



Einführung

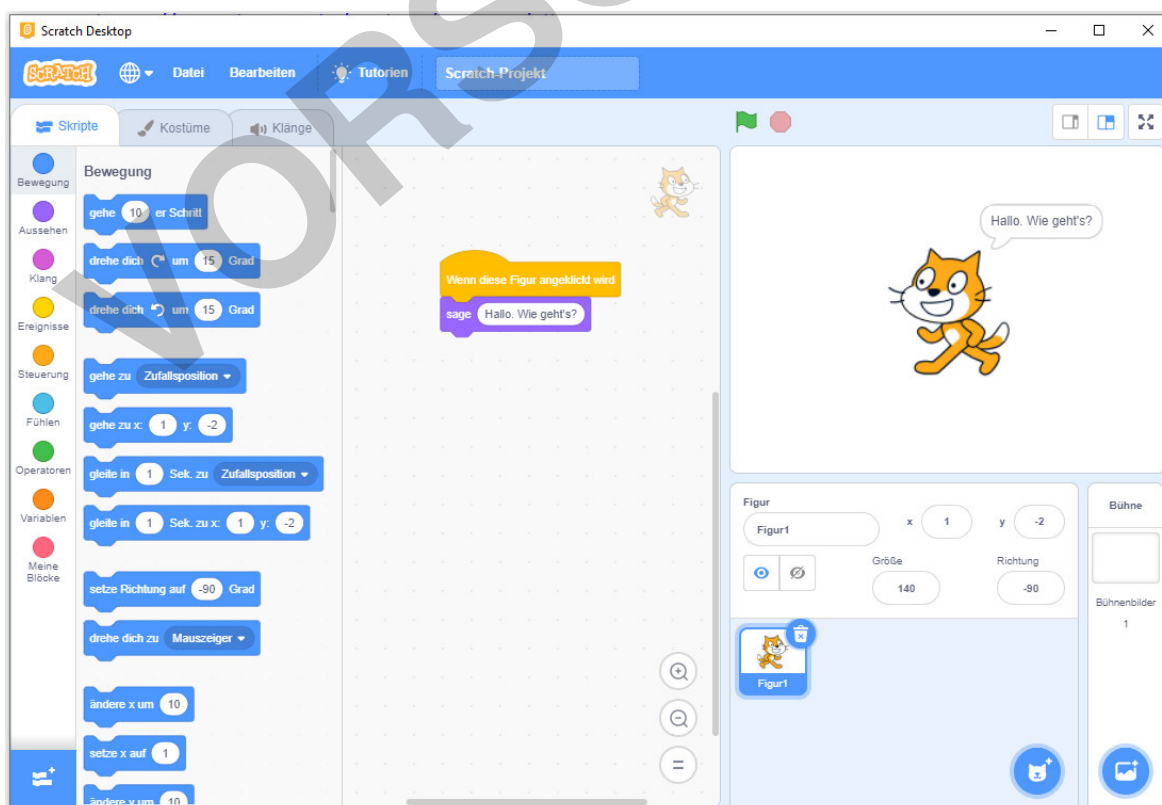
Diese Einheit führt die Schüler über die Arbeit mit Scratch an grundlegende algorithmische Strukturen heran. Theoretisch eingeführt werden der Algorithmen-Begriff, Anweisungen und Sequenzen, Schleifen, Bedingungen und Alternativen. Der Schwerpunkt des Moduls liegt aber auf der praktischen Arbeit mit Scratch und dem entdeckenden Lernen. Scratch ist dafür optimal geeignet und bietet die Möglichkeit einer Lernprogression vom ersten Einstieg in die Welt der Programmierung bis hin zu sehr komplexen interaktiven Animationen, Geschichten und Spielen.

Was ist Scratch?

Scratch ist eine grafische Programmiersprache mit der zugehörigen Entwicklungsumgebung, in der komplexe Programme aus einfachen Anweisungsblöcken erstellt werden können. Außerdem umfasst es eine weltweite Online-Community, in der Projekte geteilt und weiterentwickelt werden können. Scratch kann browserbasiert in nahezu allen gängigen Browsern (nicht Internet Explorer!) oder in einer Desktop-Version verwendet werden. Projekte in diesem Modul sind als Desktop-Projekte angelegt, können aber auch in ein Online-Studio hochgeladen werden. Um die Projekte anzuzeigen, muss Scratch in der Version 3.0 oder später verwendet werden.

Scratch ist ein Projekt der [Scratch Foundation](https://scratch.mit.edu) in Zusammenarbeit mit der Lifelong Kindergarten group am MIT Media Lab. Es ist kostenlos auf <https://scratch.mit.edu> erhältlich. Die Desktop-Scratch-Version 3.6.0 liegt diesem Modul im Ordner „7. Material“ bei.

Einen Eindruck der vielfältigen Möglichkeiten, die Scratch bietet, erhält man auf <https://scratch.mit.edu/explore/projects/all>.





Rezeptpuzzle

Das folgende Rezept für einen leckeren Apfelkuchen ist etwas durcheinandergeraten. Versuche die einzelnen Anweisungen so zu sortieren, dass beim Backen auch wirklich ein leckerer Kuchen herauskommt.



<https://pixabay.com/de/photos/apfelkuchen-kuchen-streusel-s%C3%BC%C3%9F-282747/>
(Pixabay License, freie kommerzielle Nutzung, 21.10.2019)

Eine Springform (Durchmesser 26 cm) gut mit Butter einfetten.

Den Teig in Frischhaltefolie wickeln und 60 Min im Kühlschrank kaltstellen.

Äpfel in einer Schüssel mit 1 Prise Zimt und 1 Teelöffel Zitronensaft vermengen.

Den Teig in die Form legen und mit den Fingern zurecht drücken, bis ein etwa 3 cm hoher Rand entstanden ist.

Fertigen Kuchen aus der Form lösen.

Apfelwürfel auf dem Teig verteilen.

Kuchen vollständig auskühlen lassen

Für die Füllung 800 g Äpfel schälen, Kerngehäuse entfernen und würfeln.

Für die Streusel 125 g Butter (zimmerwarm), 80 g Zucker, 1 Päckchen

Den Ofen auf 180 Grad (Umluft 160 Grad) vorheizen.

Streusel auf dem Kuchen verteilen.

Auf leicht bemehlter Arbeitsfläche den Teig rund ausrollen, bis er etwas

Kuchen im vorgeheizten Ofen 55-60 Minuten backen.

Den Teig aus dem Kühlschrank nehmen.

Für den Teig 125 g Butter, 100 g Zucker, 1 Päckchen Vanillezucker, 1 Ei, 250 g Mehl, 1/2 Päckchen Backpulver und 1 Prise Salz miteinander verkneten.



Rezeptpuzzle

Dies ist das Original-Rezept¹:

1. Für den Teig 125 g Butter, 100 g Zucker, 1 Päckchen Vanillezucker, 1 Ei, 250 g Mehl, 1/2 Päckchen Backpulver und 1 Prise Salz miteinander verkneten.
2. Den Teig in Frischhaltefolie wickeln und 60 Min im Kühlschrank kaltstellen.
3. Den Ofen auf 180 Grad (Umluft 160 Grad) vorheizen.
4. Eine Springform (Durchmesser 26 cm) gut mit Butter einfetten.
5. Den Teig aus dem Kühlschrank nehmen.
6. Auf leicht bemehlter Arbeitsfläche den Teig rund ausrollen, bis er etwas größer als die Springform ist.
7. Den Teig in die Form legen und mit den Fingern zurecht drücken, bis ein etwa 3 cm hoher Rand entstanden ist.
8. Für die Füllung 800 g Äpfel schälen, Kerngehäuse entfernen und würfeln.
9. Äpfel in einer Schüssel mit 1 Prise Zimt und 1 Teelöffel Zitronensaft vermengen.
10. Apfelwürfel auf dem Teig verteilen.
11. Für die Streusel 125 g Butter (zimmerwarm), 80 g Zucker, 1 Päckchen Vanillezucker, 200 g Mehl und 1 Prise Salz zu groben Streuseln kneten.
12. Streusel auf dem Kuchen verteilen.
13. Kuchen im vorgeheizten Ofen 55-60 Minuten backen.
14. Kuchen vollständig auskühlen lassen
15. Fertigen Kuchen aus der Form lösen.

Versuch doch einmal, den Kuchen zu backen. Guten Appetit!



<https://pixabay.com/de/photos/apfelkuchen-kuchen-streusel-s%C3%BC%C3%9F-282747/>
(Pixabay License, freie kommerzielle Nutzung, 21.10.2019)

¹ Rezept bearbeitet nach: <https://www.einfachbacken.de/rezepte/apfelkuchen-mit-streusel-nach-omas-rezept> (22.10.2019)



1. Was sind Algorithmen?

In diesem Kapitel geht es um **Algorithmen**. Vielleicht hast du den Begriff schon einmal gehört. In den Medien kommt er immer wieder vor. Oft hören wir ihn zum Beispiel im Zusammenhang mit Google und Facebook, die mit ihren Algorithmen großen Einfluss auf unseren Alltag haben². Aber viele Menschen wissen trotzdem nicht, was sich hinter dem Begriff wirklich verbirgt. Dabei ist es eigentlich ganz einfach.

Merke: Ein Algorithmus ist eine Reihe von Anweisungen, die Schritt für Schritt ausgeführt werden, um ein Problem zu lösen.

So wie bei dem Kuchenrezept zum Einstieg.

Problem:

Wie backe ich einen leckeren Apfelkuchen?

Lösung:

Schritt 1: Für den Teig 125 g Butter, 100 g Zucker, 1 Päckchen Vanillezucker, 1 Ei, 250 g Mehl, 1/2 Päckchen Backpulver und 1 Prise Salz miteinander verkneten.

Schritt 2: Den Teig in Frischhaltefolie wickeln und 60 Min im Kühlschrank kaltstellen.

Und so weiter.

Wenn du die Schritte genau befolgst, sollte am Ende das Problem gelöst sein und ein leckerer Apfelkuchen auf dem Tisch stehen.

Wichtige Eigenschaften von Algorithmen³

Damit eine Serie von Anweisungen zu einem Algorithmus wird, muss sie jedoch einige Voraussetzungen erfüllen. Nicht jede Reihe von Anweisungen ist also auch schon ein Algorithmus.

1. Die Anweisungen eines Algorithmus müssen eindeutig ausführbar sein.

Stell dir vor, Schritt 13 hieße nur: Kuchen im vorgeheizten Ofen backen.

Diese Anweisung wäre nicht eindeutig. Man wüsste nicht, wie lange der Kuchen gebacken werden muss. Für einen erfahrenen Bäcker wäre dies kein Problem. Man wartet solange, bis der Kuchen gut riecht und leicht braun aussieht. Zur Kontrolle steckt man ein Holzstäbchen in den Kuchen und wenn kein Teig mehr daran haftet, dann



Tip: Der Begriff „Algorithmus“ oder Mehrzahl „Algorithmen“ ist schon sehr alt. Im 9. Jahrhundert verfasste der persische Mathematiker Muhammad Ibn-Musa al-Chwarizmi ein Lehrbuch, das im 12. Jahrhundert ins Lateinische übersetzt wurde. Dabei wurde aus seinem Namen „al-Chwarizmi“ das Wort „Algorismi“ und später unser heutiges Wort „Algorithmus“. Algorithmen sind also keine neue Erfindung. Das Bild zeigt Muhammad Ibn-Musa al-Chwarizmi auf einer sowjetischen Briefmarke anlässlich seines 1200-jährigen Geburtsjubiläums.

[https://de.wikipedia.org/wiki/Datei:1983 CPA 5426 \(1\).png](https://de.wikipedia.org/wiki/Datei:1983 CPA 5426 (1).png) (kein Urheberrecht, 22.10.2019)

² Bei Google und Facebook entscheiden Algorithmen zum Beispiel, welche Suchergebnisse oder welche Werbung du angezeigt bekommst.

³ Anmerkung: Die Eigenschaften von Algorithmen sind stark didaktisch reduziert und verwenden Analogien, um die wesentlichen Konzepte deutlich zu machen. Eine tiefergehende Behandlung ist an dieser Stelle zu komplex.

ist der Kuchen fertig. Für einen unerfahrenen Bäcker könnte dieser Schritt jedoch dazu führen, dass sein Kuchen hinterher innen noch nicht durch ist oder aber verbrannt.

Die Anweisungen eines Algorithmus müssen immer so formuliert sein, dass sie eindeutig ausgeführt werden können und dass man immer genau weiß, welcher Schritt als Nächstes kommt.

2. Das Ergebnis des Algorithmus muss wiederholbar und korrekt sein.

Werden die Anweisungen des Algorithmus ausgeführt, müssen sie bei gleichen Voraussetzungen immer zum selben und korrekten Ergebnis führen – es soll in dem Rezeptbeispiel also immer ein leckerer Kuchen dabei herauskommen. Wenn ich einem Backrezept genau folge und das Ergebnis ist einmal ein roher, einmal ein verbrannter und einmal ein leckerer Kuchen, dann ist das Rezept nicht besonders gut – und als Algorithmus ungeeignet.

3. Der Algorithmus muss irgendwann enden.

Stellen wir uns vor, nicht ein Mensch, sondern ein Roboter würde das Rezept ausführen. Träfe der Roboter auf den Schritt „Kuchen im vorgeheizten Ofen backen“, gäbe es ein Problem. Die Anweisung sagt nur, dass man den Kuchen im Ofen backen soll, aber nicht, wann man aufhören soll zu backen. Da diese Anweisung kein „Ende“ hat, würde der Roboter einfach immer weiter backen. Der Algorithmus würde niemals enden. Natürlich geht das nicht. Irgendwann würde es fürchterlich anfangen zu qualmen, während der Kuchen im Ofen verschmort ... Ein echter Algorithmus muss daher immer festlegen, wie lange, wie oft oder unter welchen Bedingungen etwas getan werden soll – und wann damit aufgehört werden soll.

Merke:

Ein Algorithmus muss folgende Eigenschaften besitzen:

- * Die Anweisungen müssen eindeutig sein.
- * Die Anweisungen müssen ausführbar sein.
- * Es muss immer klar sein, welcher Schritt als Nächstes folgt.
- * Der Algorithmus muss bei gleichen Voraussetzungen immer dasselbe und korrekte Ergebnis liefern.
- * Der Algorithmus muss nach einer endlichen Anzahl von Schritten zum Ende kommen.

Struktogramme 1

Obwohl wir im Alltag häufig Algorithmen begegnen (zum Beispiel in Koch- oder Backrezepten, Bastel- oder Montageanleitungen etc.), sind sie vor allem in der Computerwelt wichtig. Hat man eine Reihe von Anweisungen als Algorithmus formuliert, dann kann man sie in eine Programmiersprache übersetzen, sodass ein Computer sie ausführen kann. Dies nennen wir Programmieren. Deshalb sind Algorithmen die Grundlage von vielen Computerprogrammen.

In der Informatik beschreiben wir die notwendigen Schritte eines Algorithmus jedoch oft nicht in einfachen Sätzen. Stattdessen verwenden wir oft grafische Darstellungen. Ein Weg, einen Algorithmus darzustellen, ist ein **Struktogramm**.

Im einfachsten Fall werden dabei die Anweisungen einfach in jeweils einen rechteckigen **Strukturblock** geschrieben.

Ein einfaches Struktogramm sieht so aus:

Anweisung 1
Anweisung 2
Anweisung 3

← Strukturblock

...

Ein Programm, das eine solche (lineare = gradlinige) Struktur aufweist, nennt man **Sequenz**. (Anmerkung: Später lernen wir Programme mit komplizierteren Strukturen und Struktogrammen kennen.)

Für unseren Apfelkuchen könnte das einfache Struktogramm so aussehen:

125 g Butter, 100 g Zucker, 1 Päckchen Vanillezucker, 1 Ei, 250 g Mehl, 1/2 Päckchen Backpulver und 1 Prise Salz verkneten
Teig in Frischhaltefolie wickeln und 60 Min in Kühlschrank stellen
Ofen auf 180 Grad vorheizen
Springform (Durchmesser 26 cm) mit Butter einfetten
Teig aus Kühlschrank nehmen
Arbeitsfläche mit Mehl bestreuen, Teig als Kreis mit 30 cm Durchmesser ausrollen
Teig in die Form legen, mit den Fingern so andrücken, dass rundherum ein 3 cm hoher Rand entsteht.
800 g Äpfel schälen, Kerngehäuse entfernen, Äpfel würfeln
Apfelwürfel in einer Schüssel mit 1 Prise Zimt und 1 Teelöffel Zitronensaft vermengen
Apfelwürfel auf Teig verteilen



1. Was sind Algorithmen?

Aufgaben

1. Schreibe einen Algorithmus zum Hochfahren deines Computers. Diskutiere anschließend mit deinem Nachbarn dein Ergebnis. Habt ihr dieselben Schritte aufgeschrieben? Sind die Schritte wirklich eindeutig und klar?

2. Stellt euer Ergebnis aus Aufgabe 1 in einem Struktogramm dar.

3. Schreibe mit deinem Partner zusammen einen Algorithmus zum Putzen der Zähne am Morgen. Welche Schritte sind nötig, um sicherzustellen, dass der nächste Zahnarztbesuch nicht unangenehm wird?



1. Was sind Algorithmen?

Aufgaben

1. Schreibe einen Algorithmus zum Hochfahren deines Computers. Diskutiere anschließend mit deinem Nachbarn dein Ergebnis. Habt ihr dieselben Schritte aufgeschrieben? Sind die Schritte wirklich eindeutig und klar?

1. Startknopf des Computers drücken
2. Bildschirm einschalten
3. Computer hochfahren lassen
4. Ein Konto aussuchen
5. Passwort eingeben (sich anmelden)

2. Stellt euer Ergebnis aus Aufgabe 1 in einem Struktogramm dar.

Startknopf des Computers drücken
Bildschirm einschalten
Computer hochfahren lassen
Ein Konto aussuchen
Passwort eingeben (sich anmelden)

3. Schreibe mit deinem Partner zusammen einen Algorithmus zum Putzen der Zähne am Morgen. Welche Schritte sind nötig, um sicherzustellen, dass der nächste Zahnarztbesuch nicht unangenehm wird?

1. Ins Bad gehen
2. Zahnbürste aus Behälter nehmen
3. Zahnpasta-Tube aufschrauben
4. Zahnpasta aus Tube drücken
5. 3 Minuten putzen
6. Mit Wasser ausspülen
7. Zahnbürste ausspülen
8. Zahnbürste in Behälter stellen

Anmerkung: Man könnte hier noch berücksichtigen, dass die Zahnpasta vielleicht leer ist oder die Zahnbürste schon so alt, dass sie ausgewechselt werden muss. Berücksichtigt man solche

Eventualitäten, dann kann ein solcher Algorithmus noch deutlich komplizierter werden. Abweichungen können sich auch dann ergeben, wenn zum Beispiel eine elektrische Zahnbürste verwendet wird.

VORSCHAU



Algorithmen im Überblick

Merke: Ein **Algorithmus** ist eine Reihe von Anweisungen, die Schritt für Schritt ausgeführt werden, um ein Problem zu lösen.

Ein Algorithmus muss folgende Eigenschaften besitzen:

- * Die Anweisungen müssen eindeutig sein.
- * Die Anweisungen müssen ausführbar sein.
- * Es muss immer klar sein, welcher Schritt als Nächstes folgt.
- * Der Algorithmus muss bei gleichen Voraussetzungen immer dasselbe und korrekte Ergebnis liefern.
- * Der Algorithmus muss nach einer endlichen Anzahl von Schritten zum Ende kommen.

Algorithmen lassen sich in **Struktogrammen** darstellen.

Eine lineare Abfolge von Anweisungen heißt **Sequenz** und wird folgendermaßen dargestellt:

Anweisung 1
Anweisung 2
Anweisung 3

← **Strukturblock**

...

VORSCHAU



1. Was sind Algorithmen?

Lernzielkontrolle

1. Beschreibe in einem Algorithmus deinen Ablauf am Morgen an einem Schultag.

2. Stelle deinen Algorithmus als Struktogramm dar.

VORSCHAU



1. Was sind Algorithmen?

Lernzielkontrolle

1. Beschreibe in einem Algorithmus deinen Ablauf am Morgen an einem Schultag.

Zum Beispiel:

1. Aufstehen
2. Ins Bad gehen
3. Schlafanzug ausziehen
4. Duschen
5. Abtrocknen
6. Anziehen
7. Haare fönen und kämmen
8. Frühstücken
9. Vesper richten und einpacken
10. Zähne putzen
11. Schulsachen zusammenpacken
12. Jacke anziehen
13. Auf Bus rennen

2. Stelle deinen Algorithmus als Struktogramm dar.

Aufstehen
Ins Bad gehen
Schlafanzug ausziehen
Duschen
Abtrocknen
Anziehen
Haare fönen und kämmen
Frühstücken
Vesper richten und einpacken
Zähne putzen
Schulsachen zusammenpacken
Jacke anziehen
Auf Bus rennen

2. Scratch entdecken

In den folgenden Kapiteln arbeiten wir mit **Scratch** und werden eigene Algorithmen entwickeln und testen.

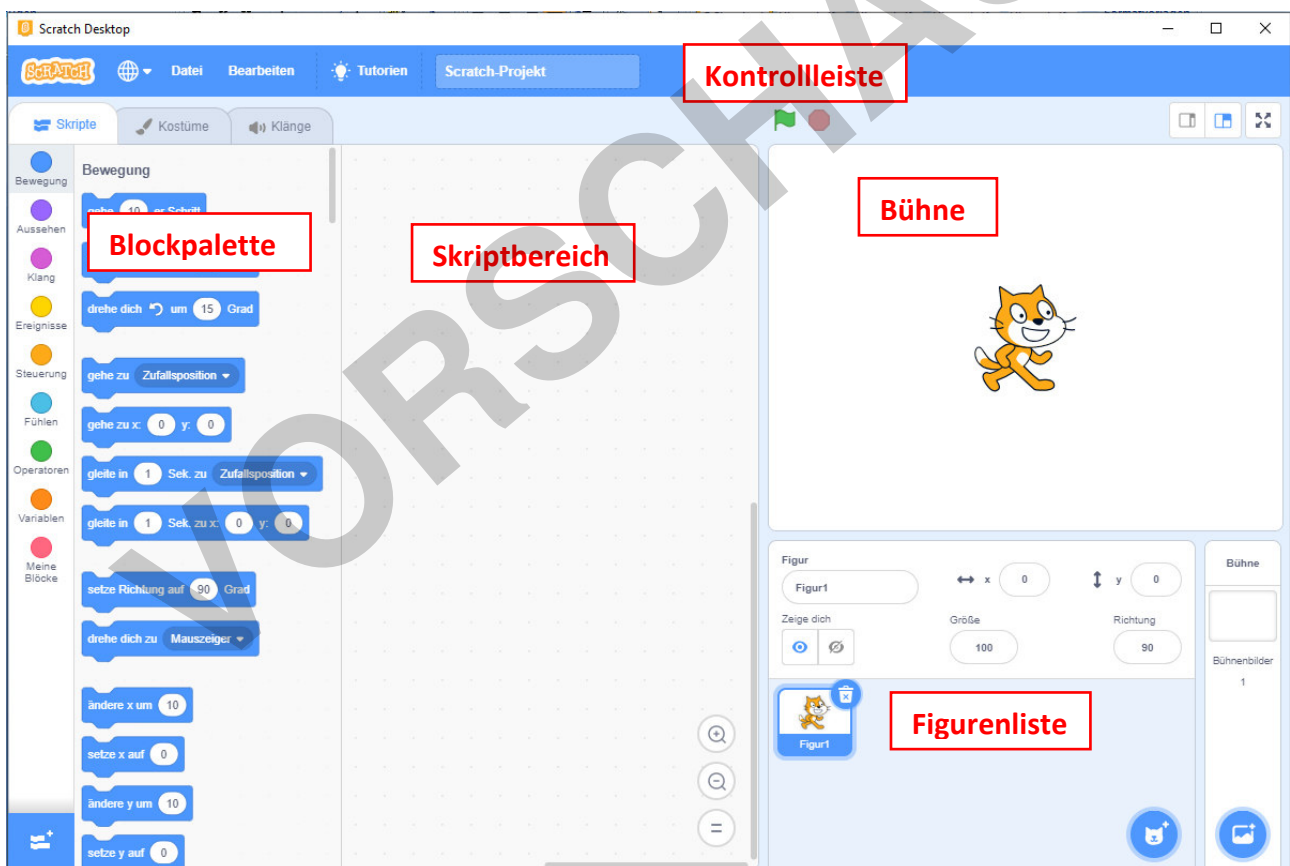
Scratch ist eine grafische Programmiersprache mit einer eigenen Entwicklungsumgebung, in der Anweisungen ähnlich wie bei einem Struktogramm zu Programmen kombiniert werden können. Auf diesem Weg kannst du einfache Animationen, aber auch interaktive Geschichten und selbst ganze Spiele programmieren.

Scratch hat eine große Online-Community. Wenn du dich anmeldest, kannst du deine eigenen Projekte mit anderen teilen, andere Projekte kommentieren und „**liken**“ und sogar verändern und weiterentwickeln. Für die Arbeit mit diesem Modul ist jedoch kein eigenes Konto nötig. Wir werden nur mit der Desktop-Version von Scratch arbeiten.

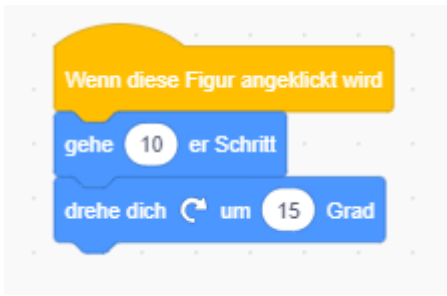


Hallo, ich bin Fred, die Scratch-Katze!

Die wichtigsten Funktionen der Scratch-Oberfläche



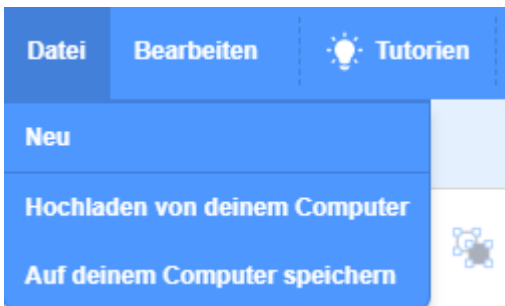
Auf der Registerkarte **Skripte** siehst du links die **Blockpalette**. Sie enthält alle **Blöcke** mit Anweisungen, die in Scratch verfügbar sind. Die einzelnen Blöcke sind verschiedenen Kategorien (Bewegung, Aussehen, Klang usw.) zugeordnet, die auch farblich unterschieden werden, um schneller den gewünschten Block zu finden. Die Blöcke können einfach in den **Skriptbereich** gezogen und untereinander „geklickt“ werden, um sie zu einem Programm zu kombinieren.



Die Figur auf der **Bühne** führt dann die jeweiligen Anweisungen aus. Über die **Figurenliste** kannst du weitere Figuren hinzufügen und ganz rechts unten kannst du ein Bühnenbild als Hintergrund auswählen.

Über die Registerkarten **Kostüme** und **Klänge** kannst du bestehende Figuren und Klänge bearbeiten und neue hinzufügen.

In der Kontrollleiste findest du im Menü **Datei** die Option **Auf deinem Computer speichern**, um dein jeweilig aktuelles Projekt zu speichern. Über **Hochladen von deinem Computer** öffnest du ein bestehendes Projekt. **Neu** erzeugt ein ganz neues Projekt.



Über die Option **Tutorien** kannst du verschiedene Tutorials mit Anleitungen aufrufen, um verschiedene Techniken kennenzulernen. Leider sind die meisten Tutorials in Englisch.



2. Scratch entdecken

Aufgaben

1. Gehe auf <https://scratch.mit.edu/explore/projects/all>. Erkunde 10 Minuten lang die verschiedenen Projekte, die du dort findest. Welches Scratch-Projekt hat dir besonders gefallen?

Beschreibe dein Lieblingsprojekt (Name, worum geht es dabei?)

Gibt es Projekte, die du selbst einmal gerne erstellen würdest?

2. Öffne Scratch. Finde in der Blockpalette auf der linken Seite die folgenden 8 Blöcke und ziehe sie in den Skriptbereich.



Experimentiere 15 Minuten, indem du diese Blöcke in verschiedenen Weisen kombinierst. Du kannst jeden Block sooft verwenden, wie du möchtest. Mit einem Rechtsklick kannst du einzelne

oder mehrere Blöcke duplizieren. Kannst du Fred, die Scratch-Katze, ein paar interessante Sachen machen lassen?

3. Vergleiche dein Projekt mit deinen Banknachbarn links und rechts. Speichere dein Projekt ab.

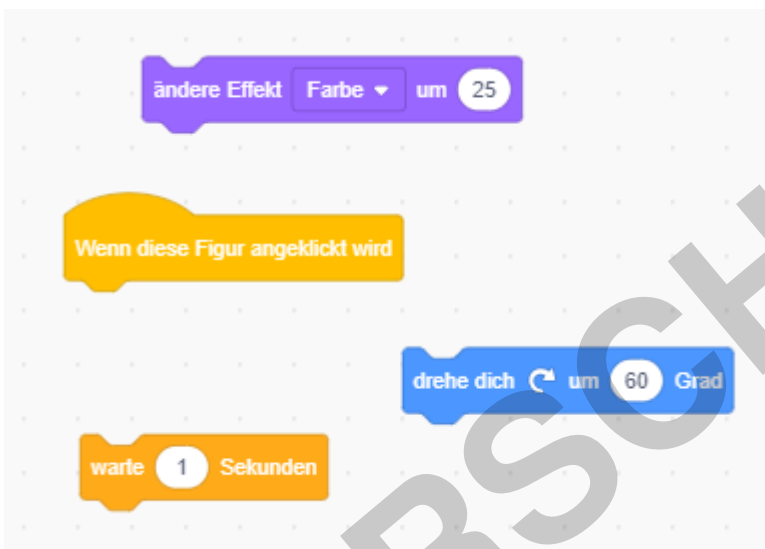
4. Betrachte die folgenden Animationen. Versuche, dasselbe Verhalten zu erreichen. Zeichne jeweils ein Struktogramm, das die zugehörige Sequenz beschreibt.

Anmerkung: In den folgenden Struktogrammen kannst du für **Wenn diese Figur angeklickt wird** einfach einen Strukturblock verwenden.

a.



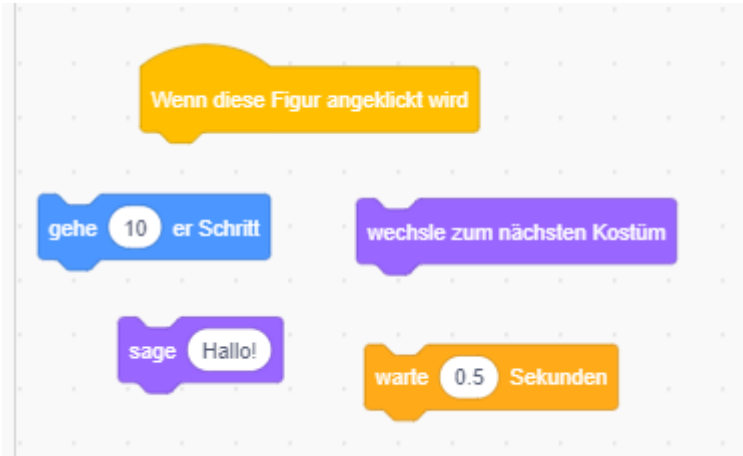
Verwende nur diese Blöcke:



b.



Verwende nur diese Blöcke:



VORSCHAU



2. Scratch entdecken

Aufgaben

1. Gehe auf <https://scratch.mit.edu/explore/projects/all>. Erkunde 10 Minuten lang die verschiedenen Projekte, die du dort findest. Welches Scratch-Projekt hat dir besonders gefallen?

Beschreibe dein Lieblingsprojekt (Name, worum geht es dabei?)

Offen.

Gibt es Projekte, die du selbst einmal gerne erstellen würdest?

Offen.

2. Öffne Scratch. Finde in der Blockpalette auf der linken Seite die folgenden 8 Blöcke und ziehe sie in den Skriptbereich.



Experimentiere 15 Minuten, indem du diese Blöcke in verschiedenen Weisen kombinierst. Du kannst jeden Block sofort verwenden, wie du möchtest. Mit einem Rechtsklick kannst du einzelne oder mehrere Blöcke duplizieren. Kannst du Fred, die Scratch-Katze, ein paar interessante Sachen machen lassen?


Offen.

3. Vergleiche dein Projekt mit deinen Banknachbarn links und rechts. Speichere dein Projekt ab.

Offen.

4. Betrachte die folgenden Animationen. Versuche, dasselbe Verhalten zu erreichen. Zeichne jeweils ein Struktogramm, das die zugehörige Sequenz beschreibt.

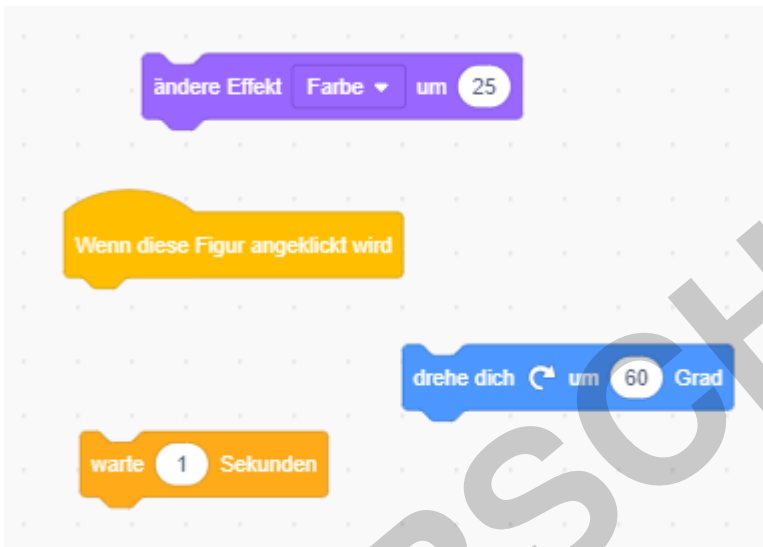
Anmerkung: In den folgenden Struktogrammen kannst du für **Wenn diese Figur angeklickt wird**

einfach einen Strukturblock  verwenden.

a.



Verwende nur diese Blöcke:



VORSCHAU



Start
drehe dich nach rechts um 60 Grad
ändere Farbe um 25
warte 1 Sekunde
drehe dich nach rechts um 60 Grad
ändere Farbe um 25
warte 1 Sekunde
drehe dich nach rechts um 60 Grad
ändere Farbe um 25
warte 1 Sekunde
drehe dich nach rechts um 60 Grad
ändere Farbe um 25
warte 1 Sekunde
drehe dich nach rechts um 60 Grad
ändere Farbe um 25
warte 1 Sekunde
drehe dich nach rechts um 60 Grad
ändere Farbe um 25