

Neues von Karol, Scratch & Co

Alternative didaktische Zugänge zur Algorithmik

Johannes Neumeyer, Korbinian-Aigner-Gymnasium Erding
Franz Jetzinger, Erasmus-Grasser-Gymnasium München

Neuerungen im Lehrplan Plus

1. Realschule
(Informationstechnologie)
2. Gymnasium
(Natur und Technik)

▼ IT 2.6 Programmierung – Algorithmen und Objekte

G1: Modellierung

Aufbauend auf d
schäftigen sich d
hierfür verwende
Diese Strukturen

- Abläufe verba
- Die Grundstru
- ter Probleme v
- Algorithmen m

▼ IT 2.6.1 Modellieren und Codieren von Algorithmen (ca. 14 Std.)

Die Schülerinnen und Schüler analysieren Abläufe, modellieren sie mit algorithmischen Grundelementen und setzen sie mit einem geeigneten Programmierwerkzeug um.

Kompetenzerwartungen

Die Schülerinnen und Schüler ...

- analysieren Abläufe (z. B. bedingte Bewegung eines Roboters) und gliedern diese in sinnvolle Teilschritte, um dazu eindeutige Handlungsvorschriften zu formulieren.
- verwenden algorithmische Grundstrukturen, um Abläufe zu modellieren und stellen sie mit geeigneten Notationsformen dar.
- setzen algorithmische Grundstrukturen und Variablen ein, um Programmabläufe zu codieren und in einer geeigneten Programmierungsumgebung zu implementieren.
- testen und optimieren ihre einfachen Programme.

Inhalte zu den Kompetenzen:

- Algorithmus: Begriff und Beispiele
- algorithmische Grundstrukturen: Anweisung, Sequenz, Auswahl und Wiederholung
- Notationsformen, z. B. Programmablaufplan, Struktogramm, Pseudocode, Aktivitätsdiagramm
- Variablenkonzept: Bezeichner, Datentypen, Wertzuweisung

(14 Std.)

en Systemen be-
nen, dass sich die
schreiben lassen.

erung geeigne-

Aktueller Lehrplan des G8

NT 7.2 Schwerpunkt Informatik (ca. 28 Std.)

NT 7.2.1 Vernetzte Informationsstrukturen - Internet (ca. 13 Std.)

NT 7.2.2 Austausch von Information – E-Mail (ca. 2 Std.)

NT 7.2.3 Beschreibung von Abläufen durch Algorithmen (ca. 13 Std.)

LehrplanPLUS des G9

2 Schwerpunkt Informatik (ca. 28 Std.)

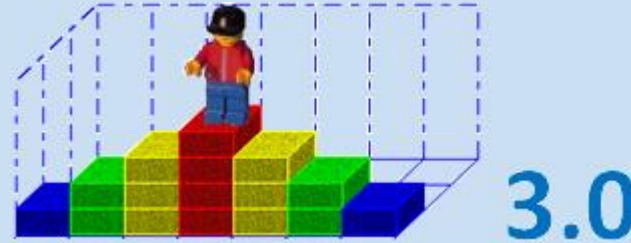
2.1 Vernetzte Informationsstrukturen (ca. 8 Std.)

2.2 Chancen und Risiken digitaler Kommunikation (ca. 5 Std.)

2.3 Beschreibung von Abläufen durch Algorithmen (ca. 11 Std.)

2.4 Projekt (ca. 4 Std.)

Übersicht vorgestellter Werkzeuge und Aufgaben

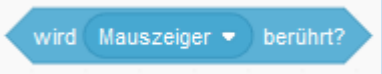


Blocksprachen





– Was ist neu?

- Seit 2. Januar 2019
- Basiert auf einer von Googles Blockly abgeleiteten Bibliothek namens „Scratch Blocks“
- GUI: komplett im „material design“ & neu gegliedert (z. B. Bühne rechts)
- Blöcke: neue Reihenfolge, neue Blöcke/Optionen, z. B. 
- Layout: (optional) waagrecht angeordnete Blocks
→ kompakter → unterstützt mobile Geräte besser
- Extensions: neue Extensions/bisherige Standard-Blocks als Extensions:
(links unten im Reiter „Code“)
 - **„digital“:**
 - text-to-speech, Translate (nutzt Google Translate), Pen, Music, ...
 - **„physical“:**
 - EV3 (Schnittstelle zu mit Lego Mindstorms EV3), ...
- Erstellung eigener Extensions via JavaScript

Quelle:

- <https://scratch.mit.edu/projects/editor/>
- https://scratch-dach.info/wiki/Scratch_3.0
- [https://en.wikipedia.org/wiki/Scratch_\(programming_language\)#Scratch_3](https://en.wikipedia.org/wiki/Scratch_(programming_language)#Scratch_3)



Blocksprachen vs.



Speichern:

- Download als *.sb3 Datei
- Speichern in der Cloud
- Offline-Editor „Scratch Desktop“

Zentral:

- Veröffentlichen in der Cloud
- Projekte Remixen
- Community, Forum
- Möglichkeit zur Einbindung von JavaScript

Hardware:

- Läuft auf Tablets seit Scratch 3.0, auf Smartphones Projekte nicht editierbar
- **Schnittstellen zur „physical world“:**
z. B. Lego Mindstorms EV3 (seit Scratch 3.0)

Speichern:

- Speichern im Browser
- Speichern in der Cloud
- Export als XML Datei

Zentral:

- Linkfreigabe via „share“
- weniger „Spielereien“; übersichtlicher
- Möglichkeit zur Einbindung von JavaScript
- „Umfangreicher“ ...

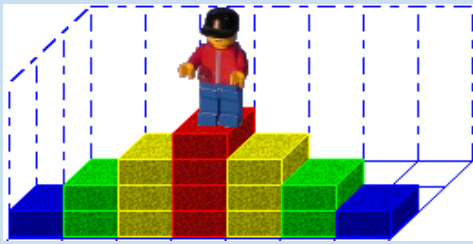
Hardware:

- nach Speichern in der Cloud auch direkt auf Smartphone ausführbar (unkomfortabel?)
- **Schnittstellen zur „physical world“:**
z. B. Arduino microcomputer (Snap4Arduino), Lego Mindstorms NXT

Blockly

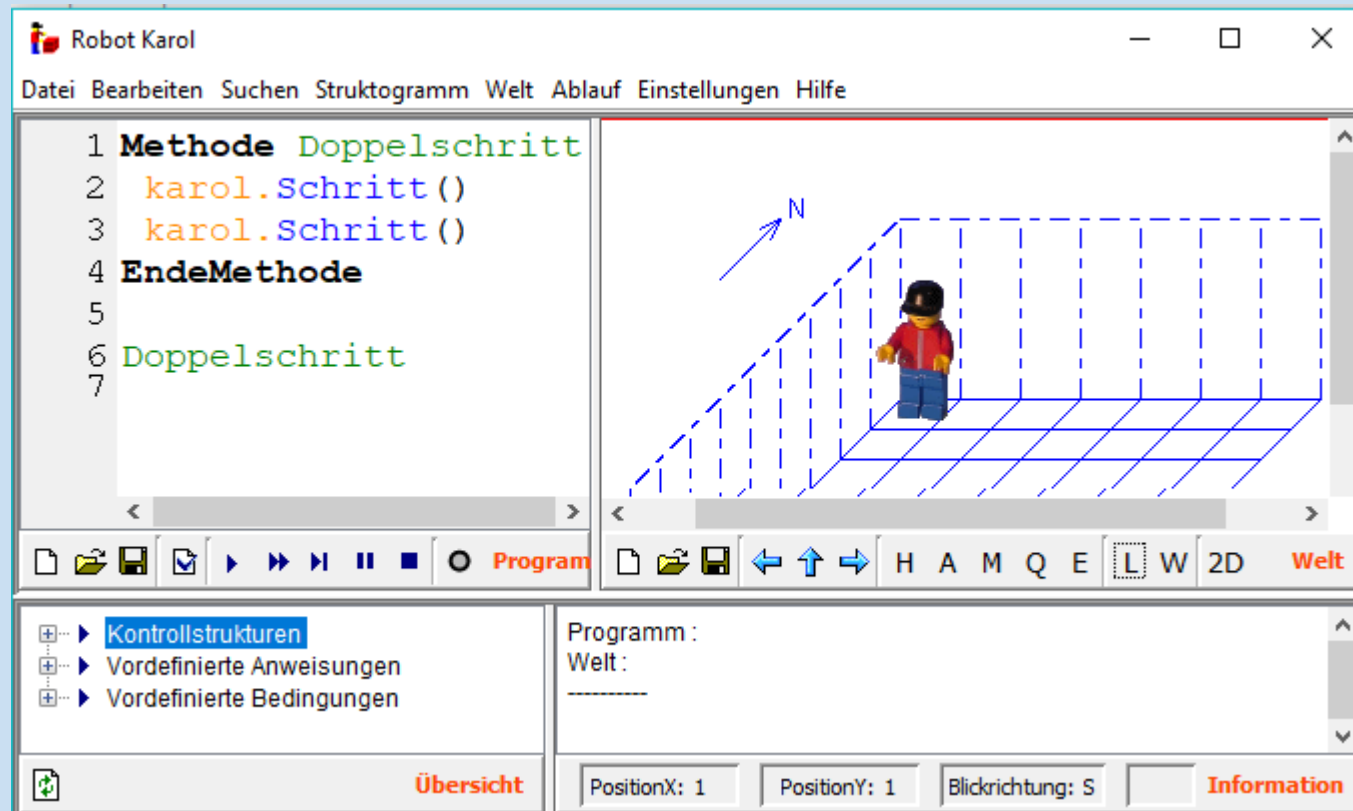
Offene Bibliothek zur Integration eines blockbasierten Editors in verschiedenen Umgebungen, z. B.:

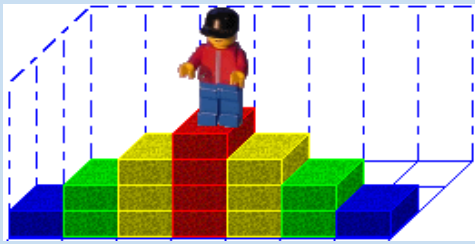
- „inspiriert von Blockly“: <https://lab.open-roberta.org/>
(Frauenhofer Institut Intelligente Analyse- und Informationssysteme) → Calliope
- www.Code.org → Hour of Code
- **BWInf: Jugendwettbewerb Informatik**
wettbewerb.jwinf.de/ → Trainieren → Übung: Roboter → Labyrinth
- [MIT App Inventor](https://www.mitsubishicommunications.com/education/)



3.0 – Bewährtes

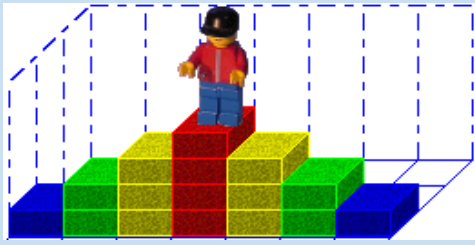
- alte Projekte weiterhin nutzbar
- Aufbau bewusst analog zu Karol 1.0





3.0 – Neuerungen

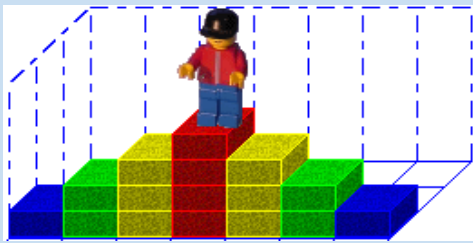
- **Plattformunabhängigkeit** auf der Basis von Java 1.8
- „Hinlegen()“ und „MarkeSetzen()“ alternativ mit **Parameter für Farbe**
- Auch bei Bedingungen Parameterwert möglich (z. B. „IstZiegel(rot)“)
- „EndeWiederhole“ statt „*Wiederhole“ möglich
- „Methode“ statt „Anweisung“ möglich
- Struktogramm in die Zwischenablage kopieren



3.0

– Warum nicht
auch blockbasiert?

- Umgang mit Tastatur und Maus als nicht mehr selbstverständliche Basiskompetenz
- Vorbereitung auf 10. Jgst ff
- Punktnotation

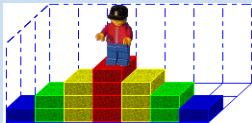


vs. Blocksprachen

- Umgang mit Tastatur und Maus als nicht mehr selbstverständliche Basiskompetenz
 - Quelltext muss eigenständig strukturiert werden; Sorgfalt notwendig
→ Vorbereitung auf Jgst. 10 ff.
 - Editor mit Syntaxhervorhebung
 - Struktogramme sind integriert – Abläufe können auf's Wesentliche reduziert werden
 - zielgerichteteres Arbeiten: Endzustand oft klar definiert
- Programmierung nach Baukastenprinzip
→ keine Tippfehler/Syntaxfehler
 - Motivation u. U. höher: schnell schöne kreative Programme
 - eher offene Projekte

Alles ausprobieren

- Suchen Sie sich eine Aufgabe zu Karol und Scratch und setzen Sie diese um!
- Versuchen Sie zusätzlich eine Scratch-Aufgabe mit Snap umzusetzen
- Stöbern Sie in den Beispielen und finden Sie je zwei Ideen, die Ihnen besonders gut gefallen!



3.0

Download:

<http://informatikschulbuch.de/robotkarol>

<https://www.mebis.bayern.de/infoportal/empfehlung/robot-karol/> („Karol 3.0“ suchen)



Online Variante:

<https://scratch.mit.edu/>

=> Entwickeln

Offline Variante:

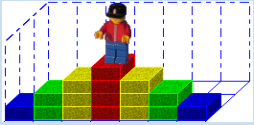
<https://scratch.mit.edu/download>



<https://snap.berkeley.edu/site/>

=> Run Snap!

Vergleich

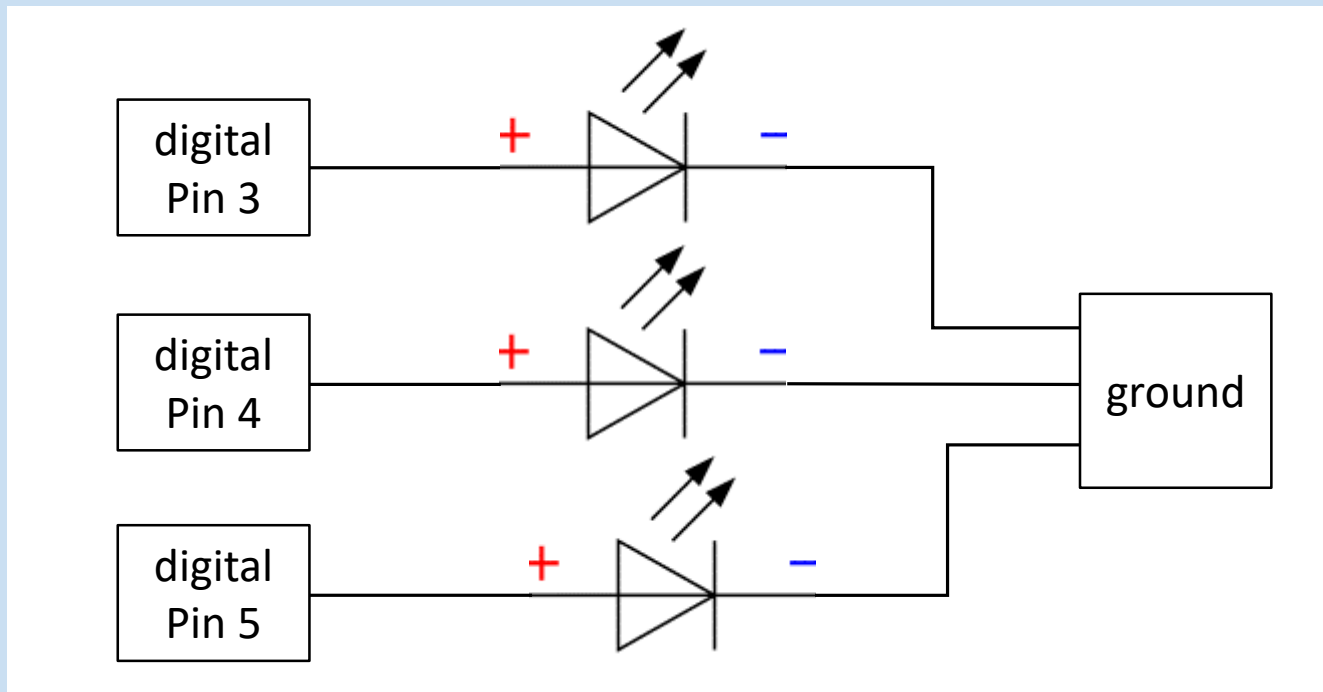
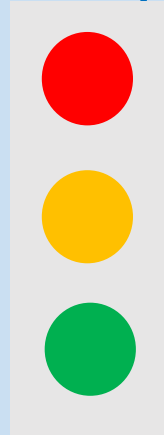


3.0



- Unterschiede und Gemeinsamkeiten
- Argumente für oder gegen den Einsatz im Unterricht
- mögliche Einsatzfelder

Lauflicht



Bildquellen

- Lupe:
https://commons.wikimedia.org/wiki/File:Magnifying_glass_01.svg
- Auto:
[https://commons.wikimedia.org/wiki/File:2007-06-16_Veritas_RS_\(05b\),_Bj._1948,_Start-Nr._97_\(ausgangs_Br%C3%BCnnchen\)_ret.jpg](https://commons.wikimedia.org/wiki/File:2007-06-16_Veritas_RS_(05b),_Bj._1948,_Start-Nr._97_(ausgangs_Br%C3%BCnnchen)_ret.jpg)
- Stuhl:
[https://commons.wikimedia.org/wiki/File:Sit_down_and_watch_the_world_pass_by..._\(30685693075\).jpg](https://commons.wikimedia.org/wiki/File:Sit_down_and_watch_the_world_pass_by..._(30685693075).jpg)

Mit Beiträgen von

- Peter Brichzin
- Klaus Reinold