

# Coding in der Volksschule

## *Alle Kinder profitieren – Mädchen in besonderem Ausmaß*

Sabine Sperk<sup>1</sup>

DOI: <https://doi.org/10.53349/resource.2023.i2.a1175>

### **Zusammenfassung**

Als Teil der MINT-Disziplinen *Mathematik, Informatik, Naturwissenschaften* und *Technik* nimmt das Erstellen von Computerprogrammen einen immer wichtiger werdenden Platz in der digitalen Bildung ein. Die Förderung des informatischen Denkens bereits im Volksschulalter nutzt allen Kindern, Mädchen jedoch in noch stärkerem Ausmaß. Durch eigene Erfolgserlebnisse sowie dem Erleben eines Abbaus von Stereotypen profitieren sie von Programmierprojekten in Form eines gestärkten informatischen Selbstbildes besonders. Dies erhöht ihre Chancen, an zukunftssträchtigen und lukrativen Berufsbildern mit informatischem Bezug teilzuhaben. Programmierprojekte an Volksschulen lassen sich flexibel umsetzen – in einem Ausmaß von wenigen Stunden bis hin zu größeren Klassen- oder Semesterprojekten. Ausschlaggebend für den Erfolg sind formelle Kriterien, wie etwa die regelmäßige Verfügbarkeit von Schul-PCs, jedoch auch die pädagogische Auswahl von Programmierumgebung und Begleitmaterial. Besonders geeignet für den Einstieg in die Programmierung mit Kindern sind die Programmierumgebung „Scratch“ sowie die Lernplattform „code.org“. Beide Umgebungen sind kostenlos, in mehreren Sprachen verfügbar und senken mit einem visuellen, blockbasierten Programmieransatz die Einstiegsbarrieren – sowohl für teilnehmende Kinder als auch für Lehrpersonen.

Coding, Programmieren, Volksschule, Grundschule, Selbstwirksamkeit

## **1 Warum Coding in der Volksschule?**

Um Kinder auf ein Leben in einer digitalisierten Gesellschaft vorzubereiten, ist es wichtig, ihnen Grundfähigkeiten in *Computational Thinking* (Informatisches Denken) zu vermitteln: „Computational thinking is a fundamental skill for everyone, not just for computer scientists.“

---

<sup>1</sup> Volksschule Wiener Neudorf, Europaplatz 6, 2351 Wiener Neudorf  
Email: [drachenklasse@outlook.com](mailto:drachenklasse@outlook.com)

To reading, writing and arithmetic, we should add computational thinking to every child's analytical ability." (Wing, 2006, S. 33)

Dies hat auch die Politik erkannt und entsprechende Inhalte in den Lehrplan der Volksschule (Bundesministerium für Bildung, Wissenschaft und Forschung, 2023) eingearbeitet, sodass Lehrer\*innen Coding-Projekte auch rechtlich abgedeckt im Unterricht durchführen können. Ein essenzieller Teil des informatischen Denkens ist dabei ein Verständnis für das *Verstehen, Erstellen, Formalisieren und Automatisieren von Schritt-für-Schritt-Anweisungen* (Antonitsch & Hanisch, 2014, S. 4). Dies sind grundlegende Kompetenzen, um Computerprogramme erstellen zu können. Angesichts der breitflächigen Digitalisierung der Gesellschaft kann sogar davon ausgegangen werden, dass sich Programmieren in einer gewissen Hinsicht als neue, zusätzliche Kulturtechnik etablieren wird:

Die Beherrschung elementarer informatischer Methoden und Werkzeuge ist damit auf dem besten Weg, neben Schreiben, Lesen und Rechnen zur vierten Kulturtechnik zu werden. Damit wird informatische Bildung zu einer gesellschaftlichen Aufgabe und sollte zukünftig ein fester Bestandteil einer grundlegenden Allgemeinbildung sein." (Haus der kleinen Forscher, 2017, S. 10)

Als ersten Schritt vermitteln Digitalisierungsinitiativen an Schulen Grundkenntnisse in der *Handhabung* von Computern. Dies ist essenziell, doch gerade an Volksschulen in Österreich bleibt es jedoch oft dabei. Um zu verstehen, wie Informatik funktioniert, ist es aber wichtig, bereits früh zusätzliche Kompetenzen zu bilden: „Wenn Kinder neben der bloßen Verwendung von Computern auch wissen sollen, wie Computer arbeiten und funktionieren, dann ist eine Auseinandersetzung mit Grundbegriffen der Programmierung nötig. Dadurch erhalten die Kinder nicht nur Einblicke auf, sondern auch hinter den Bildschirm.“ (Walter, 2018, S. 9)

## 2 Exkurs: Mädchen profitieren in besonderem Ausmaß

Die Sinnhaftigkeit einer Auseinandersetzung mit dem Thema bereits in der Volksschule, soll damit einmal als geklärt angesehen werden. Informatische Bildung muss jedoch auch unter einem geschlechtsspezifischen Aspekt betrachtet werden, denn Mädchen haben hier oft einen ganz anderen Zugang zum Thema als Jungen.

In den letzten Jahrzehnten hat sich die Bildungsteilhabe von Mädchen und Frauen positiv entwickelt. Dennoch sind sie in den besonders zukunftssträchtigen und lukrativen MINT-Disziplinen (Mathematik, Informatik, Naturwissenschaft, Technik) nach wie vor deutlich unterrepräsentiert. In der Vergangenheit wurden dafür mangelnde „Begabungen“ der Mädchen als Grund für das Fernbleiben aus entsprechenden Berufsbildern ausgemacht. Dies wurde mittlerweile durch umfassende Studien widerlegt und es ist erwiesen, dass Mädchen und

Jungen über gleiche Anlagen für MINT-Disziplinen verfügen (Hattie, 2009; Pisa 2015; TIMMS, 2019).

Warum lassen sich dann in vielen Ländern immer noch Leistungsunterschiede zwischen Jungen und Mädchen in Naturwissenschaften beobachten? Diese Ungleichheiten lassen sich mit dem Vorhandensein von Stereotypen und sozial konstruierten Attributionen erklären:

In den großen Vergleichsuntersuchungen [...] hat sich gezeigt, dass es sich beim Auftreten von Geschlechterunterschieden in den Mathematik- und Naturwissenschaftsleistungen vor allem um ein kulturelles Problem handelt. In einigen Ländern verschwinden die Unterschiede nahezu, während sie in anderen Ländern – wie z.B. Deutschland – noch ausgeprägt sind. (Jahnke-Klein, 2013, S. 48)

Die US-amerikanische Psychologin Barbara Mackoff (1998, S. 1) fasst dies in einem pädagogischen Kontext prägnant zusammen: “The biggest difference between girls and boys is how we treat them.” Eine mädchenstärkende MINT-Pädagogik muss also bedeuten, das *Selbstbild* der Mädchen positiv zu verändern (Jahnke-Klein, 2013).

Wo kann die Pädagogik nun hier ansetzen? Der kanadische Psychologe Albert Bandura (1997) hat in seinen Forschungen der kognitiven Psychologie untersucht, welchen Einfluss menschliche Selbstüberzeugungen auf das eigene Handeln haben. Vereinfacht gesagt, lassen sich diese so zusammenfassen: Wie stark ein Mensch überzeugt ist, in einem bestimmten Handlungsfeld Erfolg zu haben, hat einen positiven Effekt auf das tatsächliche Ergebnis. Diese innere Haltung wird mit dem Begriff der *Selbstwirksamkeitserwartung* definiert.

Umgelegt auf MINT-Disziplinen bedeutet dies: Wenn wir die Selbstüberzeugungen von Mädchen in diesen Bereichen stärken, dann werden sie darin auch bessere Leistungen zeigen und in weiterer Konsequenz auch MINT-bezogene Berufe ergreifen.

Pädagogische Impulse können Selbstüberzeugungen positiv beeinflussen – Bandura (1997) beschreibt hier vier Ansatzpunkte zur Veränderung:

1. häufige Erfolgserlebnisse durch selbst gemachte Erfahrungen
2. Lernen durch Vorbilder
3. Stärkung durch Zuspruch
4. Erleben eines positiven Lernklimas

Die Autorin dieses Beitrags hat anhand einer Einzelfallstudie (Volksschulklasse, Grundstufe II, sechs Monate Projektdauer) umfassend untersucht, wie sich diese Ansätze in einem Programmierprojekt in der Volksschule mädchenstärkend umsetzen lassen und welche Auswirkungen dies auf die Selbstüberzeugungen der an der Studie teilnehmenden Mädchen hatte.

Die Ergebnisse dieser Studie lassen sich in Kurzform wie folgt zusammenfassen:

- Die Mädchen konnten in Form einer höheren informatischen Selbstwirksamkeitserwartung (höhere Kompetenzerwartung) profitieren. Sie haben deshalb in Zukunft eine höhere Chance für einen persönlichen Zugang zu informatischen Disziplinen.
- Die Mädchen konnten einen Abbau von geschlechtsspezifischen Attributionen und Stereotypen erleben. Sie haben deshalb potenziell mehr Chancen und Motivation bei Schul-, Studien- und Berufswahl.
- Programmierprojekte in der Volksschule fördern nicht nur informatische Kompetenzen, sondern stärken insbesondere das Selbstbild von Mädchen. (Sperk, 2022)

Abschließend kann deshalb zusammengefasst werden, dass Programmierprojekte an der Volksschule nicht nur allgemein informatische Kompetenzen der Kinder vertiefen, sondern auch insbesondere einen positiven Beitrag zur Stärkung des Selbstbilds von Mädchen liefern und zum Abbau von geschlechterbezogenen Stereotypen führen.

### 3 Programmieren lehren und lernen – aber wie?

Unter Programmieren wird weitläufig das Erstellen von Computerprogrammen verstanden. Der Vollständigkeit halber soll dies hier aber nochmals genauer definiert werden: „Programmieren ist das Formulieren einer Problemlösung in Befehlsanweisungen mithilfe einer Sprache, die automatisch in Befehle übersetzt wird, die ein Computer ausführen kann“ (Wurm, 2013).

Computerprogramme stellen also Lösungen für Problemstellungen dar. Der Weg, eine solche Problemstellung zu lösen, lässt sich angelehnt an Ratz et al. (2018, S. 29) schematisch vereinfacht wie folgt darstellen:

1. Problem – Was soll gelöst werden?
2. Algorithmische Beschreibung – Wie kann das Problem gelöst werden?
3. Programm – Wie kann die Lösung informatisch (durch Programmierung) umgesetzt werden?
4. Problemlösung

Bevor damit begonnen werden kann, ein Programm zu schreiben, muss eine algorithmische Beschreibung erfolgen. Damit wird die Logik des Programms bestimmt, welche aus aufeinanderfolgenden Einzelschritten besteht.

Dies soll anhand eines Beispiels verdeutlicht werden – in der Abbildung sind die Schritte für den Ablauf des Händewaschens verdeutlicht:



Abbildung 1: Sequenz zum Händewaschen (eigene Darstellung)

Dieser alltägliche Vorgang stellt einen Algorithmus dar. „Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten.“ (Rogers, 1987).

Natürlich ist hier auch ein starker Bezug zur Mathematik gegeben, welche ebenfalls grundlegend auf Handlungsvorschriften (Algorithmen) basiert.

In Bezug auf das Lehren von Programmierung ist es wichtig, Kindern zu verdeutlichen, dass unsere tägliche Lebenswelt von Algorithmen durchdrungen ist. Es wird deshalb empfohlen,

als allererster Schritt in einem Programmierprojekt solche meist unbewussten Alltagsabläufe zu identifizieren und in Schritte (Sequenzierung) zu zerlegen. Anschließend kann damit begonnen werden, Problemlösungen auf logischer Ebene zu beschreiben. Für all diese Übungen ist noch kein Computer nötig („Programmieren ohne Strom“).

Sobald die Grundlagen der Sequenzierung erarbeitet wurden, können erste Übungen am Computer erfolgen. Wie dies gelingen kann, wird im nachfolgenden Kapitel beschrieben.

## 4 Empfehlungen für die Praxis an der Volksschule

Gerade an der Volksschule fällt es vielen Lehrer\*innen nicht leicht, sich informatischen Themen zu nähern. Deshalb wird der Blick nun nachfolgend auf praxistaugliche Ansätze gelegt.

### 4.1 Projektansätze und benötigte Ressourcen

Zunächst stellen sich dabei grundlegende Fragen in Bezug auf die Umsetzung des Themas in der Volksschule: Für welche Schulstufen eignet sich das Thema? In welchem Ausmaß kann Programmierung Teil des Unterrichts sein? Obwohl es Programmierkonzepte auch für jüngere Kinder gibt, wird sicheres Lesen als Voraussetzung empfohlen. Ein Programmierprojekt ist deshalb vor allem ab der Grundstufe II sinnvoll – also ab der 3. Klasse Volksschule.

Es sind dabei unterschiedliche Ansätze möglich, die sich sowohl in Hinblick auf verwendete Methoden als auch in zeitlicher Hinsicht deutlich unterscheiden. Die Autorin hat aus der eigenen Erfahrung heraus drei mögliche Projektansätze entwickelt, die nachfolgend skizziert werden:



Abbildung 2: Projektansätze Grundstufe II (eigene Darstellung)

Es sind also unterschiedliche Projektformen und -längen möglich, von einer Dauer von 2–3 Wochen bis hin zum Semester- oder Jahresprojekt.

In der Praxis stellt sich dann sehr schnell die Frage, welche Ressourcen für ein solches Projekt benötigt werden. Die wichtigsten Punkte werden hier angeführt:

- Essenziell für ein Projekt ist ein regelmäßiger Zugang zu PCs  
→ idealerweise in einem schuleigenen EDV-Raum
- Offline-Material für das Erlernen der Grundlagen  
→ „Programmieren ohne Strom“
- Softwareumgebung für PCs  
→ vorzugsweise freie (kostenlose) Angebote
- Begleitmaterial  
→ je nach Projektansatz kann der Einsatz eines Begleitbuches sinnvoll sein

Die Erfahrungen der Autorin mit einem mehrmonatigen Projekt haben gezeigt, dass die größte Herausforderung in der Infrastruktur liegt, insbesondere was die Verfügbarkeit von Schul-PCs betrifft. Bei guter Planung lässt sich ein Projekt dennoch in der Praxis gut umsetzen und setzt bei Nutzung von kostenlosen Angeboten (diese sind umfassend verfügbar) auch keine nennenswerten Investitionen für die Schule oder die Eltern der Kinder voraus.

## 4.2 Empfohlene Tools und Material für den Unterricht

Programmierung ist ein äußerst vielfältiges Themengebiet – es gibt beispielsweise hunderte Programmiersprachen. Wenn es darum geht, Methoden und Tools für den Unterricht auszuwählen, kann dieses Angebot für interessierte Lehrkräfte überwältigend sein. Nachfolgend werden einige Empfehlungen für Pädagog\*innen gegeben, die sowohl auf den praktischen Erfahrungen der Autorin basieren als auch den aktuellen Stand des Diskurses in der informatischen Pädagogik berücksichtigen.

### 4.2.1. Programmieren ohne Strom – Sequencing

Wie bereits erwähnt, ist es sinnvoll, den Kindern zu Beginn eines Projekts/Curriculums ein grundlegendes Verständnis für den schrittweisen Aufbau eines Algorithmus zu vermitteln. Im Internet lassen sich dazu viele Beispiele finden. An dieser Stelle sei ein Angebot des Westdeutschen Rundfunks WDR (2022) herausgehoben, der unter dem Motto „Programmieren mit der Maus“ umfangreiches Material veröffentlicht hat. Über ein ausdrucksbares Brettspiel lernen Kinder grundlegende Programmierkonzepte zu Befehlen, Schleifen und Funktionen. Alle Materialien sind kostenlos nutzbar.



Abbildung 3: Programmieren mit der Maus (WDR, 2022)

#### 4.2.2. Algorithmisches Denken fördern mit den Bee-Bots

Eine weitere Möglichkeit, algorithmisches Denken zu stärken, stellen die Bee-Bots dar. Diese sind kleine Bodenroboter, die mit Funktionstasten am Gerät gesteuert werden. Die Programmierung erfolgt mit den Tasten *vorwärts*, *rückwärts*, *links drehen*, *rechts drehen*, *Pause* und *löschen*. Durch Drücken der Taste GO wird eine Programmsequenz gestartet und der Roboter fährt dann auf dem Boden das entsprechende „Programm“ ab. Da der Fahrweg des Bee-Bots vorab geplant werden muss, können sich Kinder damit spielerisch dem Programmieren nähern. Bee-Bots können bereits im Kindergarten eingesetzt werden, sind jedoch auch in der Volksschule sinnvoll verwendbar (Brandhofer, 2016).



Abbildung 4: Bee-Bot Roboter (eigene Darstellung)

Bee-Bots stellen – vor allem in Klassenstärke – eine größere Investition dar, die von einer einzelnen Lehrkraft nicht geleistet werden kann. Für Niederösterreich gilt hier eine Besonderheit: Ab dem Jahr 2018 wurden alle niederösterreichischen Volksschulen mit Bee-Bots ausgestattet, so dass Pädagog\*innen diese ohne zusätzliche Kosten in ihren Unterricht integrieren können.

#### 4.2.3. Scratch – eine edukative Programmierumgebung zum Erlernen der Grundlagen

Scratch ist eine visuelle Programmiersprache für Kinder und Jugendliche, entwickelt am Massachusetts Institute of Technology (Kiefer, 2019, S. 2). Konzipiert als bildungsorientierte Programmierumgebung ermöglicht es, insbesondere Kindern, einen einfachen Zugang zum Thema „Programmierung“. Dazu werden Blöcke als visuelle Platzhalter für Programmieran-

weisungen verwendet – Scratch wird deshalb auch als blockbasierte Programmiersprache bezeichnet. Durch dieses Konzept lassen sich Programme schaffen, ohne dass Programmanweisungen textuell in Form von Programmzeilen geschrieben werden müssen.

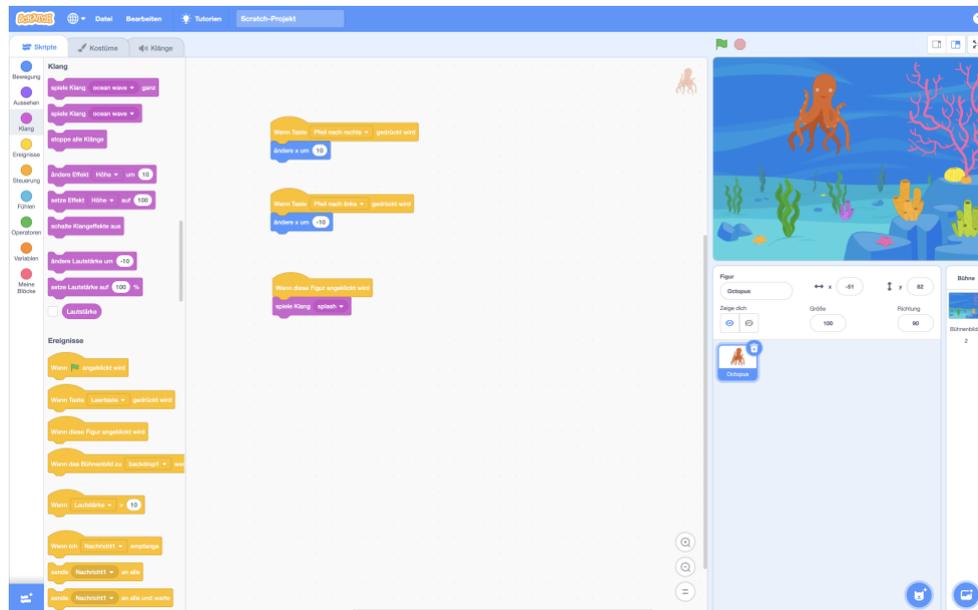


Abbildung 5: Scratch Programmierumgebung mit Befehlsblöcken (eigene Darstellung)

Scratch bietet für Lernende und Lehrende eine Reihe von Vorteilen:

- Die visuelle Umgebung unterstützt auf spielerischem Wege schnelle Lernerfolge.
- Scratch ist in vielen Sprachen nutzbar.
- Scratch ist komplett kostenlos verfügbar.
- Scratch benötigt sehr geringe Rechnerkapazitäten und lässt sich auf den meisten Computern einfach installieren.
- Im Internet gibt es umfassende Ressourcen zum Erlernen und Experimentieren.
- Es gibt auch für den deutschsprachigen Raum eine Reihe von Lehrwerken, anhand derer Scratch erlernt werden kann (Aufzählung angelehnt an Sperk, 2022).

Wie bereits erwähnt, gibt es für Scratch umfangreiches Material im Internet, aber auch in Buchform. Ein Werk, das sich im Rahmen der Studie der Autorin als sehr praxistauglich herausgestellt hat, ist *Programmieren mit der Maus* aus dem Rheinwerk Verlag (Kiefer, 2019). Durch einfache, aufeinander aufbauende Übungen werden Kinder darin spielerisch und kreativ in die Programmierung mit Scratch eingeführt.

#### 4.2.4. code.org – Lerneinheiten zum Programmieren mit Puzzle-Charakter

Während Scratch den Lernenden bei der Programm-Gestaltung weitgehende Freiheiten gewährt, verfolgt die Plattform code.org einen anderen Ansatz. Sie ist eine edukative Non-Profit-Organisation, die sich zum Ziel gesetzt hat, möglichst vielen Menschen Zugang zu Com-

puterwissenschaften zu ermöglichen. Obwohl die Organisation von äußerst potenten Sponsoren unterstützt wird (z.B. Google, Amazon, Microsoft) sind sämtliche Angebote kostenfrei im Internet via Browser verfügbar.

Auf der Internetseite code.org sind eine Vielzahl von Übungen zum Aufbau eines Grundverständnisses für die Programmierung verfügbar. Diese Übungen bauen in der Regel aufeinander auf und haben einen Puzzle-Charakter. Lernende lösen nacheinander kleine Aufgaben und arbeiten sich so in immer komplexere Themen ein. Es entstehen hier zwar keine Programme im eigentlichen Sinne, spielartige Elemente sorgen hier aber für anhaltende Motivation, die Aufgaben zu bewältigen.

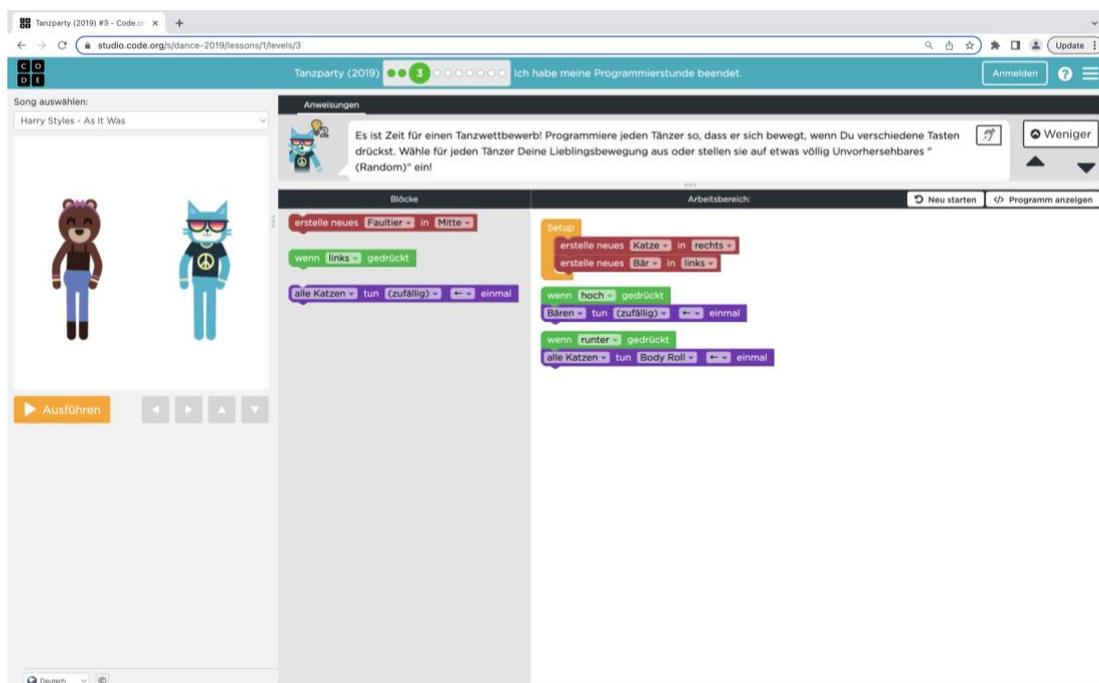


Abbildung 6: Beispielübung einer Aufgabe in code.org (eigene Darstellung).

Code.org ist ebenfalls in einer Vielzahl von Sprachen verfügbar und verfolgt ähnlich wie Scratch einen blockbasierten Programmieransatz.

Während Scratch zwar eine vollwertige Programmierumgebung darstellt, nimmt eine Erarbeitung in der Regel doch zumindest mehrere Wochen in Anspruch. Code.org kann im Vergleich dazu nicht als volle Programmierumgebung angesehen werden. Jedoch ist es mit code.org auch möglich, kürzere Projekte (bis hin zu einzelnen Projekttagen) damit sinnvoll zu gestalten. Eine Besonderheit von code.org stellt die hochwertige Aufmachung dar. Durch das Sponsoring von großen Unternehmen, kann code.org Lerneinheiten für beliebte Themen, wie *Minecraft*, *Star Wars*, *Angry Birds* und weitere Franchise-Marken, anbieten.

Durch einen kreativen und spielerischen Ansatz sind aber beide Plattformen – sowohl Scratch als auch code.org – in der Lage, Lernende anhaltend zu motivieren.

## Literatur

- Antonitsch, P., Hanisch, L. (2014). *Aspekte von Computational Thinking im Unterricht der Primarstufe*. IMST.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. Freeman.
- Brandhofer, G. (2016) *BeeBot*. <https://eis.ph-noe.ac.at/beebot/>
- Bundesministerium für Bildung, Wissenschaft und Forschung (2023). *Volksschul-Lehrplan*. [https://www.bmbwf.gv.at/Themen/schule/schulpraxis/lp/lp\\_vs.html](https://www.bmbwf.gv.at/Themen/schule/schulpraxis/lp/lp_vs.html)
- Hattie, J. (2013). *Lernen sichtbar machen*. Überarbeitete deutschsprachige Ausgabe. Baltmannsweiler.
- Haus der kleinen Forscher (Hrsg.) (2017). *Informatik entdecken mit und ohne Computer*. Stiftung „Haus der kleinen Forscher“.
- Institut des Bundes für Qualitätssicherung im österreichischen Schulwesen (o. D.). *TIMMS 2019*. <https://www.iqs.gv.at/timss-2019>
- Jahnke-Klein, S. (2013). *Benötigen wir eine geschlechtsspezifische Pädagogik in den MINT-Fächern? Ein Überblick über die Debatte und den Forschungsstand*. In: *Schulpädagogik heute*, 4(8), S. 46–67.
- Kiefer, P. (2019). *Programmieren lernen mit der Maus*. Rheinwerk.
- Mackoff, B. (1998). *Was wollen die Mädchen? 7 Strategien zur Erziehung starker und selbstbewusster Töchter*. Beltz.
- Organisation für wirtschaftliche Zusammenarbeit und Entwicklung (Hrsg.) (2015). *Pisa im Fokus 49*. [https://www.oecd.org/pisa/pisaproducts/pisainfocus/PIF-49%20\(ger\).pdf](https://www.oecd.org/pisa/pisaproducts/pisainfocus/PIF-49%20(ger).pdf)
- Ratz, D., Schulmeister-Zimolong, D., Seese, D. & Wiesenberger, J. (2018). *Grundkurs Programmieren in Java*. Carl Hanser.
- Rogers, H. (1987). *Theory of Recursive Functions and Effective Computability*. MIT Press.
- Sperk, S. (2022). *Girlpower durch Coding ? [!]*. Pädagogische Hochschule Baden.
- Walter, D. (2018). *Programmieren! – Auch schon in der Grundschule? Grundschulunterricht Mathematik*. 1/2018, S. 8–12.
- Westdeutscher Rundfunk (2022). *Programmieren ohne Strom*. <https://www.planet-schule.de/thema-/programmieren-mit-der-maus-unterricht-programmieren-ohne-strom-100.html>
- Wing, J. (2006). *Computational Thinking. Communication of the ACM*. 49 (3), S. 33–35.
- Wurm, B. (2013). *Programmieren lernen: Schritt für Schritt zum ersten Programm*. Galileo Press.