.
- ❑ Beispiele für grafische Programmiersprachen nennen.
- ❑ algorithmische Grundbausteine aufzählen und sie in einer grafischen Programmiersprache schreiben.
- ❑ einfache Algorithmen in einer grafischen Programmiersprache umsetzen.

Grafische Programmiersprachen



- Hour of Code
- Informatik an saarländischen Schulen
- Algorithmische Grundbausteine
- Tutorials: Cubi/JWINF
- Scratch
 - Animation
 - Labyrinth/Ballons platzen lassen
 - Fangspiel
 - Lehrerkonto
- Lego Spike Prime, Calliope, Photon

