

Programmiersprachen im Unterricht

Die Anzahl unterschiedlicher Programmiersprachen ist riesig. Die meisten Programmiersprachen sind für den professionellen Gebrauch der Informatiker optimiert wie z.B. C++, Java und Python. Andere Programmiersprachen wie z.B. Logo und Scratch wurden extra mit didaktischen Überlegungen entwickelt, um Schülerinnen einen altersgerechten Einstieg in die Programmierung zu ermöglichen. Einige der professionellen Programmiersprachen besitzen auch Erweiterungen für Programmieranfänger wie z.B. Python mit einer steuerbaren virtuellen Schildkröte.

Im Folgenden gehen wir genauer auf die unterschiedlichen Konzepte der verschiedenen Programmiersprachen für den Unterricht ein und evaluieren die in den Schweizer Lehrmitteln gängigen Programmiersprachen.

Entwicklungsumgebung

Zum Programmieren wird häufig eine Entwicklungsumgebung verwendet. Eine solche Umgebung bietet eine grafische Oberfläche, in welcher die Programme geschrieben und ausgewertet werden können. Die Entwicklungsumgebung hat eine wichtige Kernaufgabe: das Umwandeln der Programme in eine maschinen-verständliche Sprache, den sogenannten Maschinencode. Der Computer kennt nämlich nur die beiden Zustände 0 und 1. Alle Vorgänge, welche in einem Computer ablaufen, lassen sich also mit 0en und 1en ausdrücken. Für uns Menschen wäre es allerdings viel zu umständlich, dem Computer nur mit 0en und 1en Anweisungen zu geben. Deshalb wurden Programmiersprachen erfunden, welche es uns ermöglichen, auf effiziente und leichte Art dem Computer Anweisungen zu kommunizieren. Damit der Computer unsere Anweisungen im Programm überhaupt versteht, muss die Entwicklungsumgebung das geschriebene Programm zuerst in den jeweiligen Maschinencode umwandeln. Dieser Umwandlungsprozess heisst Kompilierung. Eine Entwicklungsumgebung ist somit ein vielseitiges Werkzeug, welches die Programmiererin bei ihrer Arbeit unterstützt und ihr zum Beispiel auch bei der Fehlersuche in ihrem Programm hilft.

Zu einer professionellen Programmiersprache werden häufig mehrere unterschiedliche Entwicklungsumgebungen angeboten, so dass jede Person ihren eigenen Vorlieben entsprechend eine Umgebung auswählen kann. Bei den extra für Schüler entwickelten Programmiersprachen ist es häufig so, dass nur eine einzige Entwicklungsumgebung diese Sprache umwandeln und auswerten kann. Somit sind in diesen Fällen die Programmiersprache und die Entwicklungsumgebung stark miteinander verknüpft und wir beziehen uns immer auf eine gemeinsame Einheit, wenn wir später von solchen Programmiersprachen sprechen.

Schülerfreundliche Entwicklungsumgebungen bieten gegenüber professionellen Umgebungen eine reduziertere Funktionalität mit dem Ziel, die wichtigen Funktionen in den Vordergrund zu stellen, ohne dabei die Programmieranfänger mit Funktionen zu überfordern, die sie gar nicht benötigen.

Blockbasierte Programmiersprache

Die Grundbausteine einer blockbasierten Programmiersprache sind, wie der Name bereits sagt, Blöcke. Die Schülerinnen können mit dem Mauszeiger vordefinierte Blöcke auswählen und in beliebiger Reihenfolge zusammenstecken. Wenn sie mit ihrem Programm zufrieden sind, wandelt die Entwicklungsumgebung ihre Blöcke im Hintergrund entweder direkt in Maschinensprache um, oder aber mit

einem Zwischenschritt zuerst in ein textbasiertes Programm und anschliessend in Anweisungen in Maschinensprache, welche dann vom Computer ausgeführt werden können.



Abbildung 1 Ausschnitt aus den Bewegungs- und Kontrollblöcken in der blockbasierten Programmiersprache Scratch.

Mithilfe der Blöcke können die Schüler ein Programm schreiben, um eine Figur zu bewegen. Die blauen Blöcke in Abbildung 1 bewegen direkt die Figur und die gelben Blöcke sind für die Programmkontrolle zuständig. Mit den gelben Blöcken können zum Beispiel Schleifen gebaut werden.

Die einzelnen Blöcke enthalten meistens auch Text, so dass vorausgesetzt wird, dass die Schülerinnen bereits lesen können. Es existieren allerdings auch blockbasierte Programmiersprachen, welche nur Symbole enthalten und von Schülern oder Kindergärtnerinnen ohne Lesekompetenzen verwendet werden können (Staub et al., 2019). Es ist hier also wichtig, dem Level der Schülerinnen entsprechend, eine geeignete blockbasierte Programmiersprache zu wählen.

Blockbasierte Programmiersprachen bieten gerade Schreibanfängern einen Vorteil: Da die Schüler selbst keinen Text schreiben müssen, können sie in ihren Programmen weder Rechtschreib- noch Syntaxfehler machen. Der Computer versteht somit ihre Programme sofort, ohne dass die Schülerinnen zuerst Schreibfehler korrigieren müssen. Dies führt dazu, dass sie ihren Fokus ganz auf den Programminhalt legen können und nicht mit Rechtschreibbefehlern aufgehalten werden. Die Schüler sind ausserdem in der Lage, grosse Programme aus den Blöcken zusammenzubauen.

Eine blockbasierte Programmiersprache kann die Schülerinnen dazu verleiten, Blöcke zusammenzustecken, ohne sich zuvor über das Resultat Gedanken zu machen. Es kann deshalb vorkommen, dass die eigentlichen Programmierkonzepte wie Schleifen, Parameter und der modulare Entwurf *zwischen* den Blöcken untergehen. Dies führt dazu, dass die Schüler die Konzepte im Rahmen der blockbasierten Programmiersprache anwenden können, sich jedoch nicht bewusst sind, wie diese Konzepte konkret funktionieren. Die meist sehr ansprechende grafische Oberfläche kann die Schülerinnen und Lehrpersonen ausserdem dazu verleiten, sich auf hübsche Spielereien zu fokussieren anstatt auf die Grundkonzepte der Informatik.

Damit die Schüler gegen Ende der obligatorischen Schulzeit optimal aufs Gymnasium oder eine Berufslehre vorbereitet sind, ist es unerlässlich, dass sie im Verlauf der Schulzeit zu einer textbasierten Programmiersprache übergehen. In der Industrie wird fast ausschliesslich mit textbasierten Programmiersprachen gearbeitet. Dieser Übergang ist nicht einfach und führt häufig dazu, dass die Schülerinnen ihr Selbstvertrauen verlieren und beim Benutzen der neuen textbasierten Programmiersprache das Gefühl bekommen, dass sie bisher mit den Blöcken gar nicht richtig programmiert haben (Moors, 2016).

Ausserdem kann es vorkommen, dass die Schüler sich von der ganzen Syntax, welcher sie auf einen Schlag ausgesetzt werden, überfordert fühlen.

Ein Beispiel: Scratch

Scratch (<https://scratch.mit.edu>) ist eine blockbasierte Programmiersprache, welche 2007 am MIT (Massachusetts Institute of Technology) in Betrieb genommen wurde und seither von einem ganzen Team weiterentwickelt wird. Die Entwicklungsumgebung läuft in einem Internet Browser, so dass kein Programmdownload nötig ist. Mithilfe von Blöcken können Schülerinnen Programme erstellen. Scratch bietet ausserdem eine grosse Online-Community an, in welcher die erstellten Programme geteilt werden können. Das Ziel des Scratch Teams ist es, Leute zum Programmieren zu begeistern, welche sich bisher nicht in der Rolle des Programmierers gesehen haben (Vardi, 2009). Dafür haben sie eine soziale Plattform geschaffen, welche das einfache Erstellen und Teilen von interaktiven Geschichten, Spielen und Animationen ermöglicht.

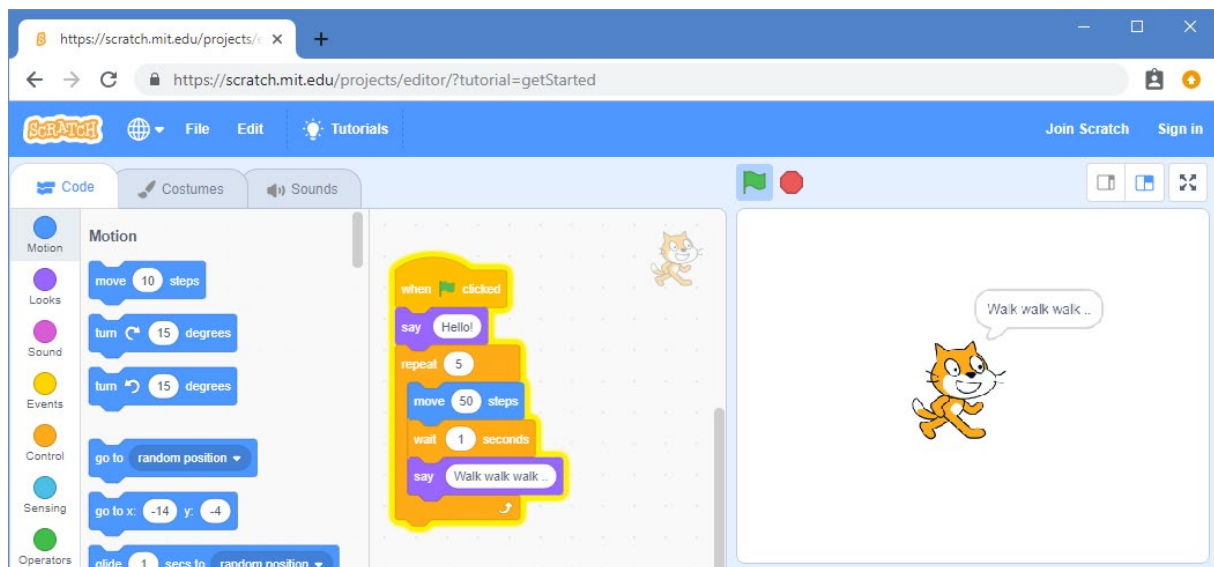


Abbildung 2 Scratch Programm: Die Katze läuft wiederholt nach rechts und redet dabei.

Abbildung 2 zeigt ein kleines Scratch Programm, bei welchem eine Katze über den Bildschirm läuft und dabei etwas sagt. Scratch bietet viele Erweiterungen in den Bereichen Musik, Robotik und Zeichnen. Dies ermöglicht es den Schülern, beim Programmieren verschiedene Interessensgebiete zu verbinden. Mit Figuren und Animationen können sie in kürzester Zeit ganze Geschichten erfinden.

Die Lehrmittel «Connected 1» (Hartmann et al., 2018) und «inform@21» (Straub Haaf & Graf, 2017) setzen beide auf die blockbasierte Programmiersprache Scratch. Scratch geht davon aus, dass die Schülerinnen lesen und schreiben können. Eine blockbasierte Entwicklungsumgebung, welche nur mit Symbolen arbeitet, wird z.B. von xLogoOnline angeboten (siehe Kapitel «Ein Beispiel: Logo mit xLogoOnline»).

Textbasierte Programmiersprache

Mit einer textbasierten Programmiersprache programmieren die Schüler direkt wie eine richtige Informatikerin. Ein Informatiker schreibt zuerst ein Programm, lässt es vom Computer ausführen und sieht anschliessend das Ergebnis. Während der Programmausführung ist nicht sichtbar, was im Computer genau abläuft. Eine Informatikerin braucht deshalb ein gutes Vorstellungsvermögen und viel Erfahrung, um abschätzen zu können, wie der Computer ihre Anweisungen Schritt für Schritt abarbeitet. Aus diesem Grund haben einige textbasierte Programmiersprachen eine fiktive Maschine eingebaut,

welche den Programmcode der Schüler ausführt (Sorva, 2013). In einigen Programmiersprachen wie z.B. Logo und Python übernimmt eine virtuelle Schildkröte diese Aufgabe.

```
forward(40)
```



Abbildung 3 Die Anweisung `forward(40)` wurde von der Schildkröte (fiktive Maschine) ausgeführt.

In Abbildung 3 ist der Zusammenhang zwischen der Programmanweisung und der fiktiven Maschine, der Schildkröte, sichtbar. Die Schildkröte verdeutlicht dabei den Schülerinnen, wie der Computer die einzelnen Anweisungen ihres Programms ausführt. Im vorherigen Kapitel «Ein Beispiel: Scratch» bewegt sich in Abbildung 2 eine Katzenfigur über den Bildschirm. Die Philosophie dahinter ist jedoch eine andere: In Scratch besteht die Hauptaufgabe darin, die Figur mit dem Programm zu bewegen und dadurch eine Geschichte zu erzählen. Die Schildkröte in Logo oder Python dient hingegen dazu, die Anweisungen in den Schülerprogrammen für die Schüler sichtbar zu machen und dadurch ihr Vorstellungsvermögen über die Ausführungsvorgänge des Computers zu stärken.

Mit einer textbasierten Programmiersprache ist es leichter sicherzustellen, dass die Schülerinnen die Programmierkonzepte wirklich verstanden haben. Sie müssen nämlich die gelernten Konzepte wie z.B. Schleifen und den modularen Entwurf selber programmieren und können sie nicht aus vorgefertigten Blöcken zusammenstecken. Der textbasierte Ansatz vermittelt den Schülern ausserdem mehr Selbstvertrauen in ihre Programmierfähigkeiten als der blockbasierte Ansatz (Lewis, 2010).

Schreibfehler spielen bei einer textbasierten Programmiersprache eine Rolle, da der Computer nur korrekt geschriebene Befehle versteht. Viele der schülerfreundlichen Entwicklungsumgebungen versuchen aus diesem Grund, die Schülerinnen auf ihre Schreibfehler aufmerksam zu machen und ihnen bei der Korrektur Unterstützung anzubieten. Das Programmieren mit einer textbasierten Sprache fördert somit gleichzeitig die Schreibfähigkeiten der Schüler. Es kann vorkommen, dass Schülerinnen, welche grosse Mühe mit der Rechtschreibung haben, am Anfang etwas langsamer beim Programmieren sind.

Textbasierte Programmiersprachen haben gegenüber blockbasierten Programmiersprachen den Vorteil, dass die Schülerinnen nicht von Anfang an mit einer Vielzahl von Blöcken konfrontiert werden. Eine textbasierte Programmiersprache kann mit einem kleinen Wortschatz von Befehlen eingeführt werden. Die Schüler lernen mit der Zeit, diesen Wortschatz mit eigenen Befehlen zu erweitern. Bei einer blockbasierten Programmiersprache sind die Schüler von Beginn weg allen Blöcken ausgesetzt und können theoretisch bereits Blöcke einsetzen, von welchen sie die Informatikkonzepte dahinter noch nicht verstanden haben.

Ein Beispiel: Logo mit xLogoOnline

Die Programmiersprache Logo wurde 1967 am MIT von Seymour Papert und seinem Team erfunden (Papert, 1980). Logo richtete sich seit Beginn an Kinder und Programmieranfängerinnen. Mithilfe von wenigen Befehlen können diese einen virtuellen Schildkrötenroboter steuern und so die Informatikkonzepte kennen lernen. Papert war es dabei besonders wichtig, dass die Kinder mit Logo selbständig und auf entdeckende Art lernen, so dass sie selber neue Informatikkonzepte entdecken, ohne dass alle von der Lehrperson vorgetragen werden.

Die ETH Zürich hat 2016 die Entwicklungsumgebung xLogoOnline (<https://xlogo.inf.ethz.ch>) für die Sprache Logo entwickelt. xLogoOnline ist eine webbasierte Entwicklungsumgebung, welche komplett

im Internet Browser läuft und keinen Programmdownload erfordert. Seit 2016 wird xLogoOnline ständig verbessert und weiterentwickelt.

xLogoOnline ist bewusst einfach gehalten, so dass sich die Schüler schnell zurechtfinden und ihre entwickelten Programme im Vordergrund stehen. Das Lehrbuch «Einfach Informatik: Programmieren 5/6» (Hromkovič, 2018) basiert auf der Sprache Logo und verwendet die Entwicklungsumgebung xLogoOnline. Die Schülerinnen lernen mit dem Lehrbuch das Konzept der Schleife kennen, benennen eigene Programme, um grössere Programme mit dem modularen Entwurf zu realisieren und entdecken die Parameter.

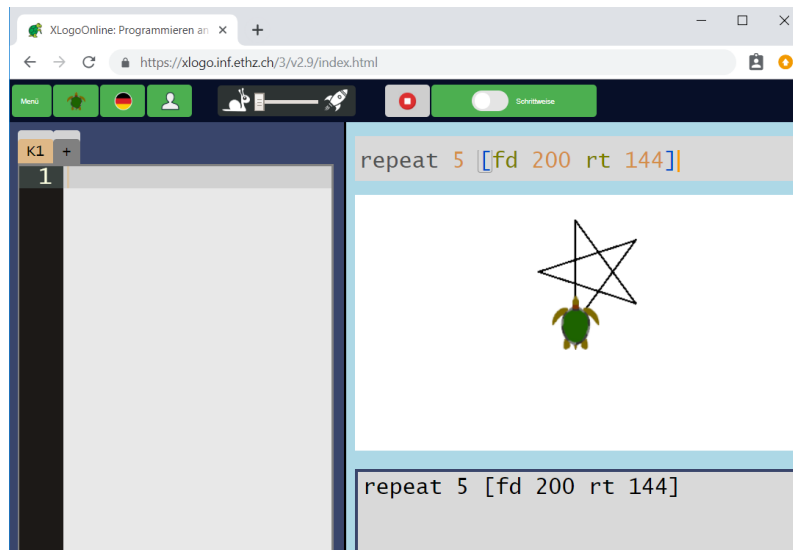


Abbildung 4 xLogoOnline

Abbildung 4 zeigt die Oberfläche der Entwicklungsumgebung xLogoOnline. Mit den Logo-Befehlen wie *fd 200* und *rt 144* wird die Schildkröte gesteuert, so dass sie einen Stern zeichnet. Die Geschwindigkeit der Schildkröte kann eingestellt werden, damit die Schüler der Schildkröte beim Ausführen der einzelnen Befehle besser folgen können.

xLogoOnline bietet neu auch eine blockbasierte Version an. Dabei stellt es zwei Varianten zur Verfügung: eine rein symbolbasierte, die ohne Parameter für die Anzahl Schritte und den Drehwinkel auskommt und somit bereits für Kindergärtnerinnen ohne grosses Zahlenraumverständnis geeignet ist. Abbildung 5 zeigt alle Blöcke der symbolbasierten Variante. Die zweite Variante erlaubt es, die Anzahl Schritte bei der Vorwärts- und Rückwärtsbewegung und den Drehwinkel einzustellen. Die blockbasierte Variante für den Kindergarten und die 1./2. Primarstufe bietet ausserdem die Möglichkeit, einen physischen Roboter (Blue-Bot) anzusteuern.

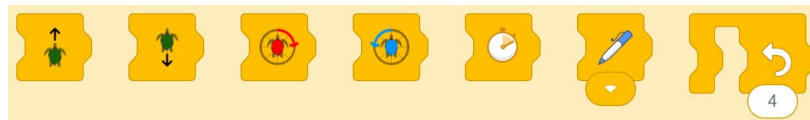


Abbildung 5 Symbolische Blöcke in xLogoOnline

Somit kann Logo, respektive die Entwicklungsumgebung xLogoOnline, vom Kindergarten bis zur 6. Klasse eingesetzt werden. Wobei spätestens zu Beginn der 5. Klasse ein Wechsel auf die textbasierte Variante stattfinden sollte.

Spiralcurriculum

Mit dem Spiralcurriculum Ansatz vom Kindergarten bis zur Matura folgen wir der Idee, dass sich die vermittelten Konzepte und gelernten Elemente mehrfach wiederholen und dabei eine Komplexitätssteigerung erfahren. Im Kindergarten und der 1./2. Primarstufe muss bei der Wahl des Programmierkonzeptes beachtet werden, dass die Kinder noch nicht lesen können oder gerade erst lesen lernen. Ausserdem kann ihre Zählkompetenz noch wenig entwickelt sein. Auf dieser Stufe bietet sich deswegen besonders eine blockbasierte Programmiersprache, welche nur mit Symbolen arbeitet, an. Im Verlauf der weiteren Schuljahre soll ein Übergang zu einer textbasierten Programmiersprache erfolgen, so dass die Schülerinnen am Ende des 3. Zyklus gut mit einer textbasierten Programmiersprache umgehen können. Eine der zentralen Fragen ist es, wann und auf welche Art und Weise der Wechsel von der blockbasierten zur textbasierten Programmiersprache stattfinden sollte.

Das Lehrmittel «Einfach Informatik: Programmieren 5/6» (Hromkovič, 2018) empfiehlt, dass dieser Übergang in der 5./6. Klasse vollzogen wird. Wird dieser Ansatz gewählt, bietet xLogoOnline ebenfalls den Vorteil, dass die Schüler die fiktive Maschine (Schildkröte), welche in der textbasierten Programmiersprache Logo verwendet wird, bereits aus der blockbasierten Version kennen und es zu keinem Wechsel der Entwicklungsumgebung kommt. Die Lehrbuchreihe «Einfach Informatik» bietet ausserdem drei weitere Lehrbücher für die Sekundarstufe 1 an, welche gleich konzipiert sind wie die «Einfach Informatik» Lehrbücher der Primarstufe und deckt somit den Informatikunterricht bis ans Ende der 9. Klasse ab. In der Sekundarstufe 1 wird die professionelle Programmiersprache Python verwendet, welche ebenfalls mit einer virtuellen Schildkröte ausgestattet ist. Somit erlaubt die Lehrbuchreihe «Einfach Informatik» die Anwendung des Spiralprinzips durch den ganzen 2. und 3. Zyklus hindurch.

«Connected 1» vom Lehrmittelverlag Zürich (Hartmann et al., 2018) ist ein Arbeitsbuch, welches für die 5. Klasse ausgelegt ist. Im hinteren Teil des Arbeitsbuches wird kurz mit Scratch gearbeitet. Die Schülerinnen werden darin jedoch noch nicht selber zum Programmieren angeleitet. Für die 6. Klasse erscheint im Juni 2019 der zweite Band «Connected 2», in welchem ein Kapitel dem Programmieren gewidmet sein soll.

«inform@21» vom Lehrmittelverlag St.Gallen (Straub Haaf & Graf, 2017) wurde für die 5./6. Klasse entwickelt. Ein Teilbereich widmet sich der Programmierung mit Scratch. Die Schüler benutzen dabei drei online zugängliche Arbeitsblätter zu den Themen «Figur bewegen», «Wiederholungen» und «Bedingte Bewegung». In den Arbeitsblättern werden den Schülerinnen einige Informatikkonzepte vermittelt, jedoch werden die Schüler nicht dazu motiviert, selbst herauszufinden, wieso diese Konzepte sinnvoll sind.

Literaturverzeichnis

Hartmann, W., Jurjevic, D., Senn, F., Waldvogel, B., & Zuberbühler, U. (2018). *Connected 1: Medien und Informatik für die 5. Klasse*. Zürich: Lehrmittelverlag Zürich.

Hromkovič, J. (2018). *Einfach Informatik: Programmieren 5/6*. Baar: Klett und Balmer Verlag.

Lewis, C. M. (2010). *How Programming Environment Shapes Perception*,.

Moors, L. (2016). *Tackling the Transition from Block-based to Text-based*.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. United States of America: Basic Books, Inc. Publishers.

Sorva, J. (2013). *Notional Machines and Introductory Programming Education*.

Staub, J., Trachsler, N., & Barnett, M. (2019). Programmierunterricht von Kindergarten bis zur Matura in einem Spiralcurriculum.

Straub Haaf, B., & Graf, S. C. (2017). *inform@21*. Rorschach: Lehrmittelverlag St.Gallen.

Vardi, M. (2009). Scratch: Programming for All.

Stand: 7. Mai 2019