

pädagogische hochschule schwyz

# Scratch Projektideen



Michael Hielscher  
Beat Döbeli Honegger

11. Juli 2017



## Worum geht es?

Scratch ist eine kostenlose Programmierumgebung für Kinder. Scratch ist leicht zu erlernen und eignet sich sowohl für einfache Animationen als auch für komplexe Spiele und Simulationen. Du kannst mit wenigen Klicks eigene Bilder und Töne erstellen und damit deine ganz persönlichen Projekte entwickeln. Scratch lässt sich auch mit echten Robotern und Sensoren kombinieren. Kurz gesagt: Scratch macht einfach Spass und bietet unendlich viele Möglichkeiten! Alles was du brauchst ist ein Webbrowser.



# SCRATCH

## Wie funktioniert es?

### 1. Bei Scratch Anmelden

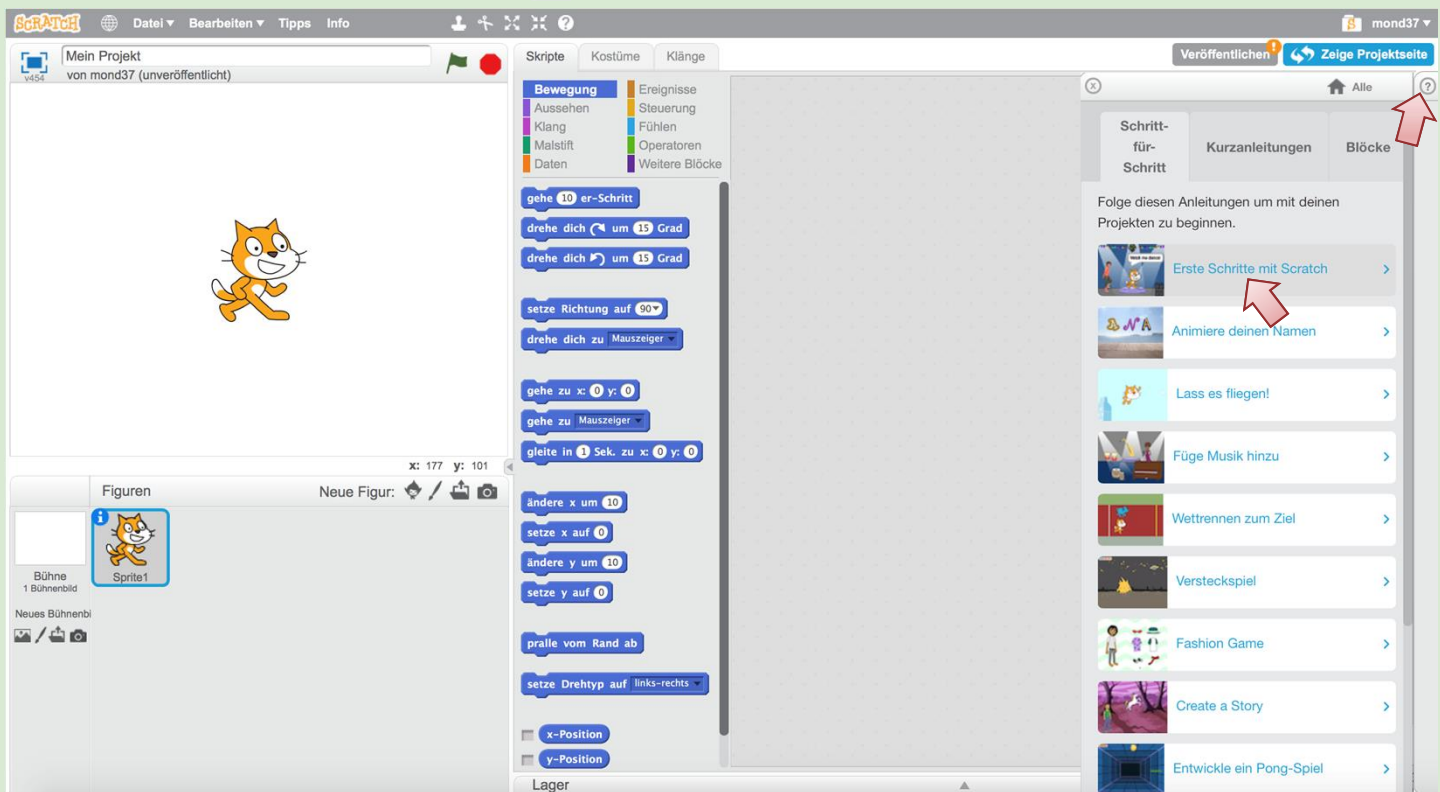
Um mit Scratch zu arbeiten, benötigst du einen Webbrowser wie Chrome oder FireFox. Gehe auf <http://scratch.mit.edu>. Es lohnt sich ein Scratch-Benutzerkonto anzulegen. Klicke dazu auf „Scratcher werden“ oben rechts.



Du benötigst die folgenden Angaben: Einen Benutzernamen (wähle etwas phantasievolles), ein Passwort, dein Geburtsdatum und eine Emailadresse.

### 2. Ein Scratch-Projekt erstellen

Klicke auf „Entwickeln“ um ein neues Scratch-Projekt zu beginnen. Wähle wenn du an einem Projekt weiterarbeiten möchtest. Klappe rechts das Hilfemenü auf ( ? ) und arbeite die Anleitung „Erste Schritte mit Scratch“ durch.



## Beispiele

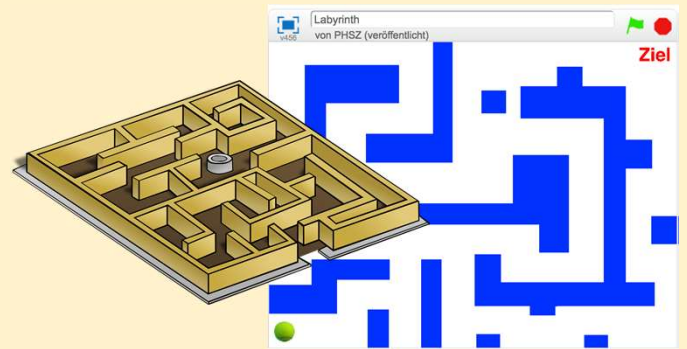
In diesem Kästchen findest du jeweils Beispiele zu den Projektideen. Schau sie dir am Besten erst an, wenn du mit deinem Projekt fertig bist, oder du nicht weiter kommst.

## Wie weiter?

In diesem Kästchen stehen jeweils Ideen für die Weiterarbeit und schwierigere Herausforderungen. Teilweise findest du Lösungen dazu in den Beispielprojekten im anderen Kasten.

## Worum geht es?

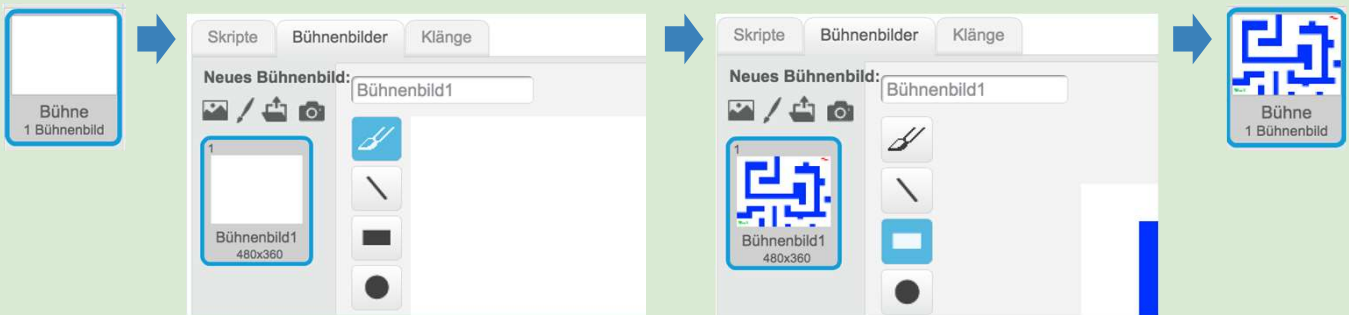
Mit Scratch lassen sich ganz einfach Geschicklichkeitsspiele erstellen, bei denen ein Objekt mit der Maus bewegt werden muss. Bei diesem Spiel geht es darum einen Tennisball ohne anzustossen durch ein Labyrinth zu navigieren. Du entscheidest, wie einfach oder schwierig das wird 😊  
Probiert gegenseitig eure erstellten Labyrinth aus.



## Wie funktioniert es?

### 1. Ein Labyrinth mit Scratch zeichnen

Klicke links unten auf „Bühne“ und anschliessend auf „Bühnenbilder“ um das das Labyrinth zu zeichnen.  
Wähle eine Farbe aus und verwende das gefüllte Rechteck-Werkzeug, um die Teile deines Labyrinths zu zeichnen.



Du kannst dein Labyrinth jederzeit überarbeiten und mehr oder weniger viele Hindernisse einbauen. Verwende jedoch immer die gleiche Farbe für alle Wände.

### 2. Spielfigur und Programmierung

Klicke auf **Neue Figur:** und suche den Tennisball in der Bibliothek heraus. Die Katze benötigen wir nicht, du kannst sie mit einem Rechtsklick „Löschen“. Bewege den Tennisball auf der Bühne an die Startposition deines Labyrinths (z. B. eine Ecke des Spielfelds). Verwende „Verkleinern“ sollte der Ball noch zu gross sein.

Erstelle für den Tennisball das rechte Skript. Die Blöcke findest du jeweils im gleichfarbigen Bereich der Blockplatte. Wenn du fertig bist, klicke auf **Start**, um das Spiel zu starten. Der Ball wartet, bis er mit der Maus berührt wird (**warte bis** **wird Mauszeiger berührt?**) und wird anschliessend der Maus folgen (**gehe zu** **Mauszeiger**), bis ausversehen eine Wand berührt wird (**wiederhole bis** **wird Farbe berührt?**).



## Beispiele

Labyrinth-Spiel

[scratch.mit.edu/projects/154416386](https://scratch.mit.edu/projects/154416386)

Whac-A-Mole Spiel

[scratch.mit.edu/projects/151083755](https://scratch.mit.edu/projects/151083755)

## Wie weiter?

- Kannst du ein Zielfeld hinzufügen, welches bei Berührung mit dem Tennisball einen Soundeffekt abspielt?
- Kannst du eine Stoppuhr einbauen, wie lange man gebraucht hat?
- Kannst du Figuren (in Wandfarbe) als bewegliche Hindernisse einbauen, die sich selbstständig hin und her bewegen?



## Worum geht es?

Du bist der Regisseur! Wer kommt in einer Szene auf die Bühne und wer sagt was? Mit Scratch kannst du kleine Filme oder Animationen erstellen. Die Figuren und Hintergründe kannst du programmieren, sodass sie sich nach deinen Ideen bewegen und sprechen. Vom kurzen Witz, bis zum kompletten Theaterstück ist alles möglich.



## Wie funktioniert es?

### 1. Schreib deine eigene Geschichte

Um eine gute Geschichte zu erzählen benötigst du zunächst ein gutes Regiebuch. Darin sind die Szenen und die verschiedenen Rollen (Figuren) und ihre Handlungen, Positionen auf der Bühne und Texte festgehalten. Beispiel:

#### Szene 1: In der Wüste

Der Ritter erscheint von links auf der Bühne und läuft hin und her. Der Ritter sagt: „Ich bin müde und muss mich ausruhen.“ Der Ritter setzt sich unter einen Kaktus, schläft ein und schnarcht laut vor sich hin.

Der Löwe erscheint von rechts auf der Bühne und läuft zur Bühnenmitte. Der Löwe bemerkt den Ritter, seufzt und sagt „Mist, schon wieder Konservenfutter!“. Der Löwe läuft schnell zum Ritter und frisst ihn auf (Ritter verschwindet).

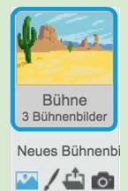
#### Szene 2: Im Schloss

Die Prinzessin erscheint auf der Bühne und sagt: „Wo sind nur alle meine Ritter hin? Ich sollte nachschauen gehen.“

Die Prinzessin läuft rechts aus dem Bild und verschwindet von der Bühne.

### 2. Bilder und Töne zusammenstellen

Nachdem du im Regiebuch die verschiedenen Figuren und Szenen beschrieben hast, kannst du diese in Scratch anlegen. Wähle aus der Bibliothek passende Figuren: und Bühnenbilder aus: . Ist kein passendes Bild dabei, kannst du auch eigene Figuren und Hintergründe zeichnen () , oder freie Bilder aus dem Internet (z.B. von [www.pixabay.com](http://www.pixabay.com)) herunterladen und in Scratch hochladen (). Unter „Klänge“ kannst du passende Geräusche für jede Figur auswählen () oder selbst aufnehmen ().



### 3. Übernimm die Regie

Wähle die „Bühne“ aus (blauer Rahmen). Der Regisseur des Theaterstücks dirigiert unsichtbar im Hintergrund und ruft allen Figuren zu, welche Szene als nächstes an der Reihe ist. Zudem wählt er das passende Bühnenbild aus. Die Figuren hören auf seine Anweisungen.

#### Regie-Script (Bühne):



Für jede Szene musst du für alle Figuren festlegen, ob und wann sie auf die Bühne kommen und gehen. In Szene 1 spielt die Prinzessin nicht mit (Regiebuch).

#### Beispiel Scripte von Ritter, Löwe und Prinzessin für Szene 1: In der Wüste



## Beispiele

Einen Witz erzählen

[scratch.mit.edu/projects/151205820](http://scratch.mit.edu/projects/151205820)

Der Ritter und der Löwe

[scratch.mit.edu/projects/151409075](http://scratch.mit.edu/projects/151409075)

## Wie weiter?

- Erstelle bei allen Figuren die Scripte für Szene 2 usw.
- Füge in der Bühne passende Hintergrundmusik zu jeder Szene hinzu.
- Kannst du die Geschichte „interaktiv“ gestalten, sodass der Zuschauer zum Beispiel Dinge anklicken oder sogar selbst steuern kann?

## Worum geht es?

Mit Scratch lassen sich sehr leicht eigene Musikinstrumente oder Soundeffekte erstellen. Scratch hat bereits viele Töne und Sounds eingebaut und über die Aufnahmefunktion lassen sich auch ganz einfach eigene Geräusche hinzufügen.

In diesem Beispiel baust du ein Klavier zum Anklicken und eine Harfe zum Spielen mit den Händen mit Hilfe einer Webcam.



## Wie funktioniert es?

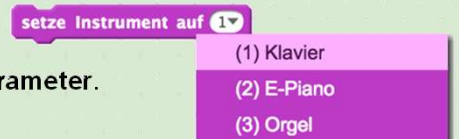
### 1. Ein Klavier bauen

Erstelle ein neues Scratch-Projekt und erstelle eine **Neue Figur:** . Zeichne mit schwarz einen Rahmen und fülle das innere mit weiss für die erste Klaviertaste. Erstelle das folgende Script für die Taste:



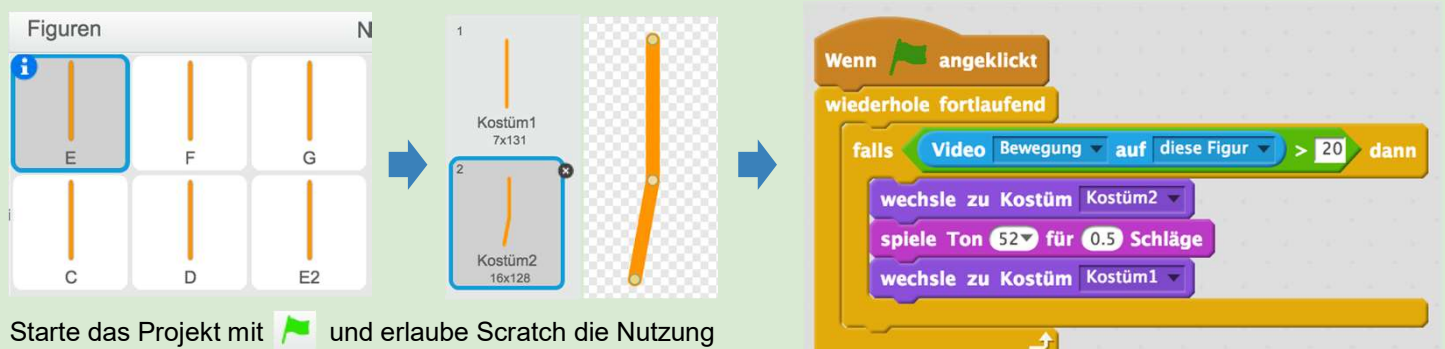
Klicke mit der rechten Maustaste auf die Klaviertaste und wähle „duplizieren“. Für die neue Taste musst du die Einstellungen im Script anpassen. Klicke auf das kleine ▼ neben der 60 und wähle in beiden Blöcken D (62) aus. Dupliziere erneut die Taste und wähle E (64), die nächste weisse Taste, aus.

Ordne die Tasten nebeneinander an und wiederhole den Vorgang, bis du ein Klavier von C (60) bis C (72) hast. Trage bei **Wenn Taste a gedrückt** jeweils einen Buchstaben a, s, d, f, g, h, j bis k ein. Probiere dein Klavier aus und ergänze auch die schwarzen Tasten (Tastatur w, e, t, z und u). Jede Figur in Scratch hat ihr eigenes Instrument. Mit „setze Instrument auf ...“ kannst du jeder Figur ein anderes Instrument zuteilen. Solche Einstellungen an Blöcken nennt man auch **Parameter**.



### 2. Ein Webcam-Harfe bauen

Für dieses Projekt benötigst du einen Notebook oder PC mit Webcam. Scratch kann Bewegungen im Webcam-Bild erkennen. Erstelle eine **Neue Figur:** und zeichne im Vektormodus eine gerade Linie. Dupliziere das Kostüm per Rechtsklick und verändere die Linie leicht, damit diese wie „gezupft“ aussieht. Erstelle ein Script wie unten gezeigt und dupliziere die Figur mehrfach. Ordne die Seiten auf der Bühne an. Stelle für jede Harfenseite den passenden Ton ein.



Starte das Projekt mit und erlaube Scratch die Nutzung deiner Webcam. Bewege deine Hände an die Seiten der Harfe.

## Beispiele

Klavier

[scratch.mit.edu/projects/151957886](https://scratch.mit.edu/projects/151957886)

Webcam Harfe

[scratch.mit.edu/projects/151145571](https://scratch.mit.edu/projects/151145571)

Webcam Schlagzeug

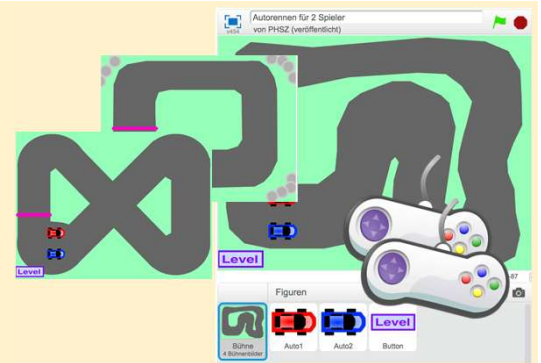
[scratch.mit.edu/projects/151148401](https://scratch.mit.edu/projects/151148401)

## Wie weiter?

- Kannst du statt den eingebauten Töne für jede Taste ein eigenes Geräusch oder Laut mit dem Mikrophone () aufnehmen?
- Auf [www.auditorix.de](http://www.auditorix.de) findest du in der Geräusche-Box Soundeffekte zum herunterladen, die du für dein Scratch Projekt verwenden kannst. Kannst du eine Geschichte mit Hilfe von Soundeffekten erzählen?

## Worum geht es?

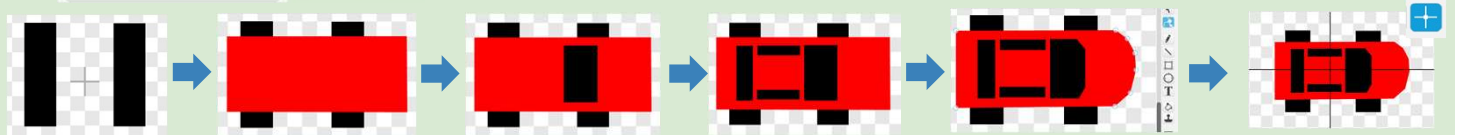
In vielen Computerspielen kann man mit zwei oder mehr Spielern zusammen spielen. In der einfachsten Variante sitzen alle Spieler gemeinsam vor einem Bildschirm und jede Spielfigur lässt sich einzeln steuern (z.B. durch bestimmte Tasten auf der Tastatur).



## Wie funktioniert es?

### 1. Rennstrecke und erstes Auto erstellen

Zeichne ein Bühnenbild mit einer Rennstrecke. Verwende eine Farbe als Begrenzung der Fahrbahn (z.B. Grasgrün). Erstelle eine **Neue Figur:** für das erste Auto. Zeichne das Auto aus der Vogelperspektive als Vektorgrafik (mit **In Vektorgrafik umwandeln**). Verwende Zoom von 800%. Wähle den Mittelpunkt des Auto mit **+** (oben rechts) aus.



Um uns Arbeit zu sparen, erstellen wir das zweite Auto erst am Schluss durch duplizieren / umfärben des ersten Autos. Die meisten Spiele verwenden eine sogenannte *Spiele Schleife*. Solange das Spiel läuft werden die Blöcke innerhalb der Schleife immer wieder abgearbeitet. In Scratch verwendet man dafür den „wiederhole fortlaufend“-Block und prüft auf bestimmte Bedingungen („falls ... dann“). Baue das folgende Programm für Auto1 nach:

Die Zahlen **5** und **-2** geben an, wie schnell sich das Auto bewegt und dreht. Experimentiere mit diesen Werten.

### 2. Auf der Strecke bleiben

Klick auf für eine Probefahrt mit den Pfeiltasten. Ein Spiel braucht eine Herausforderung und beim Autorennen geht es darum, auf der Strecke zu bleiben. Sobald das Auto die grüne Rasenfläche berührt, hat der Spieler verloren und das Auto soll wieder an den Start gestellt werden. Füge einen „falls ... dann“ Block wie im Bild unten hinzu. Um die Farbe zu wählen klicke auf das Rechteck und danach in der Bühne auf eine Rasenfläche.



### 3. Spieler Zwei

Wähle das erste Auto unter „Figuren“ aus und klicke mit der rechten Maustaste darauf. Wähle „duplizieren“ um eine Kopie zu erstellen. Bearbeite das Kostüm und lackiere das zweite Auto in einer anderen Farbe. Wähle andere Tasten in den hellblauen Blöcken aus (z.B. W,A,S und D). Starte das Spiel mit neu und teste die Bewegung des zweiten Autos.

## Beispiele

Autorennen für zwei Spieler  
[scratch.mit.edu/projects/150489552](https://scratch.mit.edu/projects/150489552)

Stern ausweichen  
[scratch.mit.edu/projects/135827798](https://scratch.mit.edu/projects/135827798)

## Wie weiter?

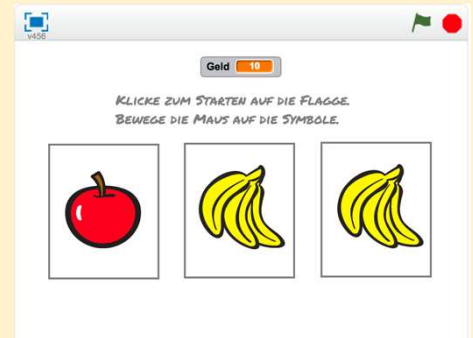
- Kannst du eine Ziellinie einfügen die beim Überfahren einen passenden Soundeffekt auslöst?
- Kannst du einen Rundenzähler für jeden Spieler einbauen?
- Kannst du verschiedene, auswählbare Rennstrecken einbauen?
- Kannst du einen Computergegner erstellen, der selbstständig fährt?



## Worum geht es?

Scratch eignet sich hervorragend, um kleine Spiele zu entwickeln. Viele Spiele verwenden den „Zufall“, zum Beispiel in Form eines Würfels, damit das Spiel nicht vorhersehbar wird und spannend bleibt. Scratch bietet einen Befehlsblock für Zufallszahlen an, um zum Beispiel eine Zahl zwischen 1 und 6 zu würfeln.

Wir bauen in diesem Beispiel das Spiel Slot-Machine, bei dem der „Zufall“ eine wichtige Rolle spielt.



## Wie funktioniert es?

### 1. Slots und Hintergrundbild erstellen

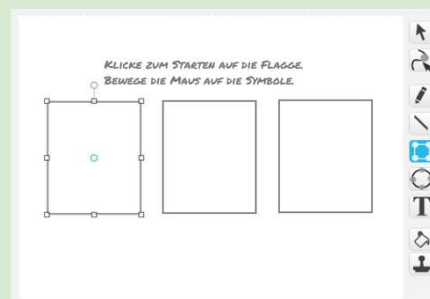
Erstelle ein neues Scratch-Projekt. Wähle **Neue Figur:** aus und suche dir aus der Bibliothek eine erste Figur heraus - zum Beispiel den Apfel. Benenne die Figur „Slot1“. Wechsel auf **Kostüme** und lade einige weitere Bilder dazu. Passe die Grösse der einzelnen Bilder so an, dass sie alle etwa gleich gross sind. Erstelle ein neues Bühnenbild und zeichne im Vektor-Modus drei gleich grosse Rechtecke:

### 2. Slots programmieren

Erstelle zwei neue Variablen „Geld“ und „angehalten“ unter **Daten**. Erstelle das folgende Skript für die Figur Slot1 und probiere das Spiel mit einem Klick auf aus:



Bei „Zufallszahl“ musst du die Anzahl der Kostüme eintragen, die deine Figur verwendet. Damit wird ein zufälliges Startkostüm gewählt.



Klicke rechts auf Slot1 und wähle „Duplizieren“ aus, um Slot2 und anschliessend Slot3 zu erstellen.



### 3. Geld ausgeben und gewinnen

Jedes Spiel soll etwas Geld kosten – mit Scratch geht das mit „ändere Geld um -1“. Immer wenn ein Slot gestoppt wird, erhöht sich die Variable „angehalten“ um 1. Wir warten bis gehalten = 3 ist und prüfen dann, ob die Symbole der Slots gleich sind. Erstelle das folgende Skript für die Bühne:



Bau das Skript so aus, dass die Spieler auch etwas Geld für zwei gleiche Symbole bekommen.

## Beispiele

Slot Machine

[scratch.mit.edu/projects/167744761](https://scratch.mit.edu/projects/167744761)

## Wie weiter?

- Kannst du die Slots so anpassen, dass sie beim Berühren mit der Maus noch 2-3 Bilder weiterlaufen und dabei langsamer werden?
- Kannst du die Auswertung so umbauen, dass unterschiedliche Symbole unterschiedlich grosse Gewinne ergeben? (z.B. 2 Herzen sind besser als 2 Äpfel)

## Worum geht es?

In Scratch Programmen kann eine Figur eine Frage stellen, auf die man eine Antwort eingeben muss. Mit dieser Funktion können zum Beispiel Ratespiele erstellt werden. Kannst du ein spannendes Quiz zu deinem Lieblingsthema entwickeln?



## Wie funktioniert es?

### 1. Hintergrund und Figuren wählen und Quiz programmieren.

Erstelle ein neues Scratch-Projekt. Wähle **Neue Figur:** aus und suche dir aus der Bibliothek eine Figur heraus - zum Beispiel den Pinguin. Wähle zu deiner Figur ein passendes Bühnenbild aus: Wähle deine Figur aus (blauer Rahmen) und erstelle ein Script für die erste Quizfrage wie in folgendem Beispiel gezeigt:



Wenn in Scratch ein hellblauer frage-Block verwendet wird, legt Scratch den eingegebene Text in die **Antwort** Variable. Mit lassen sich zwei Werte miteinander vergleichen. Es wird also geprüft, ob „Biene“ als Antwort eingegeben wurde: . Du kannst mit dem ODER-Block auch mehrere Antworten als richtig werten:



Hinweis: Schiebe in Scratch immer die linke obere Ecke eines Bausteins in das freie Feld. Nur wenn der Baustein einrastet, wird das Script richtig funktionieren. Das Feld leuchtet dann auf:

Um eine weitere Spielfrage hinzuzufügen, klicke mit der rechten Maustaste auf den hellblauen frage-Block und „Dupliziere“ ihn. Alle Folgeblöcke werden mit kopiert und du musst nur noch die Frage- und Antwort-Felder neu ausfüllen.

### 3. Ein eigener Scratch-Block

Wenn du in Scratch mehrere Blöcke immer wieder verwendest, kannst du unter **Weitere Blöcke** auch einen eigenen Block mit **Neuer Block** erstellen. Du kannst selbst festlegen, welchen Namen und welche Einstellungen dein neuer Block haben soll. Für das Quiz können wir einen Block **Eine Frage stellen** mit zwei Textfeldern für Frage und Antwort anlegen (2x anklicken). Schau dir das folgende Beispiel genau an und baue es nach. Beim Programmieren spricht man bei eigenen Blöcken auch von „Unterprogrammen“ und die Einstellungen bezeichnet man als „Parameter“, die an das Unterprogramm übergeben werden. Programmierer teilen ihre Programme häufig in kleine Unterprogramme auf, da sich diese einfacher wiederverwenden lassen und bei Änderungen nur jeweils an einer Stelle angepasst werden muss.



## Beispiele

Penguin Quiz

[scratch.mit.edu/projects/151391118](https://scratch.mit.edu/projects/151391118)

Mathe Quiz

[scratch.mit.edu/projects/151955640](https://scratch.mit.edu/projects/151955640)

## Wie weiter?

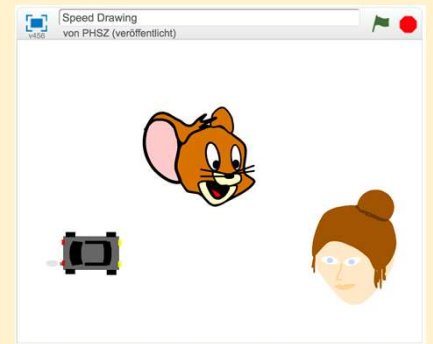
- Kannst du einen Punktezähler einbauen, der immer um 1 geändert wird, wenn eine Frage richtig beantwortet wurde?
- Kannst du im „Unterprogramm“ einbauen, dass bei einer falschen Antwort die richtige Lösung angezeigt wird?



## Worum geht es?

Hast du schon einmal versucht deine Lieblingszeichentrickfigur selbst zu zeichnen? Auf YouTube findest du ganz viele Anleitungen und Erklärvideos, wie man Figuren zeichnen kann. In Zeitrafferaufnahmen kann man verfolgen, wie die tollsten Zeichnungen Schritt für Schritt entstehen.

Mit Scratch lassen sich solche Zeitrafferaufnahmen ganz einfach selbst erstellen und dabei einiges über das Zeichnen am Computer lernen.



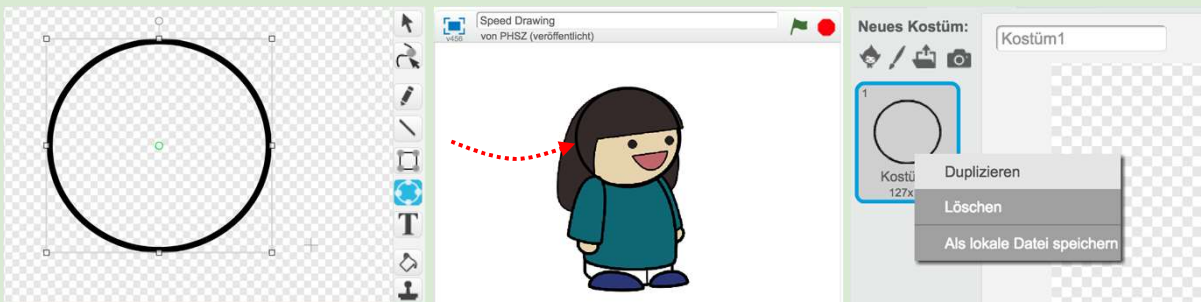
## Wie funktioniert es?

### 1. Zeichentrickfigur als Hintergrund laden.

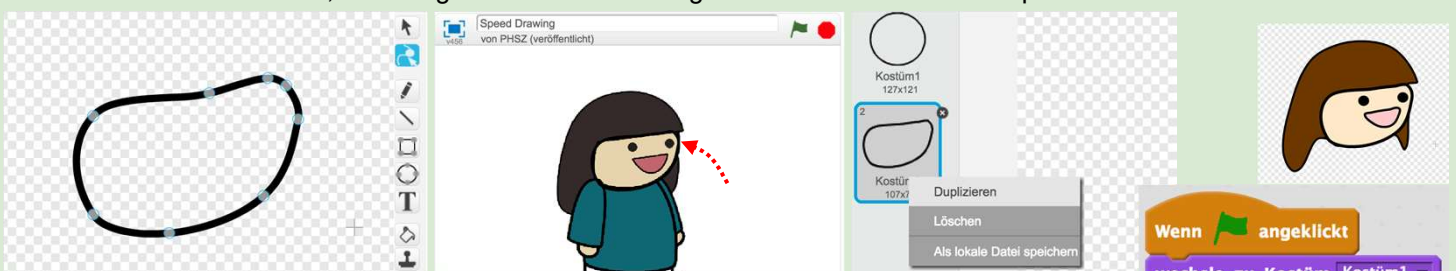
Suche auf einer Seite wie [pixabay.com](https://pixabay.com) nach einer Figur, die du gern zeichnen möchtest. Wähle für den ersten Versuch eine einfache Figur mit grossen, einfarbigen Flächen aus. Lade das Bild mit „Bild speichern unter...“ im Browser herunter. Lade das Bild als Bühnenbild in Scratch hoch: Wechsele in den Vektor-Modus und verkleinere das Bild etwas, so dass es gut in der Bühnenmitte passt.

### 2. Figur schrittweise nachzeichnen

Erstelle ein neues Scratch Projekt und klicke auf **Neue Figur:** , um eine neue Figur zu zeichnen. Wechsele in den Vektor-Modus und zeichne einen ersten Kreis für den Kopf. Schiebe den Kreis mit dem Pfeilwerkzeug so, dass dieser genau auf den Kopf des Hintergrundbildes liegt. Ziehe an den Ecken des Kreises, bis die Form in etwa übereinstimmt.



Klicke das Kostüm mit der rechten Maustaste an und „Dupliziere“ es. Verwende nun das Werkzeug im neuen Kostüm und Bearbeite den Kreis so, dass er genau mit dem Hintergrundbild übereinstimmt. Dupliziere nun Kostüm2.



Füge nun einen weiteren Kreis zum Beispiel für die Augen oder Mund hinzu und verwende . Dupliziere nach jedem Bearbeitungsschritt das Kostüm. Verwende das Füllwerkzeug, um Flächen mit Farben zu füllen, sobald du alle Umrisse eingezeichnet hast. Erstelle das rechts gezeigte Skript, um die Entstehung deines Bildes als Film abzuspielen.

## Beispiele

Speed Drawing Pokemons  
[scratch.mit.edu/projects/166022680](https://scratch.mit.edu/projects/166022680)

Speed Drawing Lisa Simpson  
[scratch.mit.edu/projects/154852469](https://scratch.mit.edu/projects/154852469)

## Wie weiter?

- Gib auf [scratch.mit.edu](https://scratch.mit.edu) „speed drawing“ ins Suchfeld ein und schau dir weitere Beispiele an.
- Verwende eine schwierigere Figur als Vorlage oder versuche ganz ohne Vorlage eine Figur zu zeichnen.

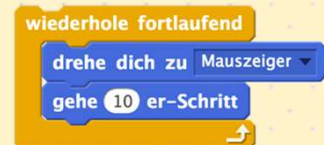
## Sequenz (Abfolge)

Die Reihenfolge der Befehle in einem Computerprogramme spielt eine wichtige Rolle. Diese muss systematisch geplant und durchdacht werden.



## Wiederholung (Schleife)

Befehle lassen sich endlos (bis zum beenden des Programms) oder solange bis eine bestimmten Bedingung eintritt wiederholen.



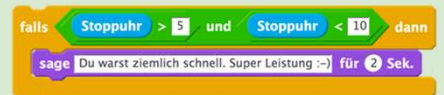
## Entscheidung (Bedingung)

In Abhängigkeit einer Bedingung kann ein Programm entscheiden, ob eine Befehlsfolge abgearbeitet wird oder nicht. Bedingungen sind immer *wahr* oder *nicht wahr*.



## Aussagelogik (boolesche Operatoren)

Bedingungen lassen sich mit den Operatoren UND, ODER und NICHT verknüpfen und damit neue Bedingungen bilden.



## Variable

Eine Variable kann man sich als eine beschriftete Schachtel vorstellen. In der Schachtel wird eine Information (eine Zahl, ein Text usw.) abgelegt und kann zu einem späteren Zeitpunkt wieder gelesen werden.



## Listen (Arrays)

Eine Liste kann man sich als Regal voller Schachteln (Variablen) vorstellen. Jede Schachtel erhält eine aufsteigende Nummer, über die sie eindeutig angewählt werden kann. Zur Liste lassen sich dynamisch weitere Variablen hinzufügen oder entfernen. Listen lassen sich auch nach Werten durchsuchen.



## Zufall

Computer können zufällige Zahlen erzeugen. Damit lassen sich unvorhersehbare (zufällige) Elemente in ein Programm einfügen.



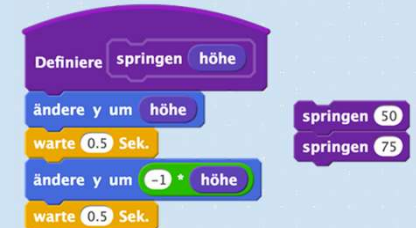
## Unterprogramme (Prozeduren)

Werden an unterschiedlichen Stelle in einem Programm die gleichen Befehlsfolgen verwendet, können stattdessen Unterprogramme (eigene Blöcke) verwendet werden. Ist später eine Änderung nötig, muss diese nur an genau einer Stelle erfolgen. Die Zerlegung und Lösung von Teilproblemen in kleineren, wiederverwendbaren Bausteinen ist ein wichtiges Konzept der Informatik.



## Parameter

Viele Befehle lassen sich durch Parameter steuern. Mit Parametern lässt sich zum Beispiel angeben, wie weit eine Figur laufen, wie laut ein Klang abgespielt oder wie lange auf den nächsten Befehl gewartet werden soll. Bei eigenen Unterprogrammen lassen sich die Parameter auch selbst definieren und als Variablen im Unterprogramm verwenden.



## Zeichenketten

Beim Programmieren wird Text als lange Folge einzelner Zeichen, sogenannte Zeichenketten definiert. Jedes Zeichen hat eine aufsteigende Nummer und lässt sich wie bei Listen einzeln lesen und schreiben. Zeichenketten können zusammengefügt werden, um neue Zeichenketten zu erzeugen.



## Ereignisbehandlung (Event handling)

Computerprogramme können auf Ereignisse reagieren und z. B. beim Drücken einer Taste, oder bei einem Geräusch (Mikrofon) ein Teilprogramm starten.



## Koordination und Synchronisation (messaging)

Passieren mehrere Dinge parallel in einem Programm, müssen manchmal einzelne Programmteile aufeinander abgestimmt werden. Zum Beispiel soll eine Figur erst dann auf der Bühne erscheinen, wenn der Regisseur ihr ein Zeichen gibt. So ein Zeichen (oder Nachricht) kann gesendet und empfangen werden.



## Instanziierung (cloning)

Programme können sich teilweise auch selbst vervielfältigen. Aus einem Programm für einen herumfliegenden Ball lassen sich mühelos 10 Ballprogramme instanzieren, die sich anschliessend genau gleich verhalten.



## Nebenläufigkeit (Threads, parallele Ausführung)

Nur die wenigsten Computerprogramme laufen von Anfang bis Ende an einem Stück. Meist laufen mehrere Teilprogramme scheinbar gleichzeitig ab. Bei einem Autorennen wird zum Beispiel fortlaufend auf Tastatureingaben geprüft und das Auto gedreht oder beschleunigt, wobei gleichzeitig der Computergegner seine Runden fährt.





## Worum geht es?

Diese Broschüre bietet eine Reihe vielfältiger Projektideen für Scratch. Dabei wird jeweils ein Kernidee und einige Skript-Blöcke gezeigt, die ein erstes Erfolgserlebnis ermöglichen. Nach dieser Starthilfe bleibt jedoch sehr vieles offen und der individuellen Gestaltung überlassen. Die Broschüre eignet sich damit insbesondere für Projektarbeiten, wenn Scratch als Umgebung bereits bekannt und grundlegende Konzepte erarbeitet wurden.

## Was wird benötigt?

Ein Computer mit Webbrowser und aktueller Flash-Version oder die App Pyonkee auf dem iPad. Für Windows/MacOs/Linux ist eine online und eine offline Version von Scratch verfügbar. Wir empfehlen die Nutzung der online Variante unter [scratch.mit.edu](http://scratch.mit.edu)

## Was lässt sich damit lernen?

Mit Scratch lassen sich Kompetenzen in den Bereichen Medien und Informatik, Technisches Gestalten, Natur und Technik, Musik, Bewegung und Sport sowie überfachliche Kompetenzen wie Teamfähigkeit und Sozialkompetenz erlernen.

## Warum Scratch?

Scratch ist eine am MIT entwickelte, frei verfügbare Programmierungsumgebung für Kinder und Jugendliche nach dem «low floor – wide walls – high ceiling»-Prinzip. Es existiert bereits viel Material sowie unter [scratch.mit.edu](http://scratch.mit.edu) über 16 Millionen einseh- & änderbare Programme.

## Das “Low floor - wide walls - high ceiling”-Prinzip

Scratch versucht das “low floor - wide walls - high ceiling”-Prinzip umzusetzen:

- **Low floor / Leichter Einstieg:** Für erste Projekte sind keine Vorkenntnisse nötig. Erste Erfolge und Aha-Erlebnisse nach fünf Minuten!
- **Wide walls / Verschiedenste Zugangsweisen:** Mit Scratch lassen sich ganz unterschiedliche Projekte realisieren – animierte Geschichte, Simulationen, Spiele usw. Scratch-Projekte verbinden Kompetenzen unterschiedlichster Fachbereiche wie Mathematik, Deutsch, Musik, NMG und natürlich Medien & Informatik.
- **High ceiling / Nach oben offen:** Mit Scratch lassen sich auch komplexere Projekte umsetzen. Zudem lassen sich vielfältige Sensoren, Roboter und andere Elektronik mit Scratch verbinden. Damit lässt sich die Funktionspalette praktisch beliebig erweitern.

## ScratchJr

Für jüngere Kinder eignet sich ScratchJr, welches deutlich weniger Funktionen bietet. ScratchJr ist kostenlos als App für Android, iOS und Amazon verfügbar.



## Die Idee des Konstruktivismus

Scratch wurde mit der Idee des Konstruktivismus des Mathematikers und Piaget-Studenten Seymour Papert im Hinterkopf entwickelt. Die Idee des Konstruktivismus baut auf dem Konstruktivismus auf und geht davon aus, dass Menschen besonders dann etwas lernen, wenn sie etwas mit persönlicher Bedeutung selbst konstruieren und sich im Konstruktionsprozess das dafür nötige Wissen aneignen und anwenden. Das eigene aktive Handeln steht im Mittelpunkt um auch theoretische Überlegungen und Modelle konkret fassbar und begreifbar zu machen. Mehr zum Konstruktivismus erfährt man in Paperts Buch: *Mindstorms. Children, Computer and Powerful Ideas, Basic Books, New York, 1980*

## Impressum / Kontakt

Pädagogische Hochschule Schwyz  
Michael Hielscher / Beat Döbeli  
[michael.hielscher@phsz.ch](mailto:michael.hielscher@phsz.ch)  
[beat.doebeli@phsz.ch](mailto:beat.doebeli@phsz.ch)