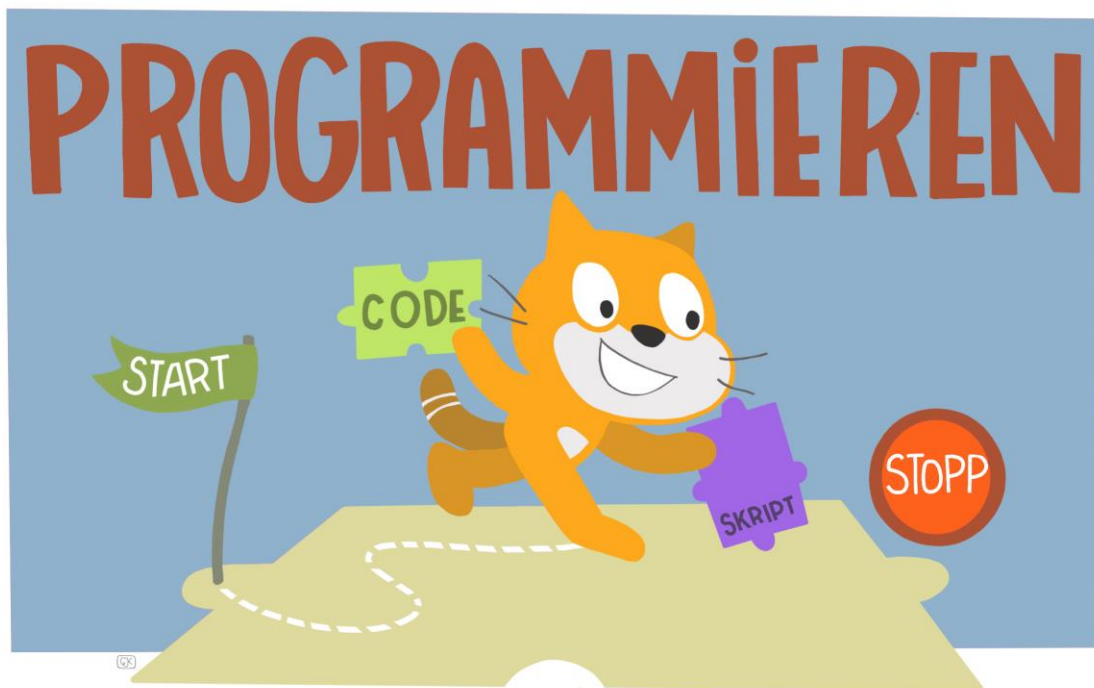


# IT2School

Gemeinsam IT entdecken



## Modul B5 – Programmieren

Leichter ProgrammierEinstieg

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



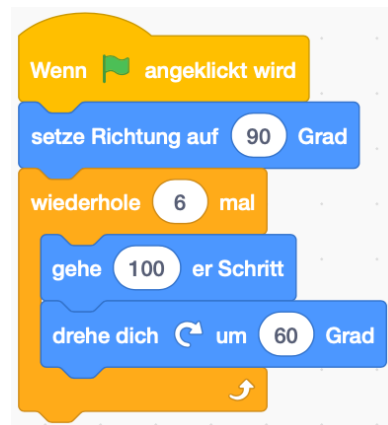
# Inhalt

1	Leichter ProgrammierEinstieg.....	3
2	Warum gibt es das Modul? .....	4
3	Ziele des Moduls.....	5
4	Rolle der Unternehmensvertreter*innen .....	5
5	Inhalte des Moduls.....	5
5.1	Die Entwicklungsumgebung von Scratch .....	5
5.2	Programmieren mit Scratch .....	7
6	Unterrichtliche Umsetzung.....	9
6.1	Grober Unterrichtsplan.....	9
6.2	Stundenverlaufsskizzen .....	10
6.2.1	Einführung in Scratch .....	10
6.2.2	Umsetzung eines eigenen Projekts.....	11
7	Einbettung in verschiedene Fächer und Themen .....	13
8	Anschlussthemen.....	14
9	Literatur und Links .....	15
10	Arbeitsmaterialien .....	15
11	Glossar.....	16
12	FAQs und Feedback.....	18

# 1 Leichter Programmiereinstieg

In diesem Modul sorgt eine einfache Entwicklungsumgebung dafür, dass die Schülerinnen und Schüler erste Erfahrungen in der Programmierung machen können. Als Hilfsmittel hierfür dient die grafische blockbasierte Programmiersprache Scratch.

Eine grafische blockbasierte Programmierung gestaltet den Einstieg für die Schülerinnen und Schüler einfacher als eine textuelle Programmiersprache und erlaubt es trotzdem, selbst komplizierte Anwendungen zu erstellen. Aufgrund der Interaktionsmöglichkeiten verschiedener Objekte (z.B. Unterhaltung zwischen verschiedenen Figuren mittels Sprechblasen) können auch schon sehr junge Schülerinnen und Schüler einen Einstieg in die Programmierung finden. Das schnelle Erfolgserlebnis und das eigenständige und intuitive Zusammensetzen der Programme stärken zudem das Selbstvertrauen.



<b>Lernfeld/Cluster:</b>	IT spielend entdecken	
<b>Zielgruppe/Klassenstufe:</b>	<b>X</b>	4. bis 5. Klasse
	<b>X</b>	6. bis 7. Klasse
	<b>X</b>	8. bis 10. Klasse
	<b>X</b>	11. bis 12. Klasse
<b>Geschätzter Zeitaufwand:</b>	6 bis 7 Einzelstunden (zusätzliche Unterrichtsstunden zur Vertiefung werden empfohlen!)	
<b>Lernziele:</b>	<ul style="list-style-type: none"> <li>• Entwickeln eines grundlegenden Programmierverständnisses</li> <li>• Eigenschaften von Programmen beschreiben</li> <li>• Programme selbst gestalten</li> <li>• Algorithmische Grundbausteine zur Programmierung verwenden</li> <li>• Entwerfen, Implementieren und Testen eigener Programme</li> </ul>	
<b>Vorkenntnisse der Schüler*innen:</b>	Erforderlich: <ul style="list-style-type: none"> <li>• Programme aufrufen und bedienen</li> <li>• Eingaben mit Maus und Tastatur</li> <li>• Laden und Speichern von Dateien</li> </ul> Empfohlen: <ul style="list-style-type: none"> <li>• Internetkenntnisse (zum Hochladen der Programme auf die Scratch-Website)</li> <li>• Grundlegende Erfahrungen mit der Bildbearbeitung (hilfreich bei der Erstellung eigener Figuren und Hintergründe)</li> </ul>	
<b>Vorkenntnisse der Lehrkraft:</b>	Erforderlich: <ul style="list-style-type: none"> <li>• Grundlegende Erfahrungen im Programmieren (durch einmaligen Durchlauf dieses Moduls zu erlangen)</li> </ul>	

<b>Vorkenntnisse der Unternehmensvertreter*innen:</b>	Empfohlen: <ul style="list-style-type: none"> <li>• Grundlegende Erfahrungen im Programmieren, Ausprobieren des Scratch-Tutorials</li> </ul>
<b>Sonstige Voraussetzungen:</b>	Erforderlich: <ul style="list-style-type: none"> <li>• Mindestens 1 Computer pro 2 Schülerinnen und Schüler</li> <li>• Scratch 2 ist in der Regel auf jedem Rechner lauffähig, trotzdem sollte dies vor Unterrichtsbeginn getestet werden</li> <li>• Seit dem Update auf Scratch 3 ist Scratch durch den Einsatz von HTML 5 auch im Browser von Tablets einsetzbar, da seit Scratch 3 kein Adobe Flashplayer mehr benötigt wird</li> </ul> Empfohlen: <ul style="list-style-type: none"> <li>• Beamer zur Präsentation sowie zum gemeinsamen Zeigen und Arbeiten mit Scratch</li> </ul>

## 2 Warum gibt es das Modul?

In diesem Modul wird den Schülerinnen und Schülern Raum für spannende eigenständige Projekte eröffnet. Der Fokus liegt dabei auf der kreativen Gestaltung von Programmen, die mit Hilfe der visualisierten Entwicklungsumgebung Scratch umgesetzt werden.

Durch das Erlernen von Programmiersprachen verstehen Kinder und Jugendliche nicht nur, wie die Geräte, die sie verwenden, funktionieren. Vielmehr erhalten sie die Möglichkeit, durch logische Abfolgen aktiv einzugreifen und Einfluss darauf zu nehmen. Dies könnte eine der wesentlichen Kompetenzen der Zukunft sein. So fasst auch der New Yorker Medienwissenschaftler Douglas Roushko<sup>1</sup> es mit wenigen Worten zusammen: „Program or be programmed“ – was auf Deutsch so viel heißt wie: „Programmiere selbst oder du wirst programmiert“.

Ziel des Moduls ist es daher, das Interesse an Informationstechnik bei Kindern und Jugendlichen zu fördern und daraus folgend auch ihre gesellschaftliche Mündigkeit zu unterstützen. Sie sollen die Grundlagen der Technik verstehen, einfache Softwarekomponenten selbst programmieren und eigene innovative Lösungen entwickeln. Dadurch werden auch das logische Denken sowie Problemlösekompetenz gefördert.

Während der aktiven Auseinandersetzung und Entwicklung eigener Projekte erfahren die Kinder und Jugendlichen etwas über moderne Arbeitsprozesse und bekommen die Möglichkeit, die digitale Gesellschaft mitzugestalten. Zu Beginn dieses Moduls steht die Heranführung an die Programmierung im Vordergrund. Es wird dabei auch aufgezeigt, wie Aspekte des Problemlösens und der Algorithmisierung, also einer eindeutigen Handlungsvorschrift zur Lösung von Problemen, vertieft werden können.

<sup>1</sup> <http://www.rushkoff.com/program-or-be-programmed/>

### 3 Ziele des Moduls

- Kennenlernen der Grundkonzepte der Computerprogrammierung
- erleben, wie (leicht) kleine Programme gestaltet werden können
- Softwareerstellung als kreativen Prozess erkennen und erleben
- Beschreiben der Eigenschaften von Programmen
- Verwendung von algorithmischen Grundbausteinen zur Programmierung

### 4 Rolle der Unternehmensvertreter\*innen

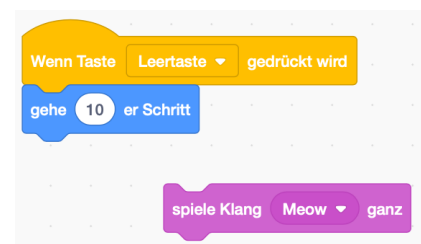
Im *Modul B5 – Programmieren* hat der\*die Unternehmensvertreter\*in mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Unterstützung der Lehrkraft, z. B. bei der Einführung in die Scratch-Programmierz-Oberfläche
- Unterstützung der Schülerinnen und Schüler bei Umsetzung ihrer eigenen kreativen Projekte
- Sie oder er kann zur Abschlusspräsentation der Projektergebnisse als Special-Guest eingeladen werden.

### 5 Inhalte des Moduls

Scratch ist eine visuelle blockbasierte Programmiersprache, die von der Lifelong Kindergarten Group des MediaLab am Massachusetts Institute of Technology (MIT) entwickelt wurde, um insbesondere Kinder und Jugendliche mit der Computerprogrammierung vertraut zu machen.

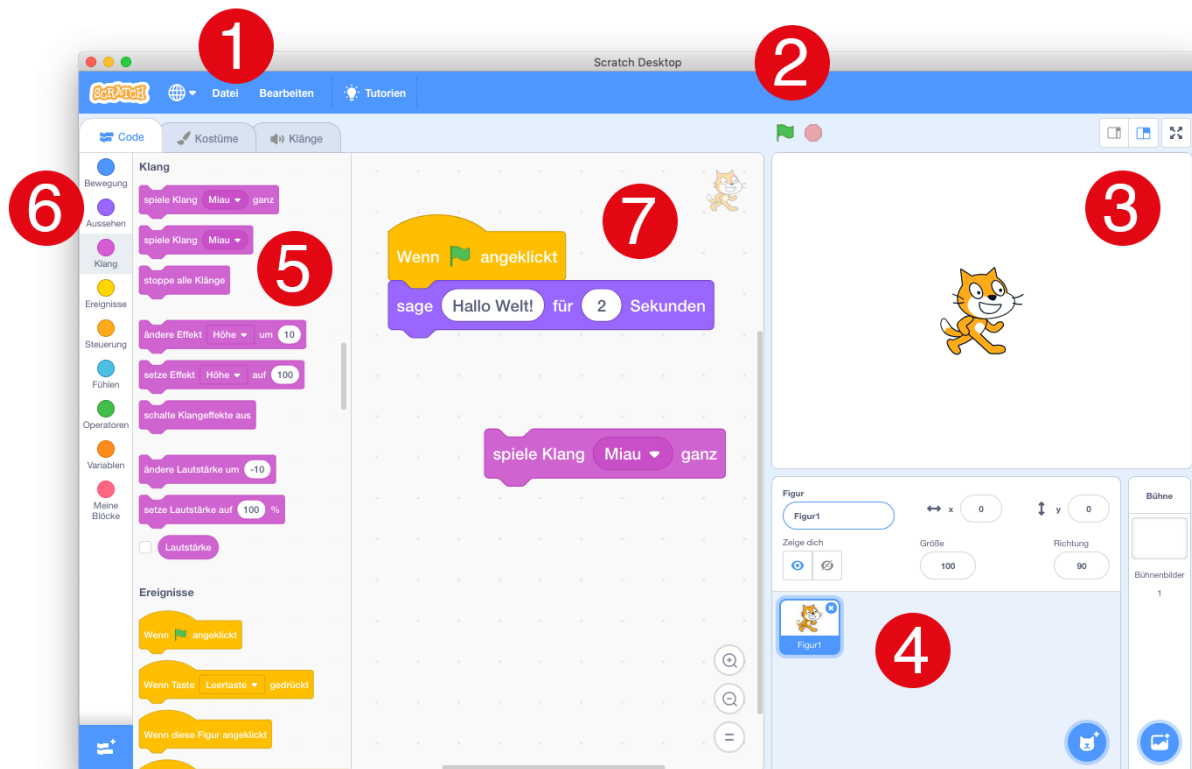
In dieser Entwicklungsumgebung werden Befehle in Form von Blöcken beziehungsweise Bausteinen dargestellt und können ähnlich wie Puzzleteile „aneinandergesteckt“ werden. Auf diese Weise müssen Schülerinnen und Schüler keine Programmierbefehle auswendig lernen oder Kommata und sonstige Sonderzeichen beachten, wie es sonst bei textuellen Programmiersprachen (z. B. C+, Java, Visual Basic) nötig ist. Die Programmierung wird auf das Wesentliche reduziert, wodurch auch Anfänger sehr schnell interessante Ergebnisse erzielen. Diese schnellen Erfolgserlebnisse fördern die Motivation sowie das Selbstvertrauen der Schülerinnen und Schüler. Durch die freie Verfügbarkeit von Scratch ist es außerdem möglich, Freunden und Verwandten die selbstgestellten Programme zu Hause vorzuführen und das Neuerlernte so zu teilen.



#### 5.1 Die Entwicklungsumgebung von Scratch

Für die Entwicklung von Programmen mit Scratch kann ein Offline- oder ein Online-Editor genutzt werden. Das bedeutet, Sie können ohne aufwendiges Installieren von Programmen online direkt im Browser (Online-Editor) oder unabhängig von der Internetverbindung offline mit einem Programm (Offline-Editor *Scratch Desktop*) programmieren. Für die Schule ist der Off-

line-Editor zu empfehlen, um nicht von der Internetverbindung der Schule abhängig zu sein und das Netz weniger zu belasten.



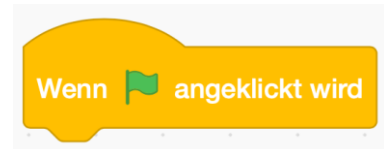
Im Folgenden werden die nummerierten Bereiche der Arbeitsumgebung beziehungsweise des Editors genauer erklärt:

1. Die verschiedenen Reiter, wie zum Beispiel **Datei** oder **Bearbeiten**, stellen das Menü des Programms dar. Diese Menüs bieten unter anderem Optionen zum Speichern und Laden von Programmen.
2. Mit der *grünen Fahne* werden üblicherweise Programme in Scratch gestartet und mit dem *roten Sechseck* gestoppt. So lässt sich ansehen, was man bisher programmiert hat.
3. Auf der *Bühne* (Stage) werden die *Figuren* (auch Objekte oder Sprites genannt) positioniert und führen die Befehle aus, die man programmiert hat. Alle sichtbaren Befehle der Scratch-Programme werden hier dargestellt.
4. In diesem Bereich werden alle Figuren, die im Programm enthalten sind, aufgelistet. Nicht alle Figuren müssen immer auf der Bühne sichtbar sein, sie können sich auch verstecken oder erst zu einem späteren Zeitpunkt erscheinen. Links daneben sind alle Hintergründe für die Bühne aufgelistet.
5. Anweisungen beziehungsweise *Befehle*, die der Computer für die Ausführung von Programmen benötigt, sind hier in Form von *Bausteinen* dargestellt. Der Computer arbeitet diese Schritt für Schritt ab.
6. Die Programmieranweisungen sind in Kategorien zusammengefasst. Unter **Bewegung** sind alle Programmierbefehle zu finden, um einzelne Figuren zu bewegen, unter **Aussehen** findet man Kostümwechsel für die Figuren, aber auch Sätze, die gesagt werden können.

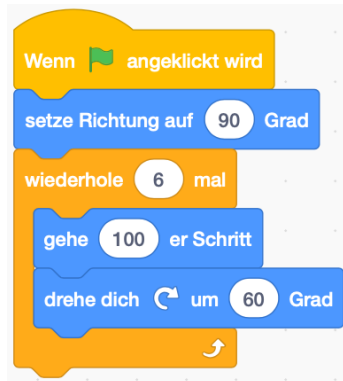
7. Der große weiße Bereich in der Mitte des Fensters wird als das Skriptfenster bezeichnet. Um ein Programm zu schreiben, werden die einzelnen Bausteine mit gedrückter Maustaste in diesen Bereich gezogen. Wenn also der Baustein „**wenn Leertaste gedrückt – gehe 10er-Schritt**“ hier positioniert wird, geht die Katze bei entsprechendem Tastendruck vorwärts. Mehrere solcher zusammengesetzter Bausteine nennt man *Skript*, wohin gegen die Gesamtheit aller Figuren, Bühnen und Skripte als *Programm* bezeichnet wird.

## 5.2 Programmieren mit Scratch

Wie bereits im Abschnitt 3.1.1 erwähnt, werden Programme in Scratch in der Regel durch das Klicken der grünen Fahne gestartet. Bei „Wenn [grüne Fahne] angeklickt“ handelt es sich um ein Ereignis. So findet man auch diesen Ereignisbaustein in der Kategorie „Ereignisse“.



An das Ereignis können weitere Bausteine angeordnet werden. Dazu werden sie einfach an die gewünschte Position gezogen. Ein so geschaffenes Skript wird immer von oben nach unten Schritt für Schritt abgearbeitet. Ein Skript kann wie folgt aussehen:

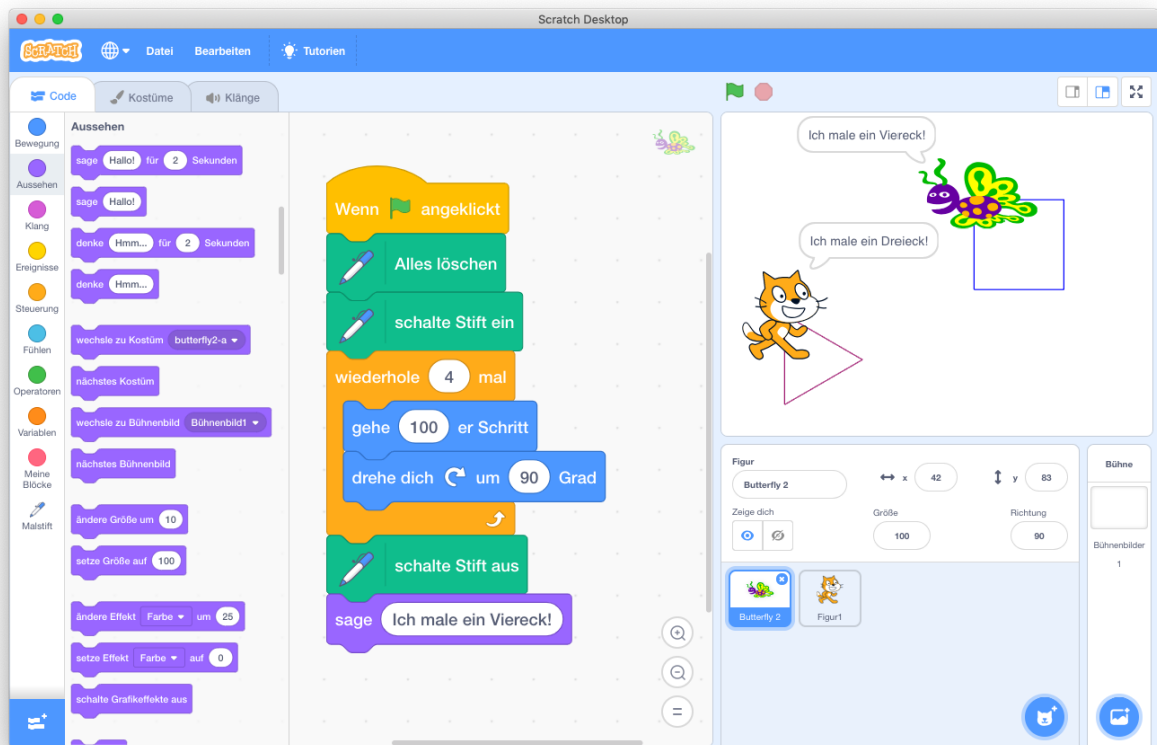


Dieses Beispiel würde dafür sorgen, dass die Figur zu Beginn nach rechts schaut und dann einmal ein Sechseck („6 mal“ „drehe dich um 60 Grad“) mit einer Schrittweite von 100 („6 mal“ „gehe 100er-Schritt“) abläuft.

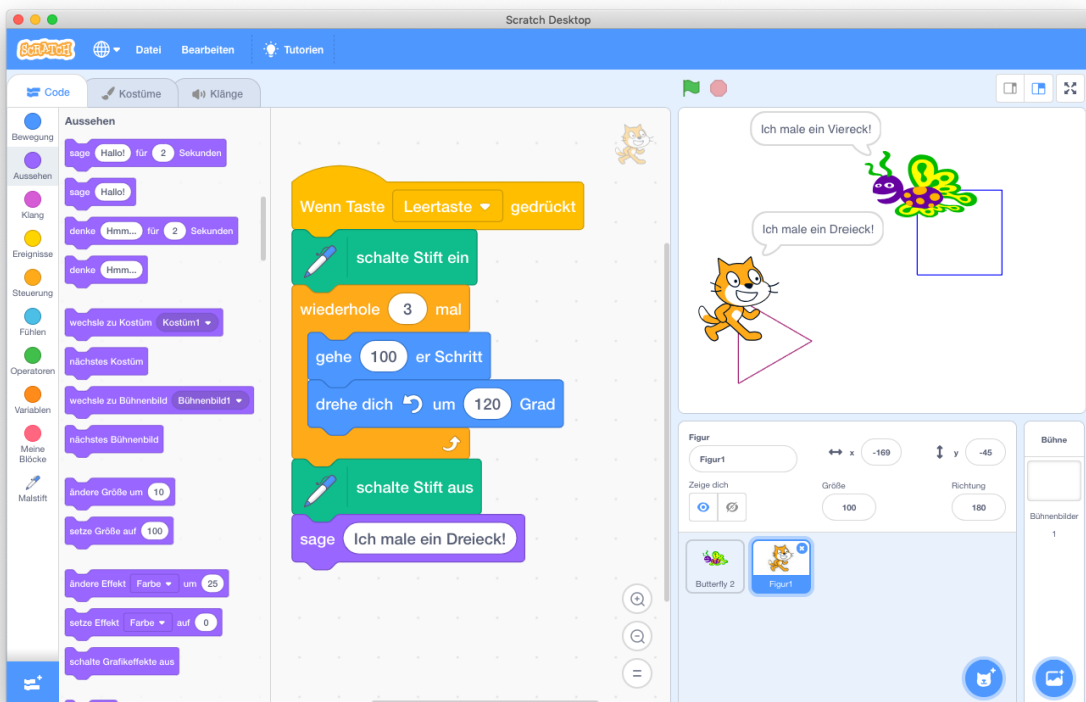
So würde sich aber nur die Figur verhalten, der dieses Skript zugeordnet wurde. Fügt man in den Bereich 4 (siehe Abschnitt 3.1.1) nun eine weitere Figur hinzu und klickt diese an, dann leert sich Bereich 7 und man sieht nur die Skripte dieser Figur. Dies kann wie folgt aussehen<sup>2</sup>:

<sup>2</sup> Hinweis: Der Malstift ist seit Scratch 3 unter den Erweiterungen zu finden.





Dasselbe Scratch-Projekt sieht bei Auswahl der Katze wie folgt aus:





## 6 Unterrichtliche Umsetzung

Auch auf der Webseite von Scratch finden sich zahlreiche Animationen, Spiele und kleine interaktive Geschichten, die einen großen Aufforderungs- und Nachahmungscharakter besitzen. Sie sollen genutzt werden, um bei den Schülerinnen und Schülern Neugierde und die Lust zu wecken, selbst mithilfe von Scratch solche Filme und Spiele zu programmieren. Danach werden die Schülerinnen und Schüler schrittweise mit der Entwicklungsumgebung von Scratch und den einzelnen Bausteinen vertraut gemacht. Dabei sollte darauf geachtet werden, dass sie schnell erste Erfolge durch das Erstellen eigener Programme erreichen. Hierfür wurde ein Tutorial entwickelt, das die Schülerinnen und Schüler schrittweise durchgehen können, um kleine Programmiersequenzen zu erstellen. Zum Abschluss können sie das Gelernte in ein größeres Projekt einfließen lassen und selbst kreativ werden.

Als Ergänzung zu Scratch bietet sich das Modul B6 Mein Anschluss an. Hier bietet der MocoMoco als Eingabegerät die Möglichkeit noch kreativere und interaktivere Programme und Projekte zu entwickeln und zu gestalten.

### 6.1 Grober Unterrichtsplan

Unterrichtsszenarien	Kurze Zusammenfassung
Einführung und erste Schritte mit Scratch	Die Schülerinnen und Schüler werden für Scratch motiviert und die Entwicklungsumgebung wird ihnen anhand eines Tutorials und mit Unterstützung der Lehrkraft nähergebracht.
Einarbeitung und Vertiefung in Scratch	Die Schülerinnen und Schüler entwickeln eine eigene Idee und setzen diese um.
Präsentation	Die programmierten Geschichten und Computerspiele werden präsentiert.

## 6.2 Stundenverlaufsskizzen

### Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschüler\*innen; SuS = Schüler\*innen; UV = Unternehmensvertreter\*in

### 6.2.1 Einführung in Scratch

Zeit	Phase	Sozialform/ Impuls	Inhalt/Unterrichtsgeschehen	Material
	Vorbereitung		Es sollte sichergestellt sein, dass Scratch entweder als Offline- oder Online-Editor auf den Computern der SuS funktionstüchtig ist. Die Editoren können unter folgenden URLs aufgerufen werden: <ul style="list-style-type: none"> <li>• <a href="https://scratch.mit.edu/scratch2download/">https://scratch.mit.edu/scratch2download/</a></li> <li>• <a href="https://scratch.mit.edu/projects/editor/">https://scratch.mit.edu/projects/editor/</a></li> </ul>	
5 Min.	Einstieg	Lehrkraftpräsentation	Begrüßung SuS; ggf. Vorstellung UV; als Einstieg werden ein bis zwei Programme mit Scratch vorgeführt, um Neugierde zu wecken und die Lust, es selbst auszuprobieren. Beispiele sind auf der Scratch-Internetseite zu finden: <a href="https://scratch.mit.edu/starter_projects/">https://scratch.mit.edu/starter_projects/</a>  Beispielprojekte: <ul style="list-style-type: none"> <li>• <a href="https://scratch.mit.edu/projects/1192/">https://scratch.mit.edu/projects/1192/</a></li> <li>• <a href="https://scratch.mit.edu/projects/13701368/">https://scratch.mit.edu/projects/13701368/</a></li> </ul>	Computer, Beamer, Internetverbindung
15 Min.	Hinführung	Lehrkraftvortrag	Nach dem Einstieg führt L kurz in die Entwicklungsumgebung ein. Folgende Fragestellungen sollten dabei beantwortet werden: <ul style="list-style-type: none"> <li>• Was ist Scratch?</li> <li>• Wie kann/soll das Programm aufgerufen werden?</li> <li>• Wie werden Programme, die mit Scratch geschrieben wurden, aufgerufen?</li> <li>• Wie ist die Entwicklungsumgebung in groben Zügen aufgebaut?</li> </ul>	

70 Min.	Erarbeitung	Einzel- oder Gruppenarbeit	SuS erhalten B5.1, arbeiten es durch und lösen die darin enthaltenen Aufgaben.	B5.1 (Einstiegstutorial)
20 Min.	Erarbeitung	Tandemarbeit	Im Anschluss vertiefen SuS ihre Kenntnisse mithilfe der Scratch-Würfel. (Wenn diese noch nicht fertig zusammengebaut sind, muss ggf. mehr Zeit eingeplant werden!) Jedes Team erhält ein Set Scratch-Würfel (B5.2 ist die Bauvorlage) sowie B5.3 zur Bearbeitung. Es wird dann geschaut, was sich aus den durch den Würfel vorgegebenen Bausteinen programmieren lässt. Dadurch entstehen verschiedene Zufallsergebnisse.	B5.2 (Bauvorlage), B5.3 (Arbeitsauftrag)
10 Min.	Sicherung	Präsentation	Die Gruppen präsentieren und erklären die entstandenen Skripte. Gegebenenfalls können die Ergebnisse in einem Wiki oder auf einer Lernplattform festgehalten werden.	Computer, Beamer

Zur Einführung finden Sie weitere Aufgabenstellungen und Arbeitsblätter unter „Zusatzmaterial“ (B5.5 – B5.8). Diese Arbeitsblätter behandeln Schritt für Schritt den Aufbau von Scratch sowie den Funktionsumfang der Entwicklungsumgebung. Dies ist vor allem für Schülerinnen und Schülern interessant, die Schwierigkeiten mit der offenen Projektarbeit haben oder mehr Hilfestellungen benötigen. Im weiteren Verlauf dieses Unterrichtsszenarios wird ein eigenes Programm der Schülerinnen und Schüler umgesetzt. Es ist aber genauso denkbar B3.5 – B3.8 zu bearbeiten.

### 6.2.2 Umsetzung eines eigenen Projekts

Zeit	Phase	Sozialform/ Impuls	Inhalt/Unterrichtsgeschehen	Material
10 Min.	Einstieg	Plenum	Begrüßung, L klärt über den Stundenverlauf auf, SuS setzen ein eigenes Programm in Teams oder Kleingruppen (unter Verwendung von B5.4) um. SuS können wählen zwischen: <ul style="list-style-type: none"> <li>• Computerspiel</li> <li>• animierte Geschichte</li> </ul> L fungiert als Lernbegleiter*in und unterstützt die Gruppen nach Bedarf. Zu	B5.4 (Projektaufgabe)

			Beginn können Internetlinks, beispielsweise zum Scratch-Wiki, an die SuS werden, damit sie auftretende Probleme eigenständig lösen können. <sup>3</sup>	
30–45 Min.	Projektplanung	Gruppenarbeit	<p>Aufteilung in Gruppen, die Gruppen entscheiden sich für ein Thema und gehen es wie folgt an:</p> <ul style="list-style-type: none"> <li>• Entwicklung einer Idee</li> <li>• Storyboard für Geschichte oder Konzept und Design für das Spiel entwickeln (alles wird mit Stift und Papier festgehalten)</li> <li>• Aufgaben in der Gruppe werden bei Bedarf verteilt</li> </ul>	Stifte, Papier zur Ideenfindung
60–120 Min.	Durchführungsphase	Gruppenarbeit	<p>SuS setzen ihre Ideen mit Scratch um:</p> <ul style="list-style-type: none"> <li>• Programmierung</li> <li>• Ausprobieren (Testing)</li> <li>• Fertigstellen</li> </ul>	Computer mit Internetverbindung
10 Min.	Präsentation	Plenum	SuS präsentieren ihre Animationen und Spiele und laden sie gegebenenfalls auch auf die Scratch-Webseite hoch.	Computer, Beamer
15 Min	Reflexion	Plenum	Auseinandersetzung mit dem Prozess: Was hat geklappt? Was hat nicht geklappt? Wie war das Arbeiten im Team? Gab es Probleme? Sollte man öfter Projektarbeit einsetzen?	

**Hinweis:** Vergessen Sie nicht, auch die Möglichkeiten des MocoMoco aus dem Modul B6 in den Folgeunterricht einfließen zu lassen.

<sup>3</sup> Weitere Informationen zu der Projektmethode finden Sie im Methodenmodul M2.

## 7 Einbettung in verschiedene Fächer und Themen

Die Erstellung von Filmen und Geschichten sowie deren Planung mithilfe eines Storyboards stellt Bezüge zum Fach Deutsch her. Im Einzelnen hervorzuheben sind hier die prozessbezogene Kompetenz „Schreiben“ sowie die inhaltsbezogene Kompetenz „Texte und Medien“ mit dem Schwerpunkt Medien (Kompetenzen nach KMK).

Die grafische Gestaltung ebenso wie die Planung eines möglichen Films lassen sich auch mit dem Unterrichtsfach Kunst verbinden.

Durch die Bezüge zu unterschiedlichen Fächern kann das gesamte Modul oder können einzelne Teile in verschiedenen Fächern eingesetzt werden. Die folgenden Kompetenzen finden sich entweder in den Bildungsstandards der Kultusministerkonferenz oder in den einzelnen Rahmenlehrplänen der Länder wieder:

### Deutsch

Die Schülerinnen und Schüler ...

- können handelnd mit Texten umgehen, mit Sprache experimentell und spielerisch umgehen z. B. illustrieren, inszenieren, umgestalten, ... (Grundschule).
- können verschiedene Medien für Präsentationen nutzen (Grundschule).
- Texte (medial unterschiedlich vermittelt) szenisch gestalten (Sek I).
- Texte mit Hilfe von neuen Medien verfassen (Sek I).

### Kunst

Die Schülerinnen und Schüler ...

- setzen digitale Medien für eine Gestaltungsaufgabe ein.
- setzen designorientierte Findungsprozesse und Lösungsstrategien ein.
- entwickeln verschiedene Ideen in einem Entwurfsprozess und stellen den Entwurf angemessen dar.
- realisieren filmische Projekte (Stopmotion, Video, Computeranimationen).

### Informatik

Die Schülerinnen und Schüler ...

- erstellen Produkte unter Anwendung fortgeschrittener Techniken von Standardsoftware; falls es das Produkt erfordert, arbeiten sie sich in geringem Umfang in Spezialsoftware ein.
- erwerben beim Bearbeiten von Softwareprojekten in angemessenem Umfang Kenntnisse über Analyse- und Modellierungsverfahren sowie Projektmanagement.
- verstehen Programmabläufe und die Arbeitsweise von Schnittstellen.
- können erdachte Systeme in technische Systeme übertragen.

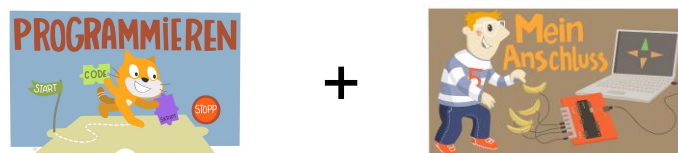
- kennen sich in Entwicklungsumgebungen/Programmierungsumgebungen aus.
- eine Spielumgebung entsprechend der Spielidee gestalten.
- eine Spielvariante mit informatischen Werkzeugen entwerfen und implementieren.

## 8 Anschluss Themen

Als Anschluss Themen im Zusammenhang mit IT2School bieten sich folgende Bausteine an:

### Beispiel: Mein Anschluss

Wenn Sie das Gesamtpaket von IT2School in Form einer Bildungspartnerschaft erhalten haben, so besitzen Sie auch einen Klassensatz von MocoMocos. Dies bietet die Möglichkeit die Programmierung durch weitere Projekte mit Einbindung des MocoMocos auszubauen.



### Beispiel: Programmieren

Eine zu Scratch ähnliche Entwicklungsumgebung stellt auch der App Inventor dar. Dieser erlaubt die Gestaltung und Programmierung eigener Apps für Android Smartphones.

Im Anschluss an die visuelle Programmierung kann die textuelle Programmierung zur Vertiefung herangezogen werden.



Tipp: Darüber hinaus bietet Scratch noch viele weitere Möglichkeiten zur Erweiterung und Vertiefung:

- **Das Pico-Board**

An den Computer kann ein sogenanntes Pico-Board (Mikrocontroller-Board) angeschlossen werden. Das Pico-Board ist speziell für Scratch entwickelt worden und integriert einen Drucktaster, einen Lichtsensor, einen akustischen Sensor sowie vier weitere Anschlüsse für Sensoren (z. B. Bewegungssensor). Durch das Pico-Board wird Scratch interaktiv und eröffnet viele Projektideen für den Unterricht.

- **Angebote von Makeblock**

Die Firma Makeblock bietet ein vielfältiges Baukastensystem aus elektronischen und mechanischen Teilen an, aus denen Roboter, Maschinen und andere Konstruktionen gebaut werden können. Alles lässt sich mit Scratch programmieren. Insbesondere das „Makeblock Inventor Electronic Kit“ ist interessant. Es beinhaltet zwölf Sensoren und Aktoren und bietet alles für schnelles Prototyping.



- **Squeak und BYOB**

Auf Scratch aufbauend kann es sich anbieten, mit den Open-Source-Entwicklungsumgebungen Squeak und Snap/BYOB (Build Your Own Blocks) weiterzuarbeiten. Squeak unterstützt den Textmodus und weitere Konzepte, die nicht in Scratch vorhanden sind. In BYOB können sogar eigene Bausteine erstellt werden und Netzwerkkommunikation ist möglich.



## 9 Literatur und Links

- Glöde, Martina & Reit, Birgit (Hg.) (2015): **Programmieren supereasy. Einfacher Einstieg in Scratch und Python**. München: Dorling Kindersley.
- Bartmann, Erik (2015): **Faszinierende Elektronik-Projekte mit Scratch, Arduino und Raspberry Pi**. 1. Aufl. Beijing: O'Reilly (O'Reillys basics).
- Engeln, Frank (2010): **Scratch me if you can – Programmieren mit Scratch, Lego® WeDo und Picoboard**. Online: <https://collection.switch.ch/objects/LOR:6384/datastreams/DS5>
- **Offizielle Scratch-Website**, Beispiele, Tipps – online: <https://scratch.mit.edu>
- Deutsches **Scratch-Wiki** – online: <http://www.scratch-dach.info/>
- Konstruktionsplattform **Makeblock**. Baukastensystem mit Sensoren und Aktoren zum Programmieren mit Scratch. Online: <http://www.makeblock.cc/>
- **Pico-Board**. Mikrocontroller speziell für Scratch. Online: <http://scratch-dach.info/wiki/PicoBoard>

## 10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
 B5.1	Die ersten Schritte mit Scratch	Arbeitsblatt mit Einstiegstutorial zum Kennen- und Anlernen der Entwicklungsumgebung Scratch.
 B5.2	Scratch-Würfel	Bau-/Kopiervorlage für ein Set Scratch-Würfel mit Bausteinen aus Scratch zum Programmieren von zufälligen Programmen.
 B5.3	Ein Scratch-Programm erwürfeln	Arbeitsblatt für das Scratch-Würfel-Set. Nutzung für den kreativen Entwurf eines zufälligen Programms mit Scratch.
 B5.4	Projektarbeit	Arbeitsblätter mit Hinweisen und Aufgaben für eine Projektarbeit mit Scratch.
 B5.5	Scratch kennenlernen	Arbeitsblatt als Alternativeinstieg. Analyse und Vergleich verschiedener Programme.
 B5.6	Bühnenbild und Bewegungen	Arbeitsblatt als Alternativeinstieg. Umgang mit dem Bühnenbild und Bewegungen.



 B5.7	Animationen und Geschichten erzählen	Arbeitsblatt als Alternativeinstieg. Erstellung von Animationen mit Scratch und Programmieren einer Kurzgeschichte.
 B5.8	Nachrichten und Variablen	Arbeitsblatt als Alternativeinstieg. Umgang mit Nachrichten und Variablen.

## Legende



Material für Schülerinnen und Schüler



Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter



Zusatzmaterial

## 11 Glossar

Begriff	Erläuterung
Algorithmus	Als Algorithmus wird eine Handlungsvorschrift bezeichnet, deren einzelnen Handlungsanweisungen eindeutig und deterministisch (endlich; zeitlich begrenzt) sind. Algorithmen beschreiben meist, wie gegebene Problemstellungen gelöst werden oder bestimmte Tätigkeiten durchzuführen sind.
Attribut	Ein Attribut ist eine zu einem Objekt (in Scratch eine Figur) gehörende Variable, mit deren Hilfe die Datenwerte einer bestimmten Eigenschaft verwaltet werden. Solche Datenwerte nennt man auch Attributwerte.
Attributwerte	vgl. Attribut.
(Scratch-)Bausteine	Bausteine werden genutzt, um Programme in Scratch zu erzeugen. Die Bausteine lassen sich wie Puzzleteile miteinander verbinden – es passt nur zusammen, was einen syntaktisch richtigen Quellcode ergibt, dadurch werden Syntaxfehler verhindert. (Scratch-)Bausteine werden auch <i>Blöcke</i> genannt.
Blöcke	vgl. (Scratch-)Bausteine.
Bühne	Die Bühne ist der Bereich des Scratch-Programms, den der Nutzer sieht. Auf ihr sind die verschiedenen Aktionen der einzelnen Figuren sichtbar. Analog zu den Figuren kann auch der Hintergrund der Bühne nach Belieben angepasst und innerhalb der Programmlaufzeit verändert werden.
Entwicklungsumgebung	vgl. IDE.
Figur	Figuren werden in Scratch auf der → Bühne (Stage) platziert und können dort Aktionen wie Bewegungen und Aussagen (Sprechblase, Gedankenblase) ausführen. Sie müssen nicht immer auf der Bühne sichtbar sein. Während der Programmlaufzeit kann die Darstellungen (deren Bilder) verändert werden.

IDE	Ist eine Abkürzung und steht für „Integrierte Entwicklungsumgebung“; abgeleitet von englisch „integrated development environment“. Gemeint ist hier die Umgebung beziehungsweise das Programm, mit dem in Scratch programmiert wird.
(Scratch-) Kategorien	Die Scratch-Programmoberfläche ist in drei Hauptbereiche (→ Bühne, Liste der → Bausteine und → Skriptfenster) eingeteilt. Im Bereich Kategorien sind ähnliche Bausteine zusammengefasst und farbkodiert. Kategorien von Scratch 2: Bewegung, Ereignisse, Aussehen, Steuerung, Klang, Fühlen, Malstift, Operatoren, Daten, weitere Bausteine. Die Kategorien der Programmoberfläche sind farbcodiert.
Kontrollstruktur	Eine Kontrollstruktur stellt einen Scratch-Baustein dar, mit dem der Ablauf des Programms gesteuert werden kann. Beispiele hierfür sind die [Wenn]-, [Wenn, Sonst]- oder [Wiederhole]-Bausteine.
Methode	vgl. Skript.
Objekt	vgl. Figur.
Programm	Bezeichnet die Gesamtheit aller verwendeten → Figuren, → Bühnen und → Skripte in Scratch. Programme können mit der grünen Fahne ausgeführt und dem roten Punkt gestoppt werden.
Programmierungsumgebung	vgl. IDE.
Sequenz	Unter einer Sequenz wird eine Abfolge von Anforderungen verstanden, die nacheinander abgearbeitet werden. Ähnlich aufzufassen wie → Skript.
Skript	Ein Skript besteht aus mehreren Handlungsanweisungen beziehungsweise Bausteine und wird einer Figur zugeschrieben.
Skriptfenster	Einer der Hauptbereiche der Scratch-Entwicklungsumgebung. Die Bausteine werden per Drag-and-Drop in das Skriptfenster hineingezogen und können dort zu Skripten zusammengesetzt werden.
Sprite	vgl. Figur.
Stage	vgl. Bühne.
Storyboard	Ein Storyboard bildet die Inhalte und Handlungsverläufe eines Drehbuchs visuell ab und dient so als Vorlage für die Erstellung eines Films. Die Handlungen aller Akteure werden hierbei auf einem Zeitstrahl dargestellt.

**Anmerkung:** Bezeichnungen in der Programmoberfläche können lokalisiert beziehungsweise individuell angepasst werden. Darum sind die Bezeichnungen nicht immer einheitlich.

## 12 FAQs und Feedback

Stolpersteine, Lessons learnt und Frequently Asked Questions (FAQs) finden Sie unter:



**<https://tinyurl.com/IT2S-FAQ>**

Wir sind auf Ihr Feedback zum Modul gespannt. Lassen Sie uns wissen, was Ihnen gefallen hat und wo Sie Verbesserungspotential sehen:



**<https://www.surveymonkey.de/r/QM82XWN>**