|  |
| --- |
| **Aufgabe 1a: Zeichnungsdiktat** |
| Färbe im Raster einige Felder so ein, dass ein Bild entsteht (ein Tier, ein Haus, etc.). |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Du bist Programmierer. Dein Partner muss dein Pixebild zeichnen, ohne dass sie / er dieses vorher gesehen hat. Dafür braucht es klare Anweisungen. Schreibe einen «Code», welcher gut verständlich dein Bild beschreibt. | |
|  | |
| **\\media.int\srf-tpc\EFS\JugendFamilie\2 Familie\SRF mySchool\02 PROJEKTE\Frag Fred\03 Unterrichtsmaterial\Bildmaterial für Umat\fred-freisteller1.jpg**  *Bild: SRF mySchool* | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |
| --- |
| **Aufgabe 1b: Zeichnungsdiktat** |
| Jetzt bist du ein Roboter und zeichnest die mündlichen Anweisungen deines Partners. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Stimmen eure Bilder überein? Waren die Anweisungen richtig? Vergleicht eure Bilder und tauscht euch aus. Welche Tipps hast du für den Programmierer? Was könnte er / sie noch besser machen? | |
|  | |
| **\\media.int\srf-tpc\EFS\JugendFamilie\2 Familie\SRF mySchool\02 PROJEKTE\Frag Fred\03 Unterrichtsmaterial\Bildmaterial für Umat\fred-freisteller1.jpg**  *Bild: SRF mySchool* | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |
| --- |
| **Aufgabe 2: Begriffe** |

|  |
| --- |
| Software |
| Bild: Keystone    Bild: Pixabay |

|  |
| --- |
| Programmiersprache  allgemein |
| Bild: Wikimedia Commons CC, SRF mySchool    Bild: Shutterstock |

|  |
| --- |
| Binärcode |
| Bild: Pixabay |

|  |
| --- |
| Quellcode |
| Bild: SRF mySchool |

|  |
| --- |
| Programmiersprache  Ada |
| Bild: Wikimedia Commons CC |

|  |
| --- |
| Programmiersprache  Scratch |
| Bild: Scratch |

|  |  |
| --- | --- |
| **Aufgabe 3a: Programmiere deinen Roboter** | |
| Der Roboter versteht einfache Befehle. Er macht immer nur genau das, was er versteht. Gleich ist es mit Computern. Computer setzen einfache Befehle um. Damit du etwas programmieren kannst, benötigst du präzise und klare Befehle. | |
|  | |
| Partnerarbeit: Zieht ein Kärtchen aus der Box und legt es an einen gut sichtbaren Ort. Einer von euch ist der Programmierer, der andere ist der Roboter. Der Programmierer gibt dem Roboter einfache Anweisungen und steuert den Roboter zum Kärtchen. Der Roboter darf weder Fehler korrigieren noch selber denken, denn er macht nur genau das, was der Programmierer ihm befiehlt. Sobald der Roboter beim Kärtchen angelangt ist, lest ihr gemeinsam das Kärtchen durch und tauscht euch kurz darüber aus. Zieht ein neues Kärtchen und tauscht die Rollen. | |
|  | |
|  | |
| **Aufgabe 3b** | |
| In der Klasse habt ihr klare und präzise Befehle besprochen und kennengelernt. Sie sind notwendig, damit dein Roboter möglichst schnell zum Ziel gelangt. Im Couvert befinden sich die Befehlskarten. Platziere irgendwo im vorgegebenen Raum einen Gegenstand, den der Roboter für dich nachher holen muss (= Ziel). Bestimme, wo dein Roboter starten wird. Schreibe nun ein Programm für deinen Roboter. Suche dafür passende Befehlskärtchen und leg sie in der richtigen Reihenfolge, von unten nach oben, auf deinem Pult aus (siehe Beispiel). Das nennt man Blockprogrammierung. | |
|  | |
| Beispiel: | |
| |  | | --- | |  | |  | |  | |  | |  | |  | |  | | 🏋 | | Tipp: Der Start und das Ziel sollten zu Beginn nicht zu weit voneinander entfernt sein. |

|  |
| --- |
| **Aufgabe 3c** |
| Such dir einen Roboter, also einen Partner, welcher dein Programm ausführt. Der Roboter fotografiert deine Blockprogrammierung mit dem Tablet und läuft vom Start bis zum Ziel. Das Tablet nimmt er auf die «Reise» mit, damit er weiss, was er machen muss. Stimmt deine Programmierung? Findet dein Roboter den Gegenstand? |
|  |
|  |
| **Aufgabe 3d** |
| Versuche eine erweiterte Blockprogrammierung, indem du die Tipps der Sprechblasen einbaust. |
|  |
| **Wiederholung:**  Mit einem «Mehrfach-Befehl» wie zum Beispiel 6x, lassen sich Befehlskarten wiederholen. So kannst du Karten sparen. |
| **Bug** heisst Ungeziefer. Das Suchen und Beseitigen von Ungeziefer nennt man beim Programmieren «Debuging.»    **Debugging:**  Läuft der Roboter zu weit oder zu kurz? Fehlt eine Drehung? Stoppe den Roboter und korrigiere deinen Fehler. Stimmt es jetzt? |

|  |
| --- |
| **Aufgabe 3f: Befehlskarten** |

|  |  |  |  |
| --- | --- | --- | --- |
| Schritt nach vorn | Schritt nach vorn | Schritt nach vorn | Schritt nach vorn |
| Schritt nach vorn | Schritt nach vorn | Schritt nach vorn | Schritt nach vorn |
| Schritt zurück | Schritt zurück | Schritt zurück | Schritt zurück |
| rechts Drehung 90° | rechts Drehung 90° | rechts Drehung 90° | rechts Drehung 90° |
| links Drehung 90° | links Drehung 90° | links Drehung 90° | links Drehung 90° |
| rechts Drehung 180° | rechts Drehung 180° | rechts Drehung 180° | rechts Drehung 180° |
| links Drehung 180° | links Drehung 180° | links Drehung 180° | links Drehung 180° |

|  |  |  |  |
| --- | --- | --- | --- |
| **0** | **1** | **2** | **3** |
| **4** | **5** | **6** | **7** |
| **8** | **9** | -Mal | -Mal |
| 🏋  Gegenstand hochheben | 🏋  Gegenstand hochheben |  |  |

|  |
| --- |
| **Aufgabe 4: Scratch** |
|  |
| Gehe auf die Hauptseite von Scratch: <https://scratch.mit.edu/>. Klicke auf «Entwickeln». |
|  |
| Bei der Weltkugel kann man die Sprache einstellen. Die Bausteine erscheinen dann in Deutsch. Die Tutorials sind nur auf Englisch und nicht in deutscher Sprache verfügbar. |
|  |
|  |
|  |
| Unter «**Bewegung**» und «**Steuerung**» lassen sich die Bausteine finden, welche zu den Kärtchen aus Aufgabe 3b passen. Schau mit der Klasse die Befehle der Skripte an und vergleiche sie mit den Befehlskarten. Welche Befehle passen zusammen? |

|  |  |  |
| --- | --- | --- |
|  | **Bereich** | **Programmierbausteine (Beispiele)** |
| **Bewegung** | gehe 10er Schritt  drehe dich um 15 Grad, … |
| **Aussehen** | wechsle zu Kostüm  sage Hallo! |
| **Klang** | spiele Klang  ändere Lautstärke um 10 |
| **Ereignisse** | Wenn Taste «s» gerückt wird  Wenn diese Figur angeklickt wird |
| **Steuerung** | wiederhole 10 mal  warte 10 Sekunden … |
| **Fühlen** | Farbe ⚫ berührt Farbe ⚫ |
| **Operatoren** | 2 + 4  Zufallszahl von 1 bis 10 |
| **Variablen** | verstecke Variable  ändere Variable um 1 |
|  | Bei diesem Symbol (unten links) kann man Erweiterungen auswählen (z.B. Befehle für Klänge) | |
|  | spiele Schlaginstrument Tamburin, setze Instrument auf Klavier, spiele Ton... | |

|  |
| --- |
| **Aufgabe 5: Scratch kurz erklärt** |

•

**Was kann man mit Scratch**

**programmieren?**

* Spiele (Games)
* Animationen
* Geschichten (Storys)
* Kunst
* Musik
* Instrumente
* Charakteren
* Simulationen
* ...

•

**FAQ von Scratch**

Schnelle Antworten auf häufig gestellte Fragen rund um Scratch sind unter diesem Link abrufbar: ⮊ [**FAQ**](https://scratch.mit.edu/info/faq#schools)

Die Fragen sind auf Deutsch und die Antworten oft in englischer Sprache verfasst.

•

**Braucht es ein Login?**

Als Lehrperson kannst du ein Scratch-Benutzerkonto für Lehrer anfordern. So hast du die Möglichkeit vereinfacht Benutzerkonten für eine Gruppe von Lernenden zu erstellen. Siehe dazu die detaillierten Infos auf Scratch [FAQs zu Lehrer-Benutzerkonten](https://scratch.mit.edu/educators/faq). Mit diesem Link kannst du direkt ein [Lehrer-Benutzerkonto erstellen](https://scratch.mit.edu/educators/register).

•

**Was sind die Systemvoraussetzungen für Scratch?**

Scratch funktioniert mit den bekannten Browsern wie Edge, Firefox, Safari oder Chrome (auf dem Tablet: Mobile Chrome oder Mobile Safari). **Der Internet Explorer funktioniert nicht**. Um ein Projekt mit Scratch zu entwickeln, braucht man einen Laptop oder ein Tablet. Auf einem Smartphone kann man Projekte ansehen, jedoch keine kreieren.

•

**Für welches Alter ist Scratch geeignet?**

Scratch ist für 8-16-jährige.

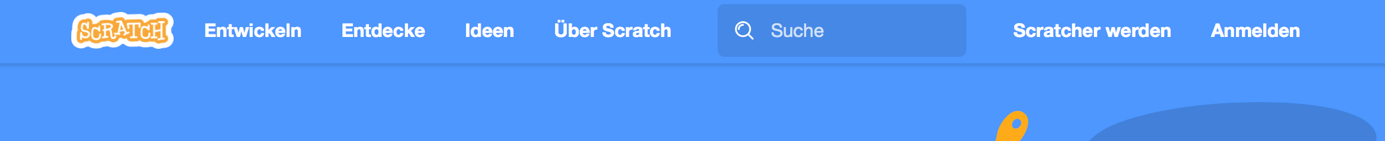
[Scratch JR](https://www.scratchjr.org) für 5-7-jährige.

•

**Was ist Scratch?**

Scratch ist eine Programmiersprache und eine Online-Gemeinschaft, in der Kinder interaktive Medien wie Geschichten, Spiele und Animationen programmieren und weltweit mit anderen teilen können. Indem Kinder ihre Ideen mit Scratch umsetzen, lernen sie kreativ zu denken, mit anderen zusammenzuarbeiten und systematisch vorzugehen.

|  |
| --- |
| **Hauptregister kurz erklärt** |



**➊**

**➋**

**➌**

**➍**

**➎**

**➏**

**➐➐**

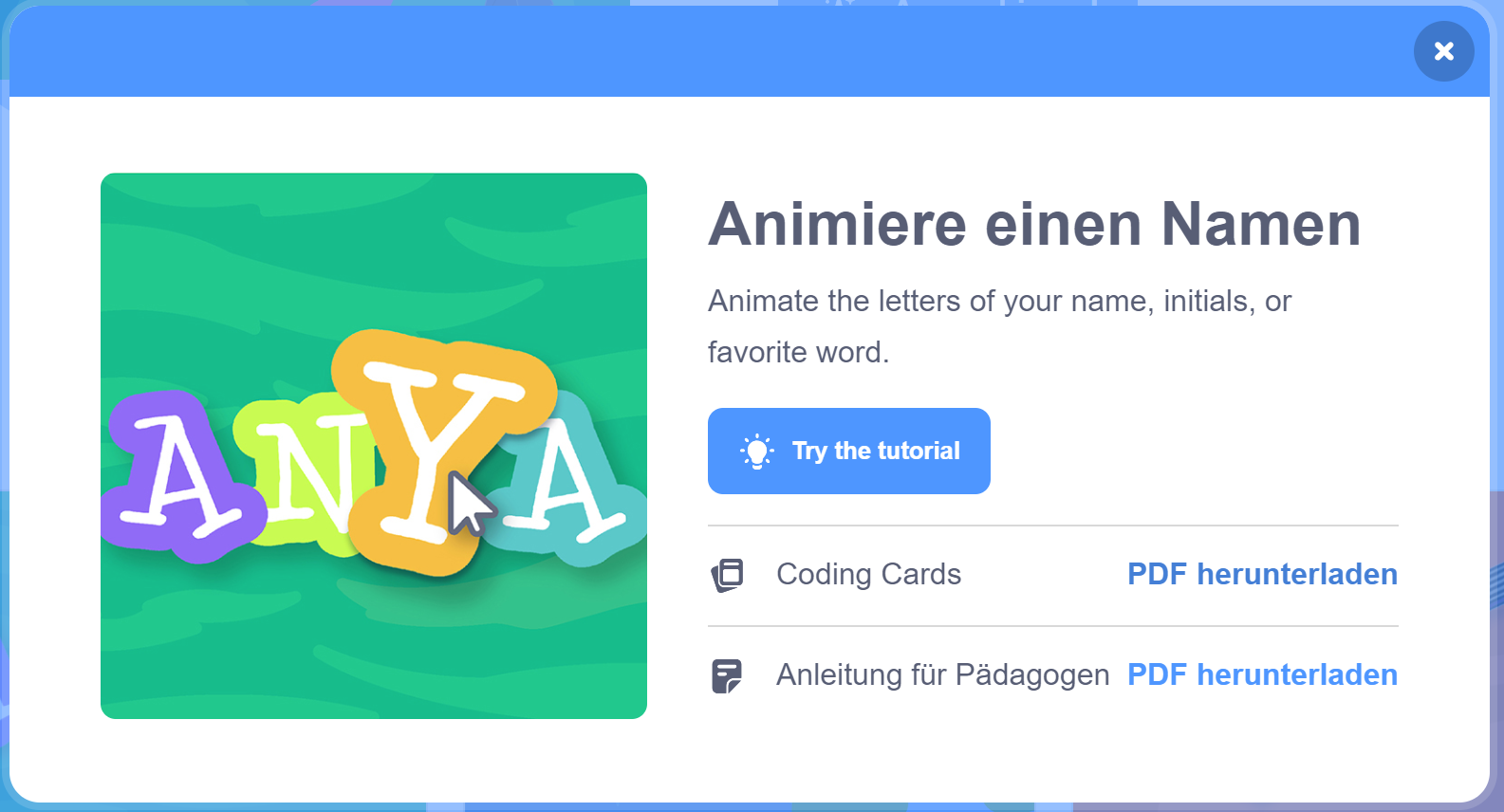
➊ Hier gelangt man zur Startseite von Scratch.

➋ Hier kann man ein Scratch-Projekt entwickeln (ohne Anmeldung). ⮊ Im Hauptregister «Entwickeln» kann man bei der Weltkugel die Sprache Deutsch wählen.

➌ Hier kann man Projekte von anderen Personen anschauen.

➍ Hier findet man verschiedene Ideen für Projekte. Jede Projektidee beinhaltet:

* 1 Tutorial
* 1x Karten für SchülerIn (Anleitung)
* 1x Karten für Lehrperson (Methodik)

Ein Beispiel:

➎ Hier sind hilfreiche Informationen für Eltern und Informationen für [Erzieher](https://scratch.mit.edu/educators/) nachzulesen.

➏ Hier kann man ein Konto erstellen. Als LP kann man ein spezielles Scratch-Benutzerkonto anfordern. So hat man die Möglichkeit, vereinfacht Benutzerkonten für eine Gruppe von Lernenden zu erstellen.

➐ Hier meldet man sich mit einem Benutzerkonto an.

**Direkte Links zu den Scratch-Projekten im PDF-Format:**

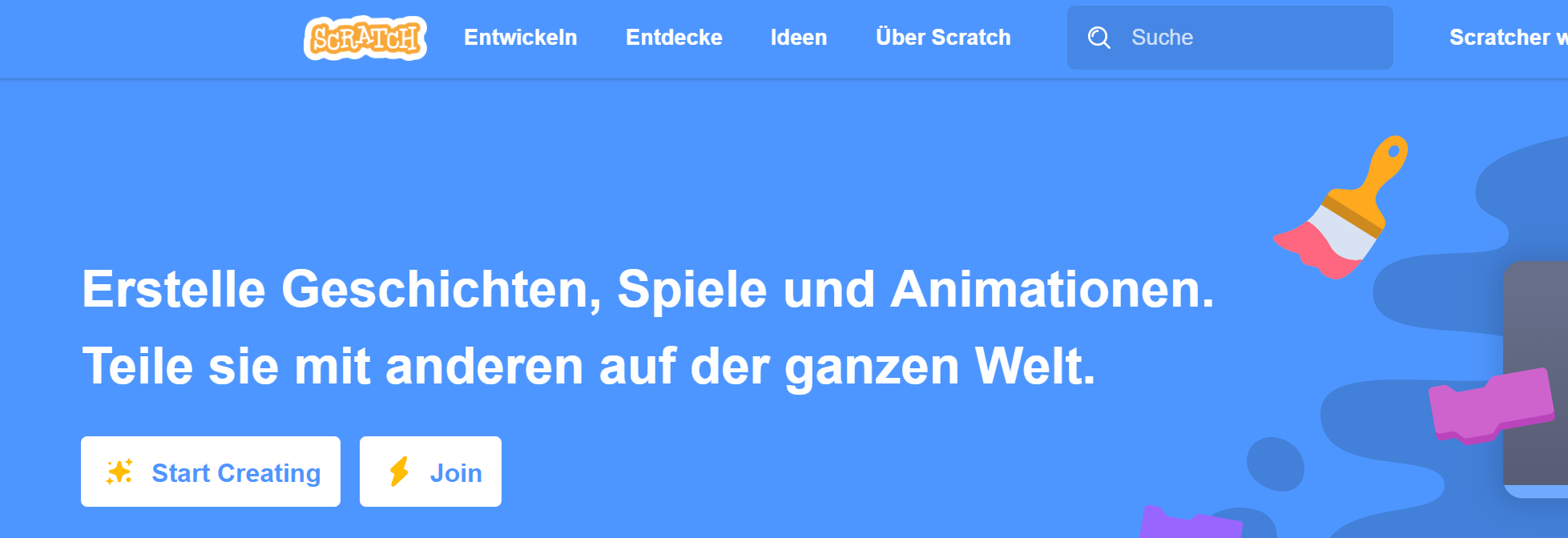
|  |  |  |
| --- | --- | --- |
| **Link zu allen Karten in einem  PDF-Format** |  | **Link zum Projektbeispiel**  **«Animate your Name»** |
| [Karten für Lehrperson](https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf)  ⮊ zur Vorbereitung von Scratch-Sequenzen |  | [Animiere einen Namen - LP](https://resources.scratch.mit.edu/www/guides/en/NameGuide.pdf)  ⮊ Anleitung für Pädagoge |
| [Karten für SchülerIn](https://resources.scratch.mit.edu/www/cards/en/scratch-cards-all.pdf)  ⮊ Anleitung für ihr Scratch-Projekt |  | [Animiere einen Namen - Coding Cards](https://resources.scratch.mit.edu/www/cards/en/name-cards.pdf)  ⮊ Anleitung für SuS |
| [Tutorials für SchülerIn](https://scratch.mit.edu/projects/editor/?tutorial=all)  ⮊ Anleitung für ihr Scratch-Projekt |  | [Animiere einen Namen – Tutorial](https://scratch.mit.edu/projects/editor/?tutorial=all)  ⮊ Tutorial für SuS |

**Scratch im Unterricht – möglicher Ablauf anhand des Beispiels «Animate your Name»:**

➊ Die LP erklärt kurz, was Scratch ist und welche Möglichkeiten das Programm bietet. (⮊ siehe erste Seite dieses Arbeitsblattes «Scratch kurz erklärt»).

➋ Die LP zeigt ein [Beispiel von anderen Scratch-Usern](https://scratch.mit.edu/explore/projects/all) (⮊ Hauptregister «Entdecke»), damit sich die SuS darunter etwas vorstellen können.

➌ Die SuS probieren frei aus. Sie klicken auf «Start Creating». Dieser Button befindet sich gleich auf der [Startseite](https://scratch.mit.edu/) von Scratch.



➍ Im Plenum schaut die Klasse das [Tutorial «Animate your name»](https://scratch.mit.edu/projects/editor/?tutorial=all). Die SuS programmieren anschliessend mit der [Anleitungs-Karte «Animate your name»](https://resources.scratch.mit.edu/www/cards/en/name-cards.pdf) und mit Hilfe des Tutorials ihren Namen auf kreative Art und Weise. Die LP hat eine eigene [Karte mit methodisch didaktischen Tipps](https://resources.scratch.mit.edu/www/guides/en/NameGuide.pdf).

➎ Schnelle SuS wählen eine neue Anleitungs-Karte (z.B. «Mach ein Fangspiel» oder «Erfinde eine Geschichte»). Die Lernenden können in ihrem Tempo arbeiten. Individualisieren ist gut möglich!