

Einführung in das Programmieren mit Scratch 3



Scratch ist ein Projekt der Scratch-Stiftung und der Lifelong Kindergarten Group am MIT Media Lab. Es steht kostenlos unter <https://scratch.mit.edu> zur Verfügung.

Workshopautor: Adrian Sauer **Co-Autorin:** Swetlana Reile

Dieser Workshop beruht tw. auf dem Werk von Eisenmann (2017) zu erreichen unter: https://lehrerfortbildung-bw.de/u_matnatech/informatik/gym/bp2016/fb1/2_algorithmen/

Dieser Workshop

... ist so aufgebaut, dass zu einem Thema jeweils ein Video erstellt ist. Dieses kannst du mit Hilfe des QR-Codes oben rechts aufrufen. Du kannst aber auch die Webadresse, die darunter steht in einen Browser eintippen. Das Video leitet dich durch die Schritte, die du vornehmen sollst und erklärt dazu einige Inhalte. Das heißt, alle Schritte, die im Video gemacht werden, sollst du in deinem Scratch ebenfalls durchführen und ausprobieren. Dabei kannst du am besten nachvollziehen und beobachten, was passiert.

Deine eigene Programmierung solltest du nach jedem Kapitel speichern. Am besten nach einem Schema:

JJMMTT-Aufgabe-Name (J=Jahr, M=Monat, T=Tag)

Beispiel: Datum 20.4.2023, Aufgabe 21, Name: Mareike

Dateiname: 230420-21-Mareike

Im Anschluss werden dir weitere Aufgaben oder Fragen gestellt, die du beantworten sollst, bevor du mit dem nächsten Schritt weiter machst.

Viel Spaß dabei!



Wie du den QR-Code scannen kannst

Öffne die Kamera-App auf deinem **iPhone** und scanne den Code damit. Für **Android** suche im Play Store nach „QR Code Secuso“. Diese App ist kostenlos und datenschutzfreundlich. **Viel Erfolg!**



Wo ist das Programm?



Browser

Mit einem Browser gehst du ins Internet. Für diese Einführung benutzen wir das freie und kostenfreie Projekt "Mozilla Firefox".

Download unter <https://www.mozilla.org/de/firefox/new/>

Solltest du einen anderen Browser benutzen, können manche Abläufe anders sein.



Wie geht es los?

Starte mit dem Video!



[ogy.de/00x-SCRATCH](https://www.scratch.de/00x-SCRATCH)

Gehe auf die Webseite!

Du musst Scratch nicht installieren. Man kann online auf Scratch zugreifen:

Direkt einsteigen: <https://scratch.mit.edu/projects/editor/>

Übersichtsseite: <https://scratch.mit.edu/>



<https://scratch.mit.edu/projects/editor/>

Auf Deutsch umschalten

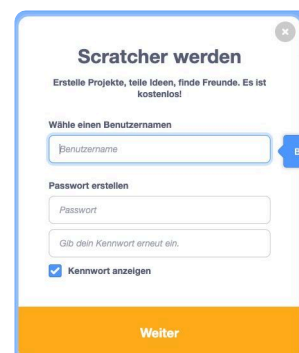
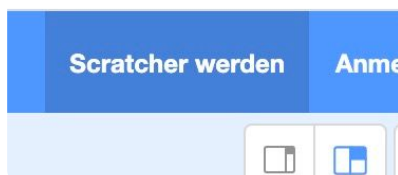
Du kannst oben links auf der "Weltkugel" die Sprache auf "Deutsch" umschalten.



Werde Scratcher

Oben rechts gibt es die Möglichkeit „Scratcher“ zu werden. So kannst du online all deine Projekte speichern und sichern.

In dieser Einführung gehen wir **nicht** davon aus, dass du dir einen Account angelegt hast, sondern werden alle Projekte downloaden und so sichern.

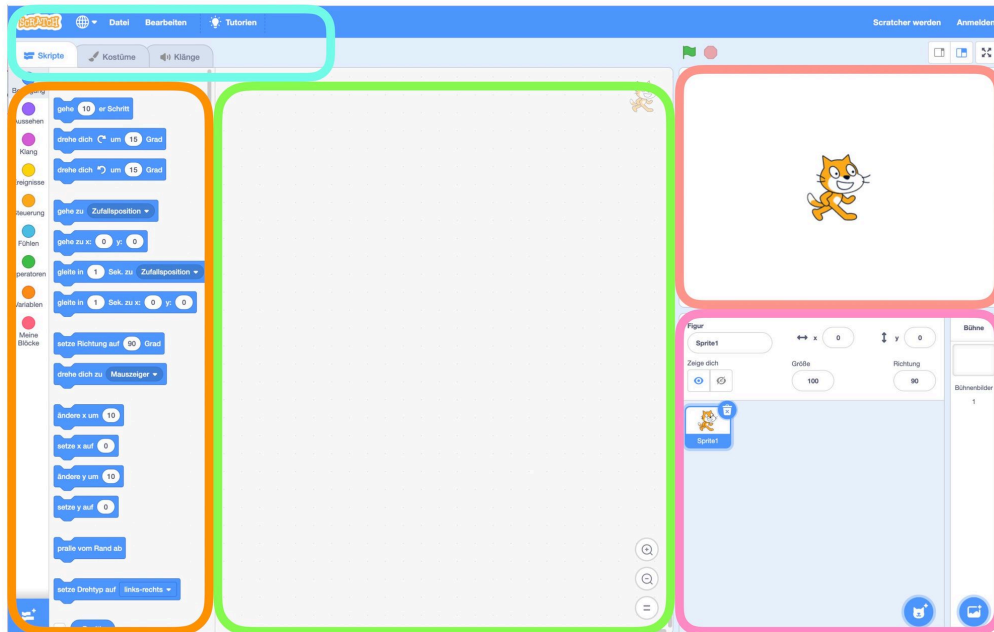


Wie funktioniert Scratch?

Starte mit dem Video!



[ogy.de/00y-SCRATCH](https://www.tutoty.de/00y-SCRATCH)



Die Oberfläche in Scratch teilt sich in diese Bereiche auf:



ORANGE: Blockpalette

In der Blockpalette findest du alle Programmierelemente, die du brauchst.

Du kannst sie per Drag&Drop in den grünen Bereich schieben.



GRÜN: Programmierbereich

Hierhin schiebst du die Programmierblöcke und fügst sie aneinander. Hier wird also "programmiert".



ROT: Die Bühne

Hier läuft alles ab, was du programmierst. Deine Figur(en) folgen den Anweisungen aus dem Programmierbereich



ROSA: Figurenbereich

Im Figurenbereich sind alle Figuren, auch "Sprites" genannt, die du benutzt. Hier kannst du sie außerdem bearbeiten.



TÜRKIS: Einstellungen

In den Einstellungen kannst du oben deine Programmierung sichern oder hochladen. Unten kannst du zwischen "Skripte" (deine Programmierung) und "Kostüme" (das Aussehen deiner Figuren) wechseln. Außerdem gibt es noch Einstellungen zu Klängen.



Start und Stopp

Über die grüne Flagge kannst du das Ablaufen deiner Programmierung starten.

Mit dem Stoppschild wieder stoppen.

Speichern, Hochladen und Umbenennen

Starte mit dem Video!



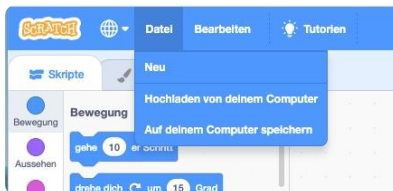
[tutogy.de/00z-SCRATCH](https://www.tutogy.de/00z-SCRATCH)

💡 Webtool

Damit du deine Programme speichern kannst und auch wieder öffnen, musst du sie herunter oder hochladen.

Scratch ist keine Software sondern ein "WebTool", das heißt, dass es nicht installiert wird sondern in deinem Webbrowser läuft.

0.1 Speichern



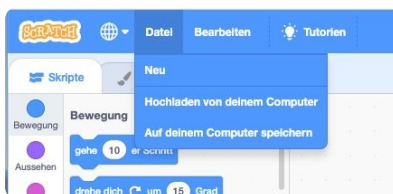
1. Klick oben links auf „Datei“
2. Klick auf „Auf deinem Computer speichern“
3. Firefox zeigt ein Fenster *Öffnen von Scratch-Projekt.sb3*
4. Klicke die Option „Datei speichern“ an und bestätige mit „OK“
5. Firefox speichert nun die Datei im eingestellten „Downloads“ Ordner.
6. Jetzt solltest du deine Datei aus dem „Download“ Ordner in deinen *Scratch* Ordner verschieben (*Ausschneiden* und *Einfügen*)
7. Die Dateien aus dem Download heißen meistens einfach „Scratch-Projekt.sb3“ - **du solltest sie umbenennen.**

Am besten nach diesem Schema:

J = Jahr; M = Monat; T = Tag ⇒ JJMMTT-Aufgabe-Name.sb3

Beispiel: 200921-Hase-Mareike.sb3

0.1 Hochladen



1. Klick oben links auf „Datei“
2. Klick auf „Hochladen von deinem Computer“
3. Firefox öffnet ein Fenster mit den Dateien auf deinem Rechner.
4. Navigiere zu der Scratch Datei, die du öffnen möchtest.
5. Klicke sie an und klicke auf „Öffnen“

💡 Achtung!

Wenn du versuchst eine Scratchdatei auf deinem Rechner mit einem Doppelklick zu öffnen, dann wird dies nicht funktionieren - da Scratch ein Webtool ist (siehe oben) und keine installierte Software! Du musst deine Projekte immer hochladen.

Hello World!

Starte mit dem Video!



[ogy.de/01-SCRATCH](https://www.tutortry.de/01-SCRATCH)

- ① Das Grundprinzip beim Programmieren ist, dass Befehle abgearbeitet werden müssen. Probiere mal ein paar Befehle aus. Dazu musst du auf sie klicken. Die Zahlen in den weißen Feldern kannst du verändern.

gehe 10 er Schritt

drehe dich um 15 Grad

drehe dich um 15 Grad

gehe zu Zufallsposition

gehe zu x: 10 y: 0

gleite in 1 Sek. zu Zufallsposition

ändere x um 10

setze x auf -230

ändere y um 10

setze y auf -72

Was passiert?

- a)
- b)
- c)
- d)
- e)
- f)
- g)
- h)
- i)
- j)

Was passiert, wenn ich hier -10 eintrage?

gehe 10 er Schritt

Anweisungen und Sequenzen

Das Grundprinzip beim Programmieren ist, dass Befehle abgearbeitet werden müssen. Man kann zu den Befehlen auch Anweisungen.

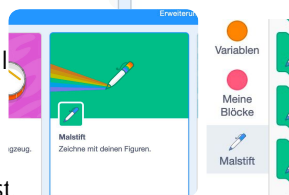
Wenn du der Katze mehrere **Anweisungen** hintereinander gibst, nennt man das **Sequenz**.

Meine Sequenz

Starte mit dem Video!



[ogy.de/02-SCRATCH](https://www.scratch.de/02-SCRATCH)



- ② Jetzt geht es aber mal richtig an das Programmieren. Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“.
Erledige die Schritte in genau dieser Reihenfolge!
- Ziehe 3 beliebige Blöcke per Drag'n'Drop von der Blockpalette in den Programmierbereich.
 - Lösche sie wieder, indem du die Blöcke vom Programmierbereich zurück auf die Blockpalette schiebst.
 - Füge den Block **Wenn Fahne angeklickt wird** zum Programmierbereich hinzu. (Du findest ihn im gelben Bereich *Ereignisse*)
 - Wechsel zurück in den blauen Bereich *Bewegung*.
Wenn du den Block **gleite in 1 Sek. zu Zufallsposition** in den Programmierbereich bewegst, dann passt er genau unter den gelben Block.
 - Klicke die grüne Fahne an und schau, was passiert.
 - Setze darunter den Block **gleite in 1 Sek. zu x: 0 y: 0** und teste wieder mit der grünen Fahne, was passiert.
 - Deine Sequenz wird immer länger! Super!
Jetzt wollen wir künstlerisch werden. Dafür müssen wir neue Blöcke „installieren“
> Klicke dazu unten links auf den blauen Block.
> Klicke auf „Malstift“
> Nun hast du einen neuen Palettenbereich „Malstift“
 - Schiebe nun den Block **schalte Stift ein** direkt unter den ersten Block deiner neuen Sequenz. Klicke ein paar mal auf die grüne Fahne.
 - Jetzt ist er sehr mühsam, jedes mal auf die grüne Fahne zu klicken. Füge deshalb die beiden blauen Blöcke noch ein paar Mal darunter ein. (Schritte d und f)
 - Sei kreativ und verändere die Zahlen hinter x und y ein wenig, schau zu, was passiert, wenn du die grüne Fahne klickst.
 - Tipp: Willst du die ganzen Spuren löschen? Klicke den Block **lösche alles** bei der Palette *Malstift* an.
 - Abschluss:** Bitte speichere die fertige Aufgabe ab. Dateinamenschema: **JJMMTT-Aufg-Name**

Wir erkunden die Skripte

Schau dir die verschiedenen Anweisungen/Befehle an, die du zu Sequenzen zusammenfügen kannst.
Ergänze die Tabelle.

Befehle	Farbe	typische Funktionen
Bewegung	_____	_____ _____
Aussehen	_____	_____ _____
Klang	_____	_____
Ereignisse	_____	_____
Steuerung	_____	_____
Fühlen	_____	_____ _____
Operatoren	_____	_____
Variablen	_____	_____
Meine Blöcke	_____	_____
Malstift	_____	_____

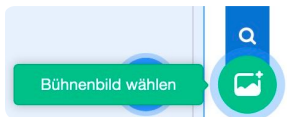
Erläutere den Unterschied zwischen einem Befehl/Anweisung und einer Sequenz.

Wir erkunden die Bühne

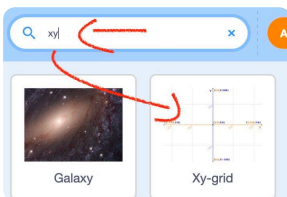
Starte mit dem Video!



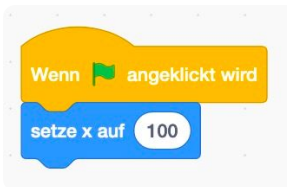
- ③ Jetzt geht es aber mal richtig an das Programmieren. Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“. Erledige die Schritte in genau dieser Reihenfolge!



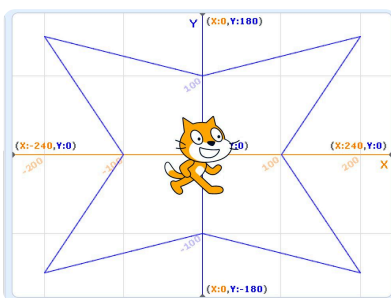
Bühnensuche



XY-Bühne



Vorlage



12

Wir sollten nun die Bühne etwas erkunden.

Diese ist aufgebaut wie ein Koordinatensystem:

- > von links nach rechts geht die X-Achse
- > von oben nach unten die Y-Achse

Wir wechseln die Bühne, dann wird das klarer: [ogy.de/03-SCRATCH](https://www.scratch.de/03-SCRATCH)

- > Klick unten Rechts auf das blaue Symbol
- > Tippe in die Suche die Buchstaben **XY** ein
- > Klicke das „Xy-grid“ an. (siehe Bild XY-Bühne)
- > Nun siehst du die Katze auf dem Koordinatensystem

Du siehst, dass das System von X= -240 bis X= 240 und von Y= -180 bis Y= 180 groß ist.

Das probiere wir nun einmal aus. Wir brauchen wieder den Fahnen-Startblock:

Füge den Block **Wenn Fahne angeklickt wird** zum Programmierbereich hinzu.

(Du findest ihn im Bereich **Ereignisse**)

Füge nun den Block **Setze x auf #** darunter. (Siehe Bild Vorlage)
Achtung! Nicht „ändere x um“!

Probiere, wie im Bild, für X=100 aus, was passiert, wenn du auf die grüne Fahne klickst. Probiere außerdem X= -100

Probiere nun eigene Zahlen aus. Klicke jedes mal auf die grüne Fahne.

Lösche nun **Setze x auf #** und ersetze ihn durch den Block **Setze y auf #** darunter.

Probiere für Y=100 aus, was passiert, wenn du auf die **grüne Fahne** klickst. Probiere außerdem Y= -100

Probiere nun wieder eigene Zahlen aus.

Anwendung

Jetzt bist du dran! Versuche folgende Aufgabenstellung zu programmieren, indem du die richtigen Blöcke nacheinander setzt:

- > **Wenn du die grüne Fahne klickst,**
- > dann gleitet die Katze in 1 Sek in die obere linke Ecke.
- > dann schaltet sie den Malstift ein.
- > dann gleitet sie in 1 Sek in die Position X=0 Y=100
- > dann gleitet sie in 1 Sek in die obere rechte Ecke.
- > dann gleitet sie in 1 Sek in die Position X=100 Y=0
- > dann gleitet sie in 1 Sek in die untere rechte Ecke.
- > dann gleitet sie in 1 Sek in die Position X=0 Y=-100
- > dann gleitet sie in 1 Sek in die untere linke Ecke.
- > dann gleitet sie in 1 Sek in die Position X=-100 Y=0
- > dann gleitet die Katze in 1 Sek in die obere linke Ecke.
- > schalte den Stift aus
- > dann gleitet sie in 1 Sek in die Mitte

Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

Musterlösung



Hinweis

Die Koordinaten für deine Ecken können natürlich etwas anders sein. Solange die Ecken erreicht werden, ist alles super!

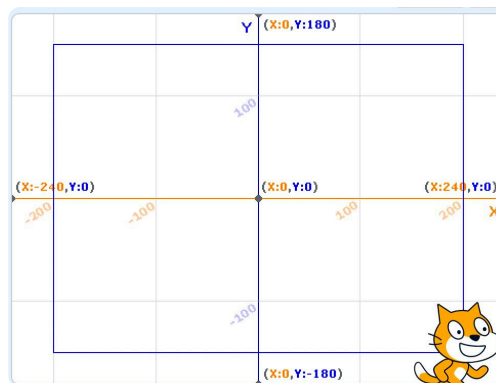
Bewegung üben

④ Jetzt geht es ans Knobeln. Schaffst du es folgende Aufträge zu erfüllen?

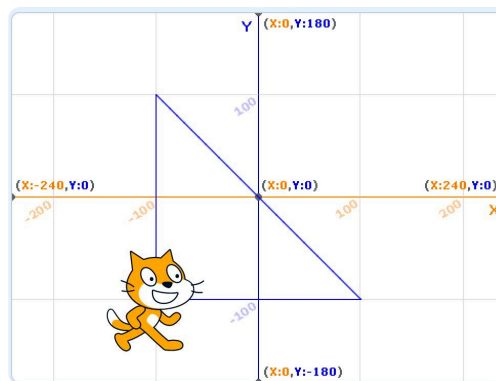
Alle Aufgaben beginnen mit einem Klick auf die grüne Fahne

Vergleiche **anschließend** mit den Musterlösungen unten.

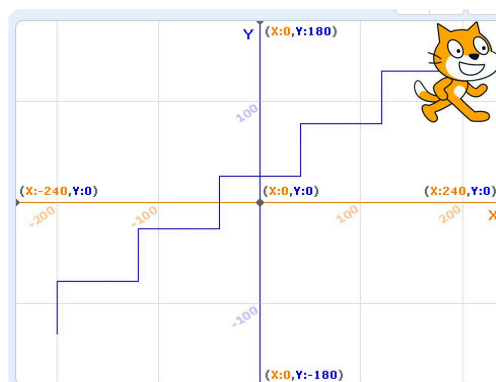
- Die Katze startet in der Ecke unten rechts und zeichnet dann ein Rechteck.
Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name
- Die Katze zeichnet ein rechtwinkliges Dreieck durch den Nullpunkt ($x=0$ & $y=0$).
Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name
- Die Katze startet unten links in der Ecke und zeichnet dann eine Treppe mit fünf Stufen nach oben rechts. Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name



a)



b)

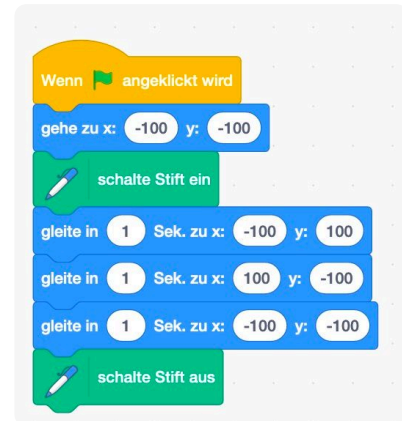


c)

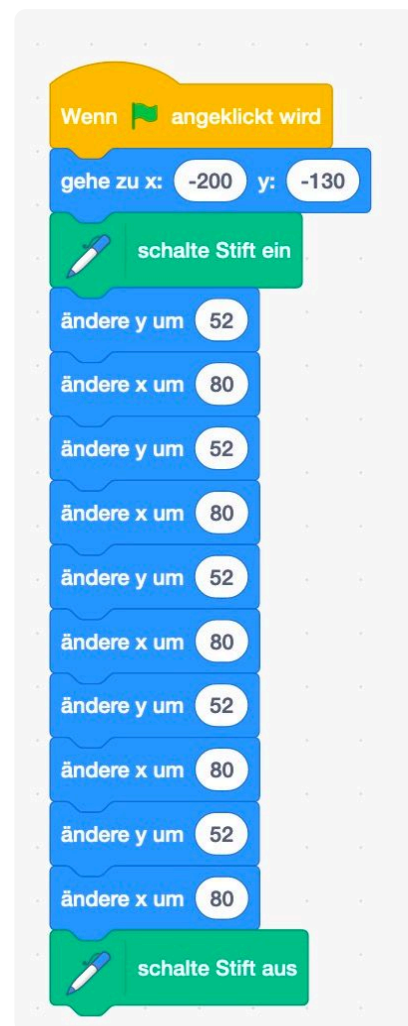
Lösungsvorschläge - deine Lösung kann anders und trotzdem richtig sein!



a)



b)



c)

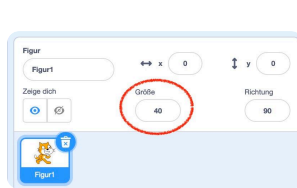
Direkte Katzensteuerung

Starte mit dem Video!



[ogy.de/05-SCRATCH](https://www.youtube.com/watch?v=ogy.de/05-SCRATCH)

- ⑤ Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“.
Erledige die Schritte in genau dieser Reihenfolge!



- 1 Verkleinere die Katze:
Gib im Figurenbereich bei Größe „40“.



- 2 Wir wollen die Katze mit den Pfeiltasten der Tastatur steuern.

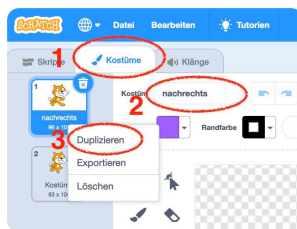
Dafür müssen wir nun mehrere Ereignisse einfügen:

Wenn Taste (Pfeil nach oben) gedrückt wird
Wenn Taste (Pfeil nach unten) gedrückt wird
Wenn Taste (Pfeil nach links) gedrückt wird
Wenn Taste (Pfeil nach rechts) gedrückt wird

- 3 Diese müssen nun einzeln programmiert werden.
Hier 2 Beispiele:

Wenn Taste (Pfeil nach oben) gedrückt wird
ändere y um (10)
Wenn Taste (Pfeil nach links) gedrückt wird
ändere x um (-10)

Ergänze die anderen beiden selbst.

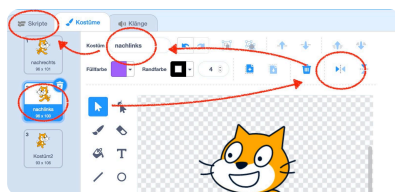


- 4 Teste deine Programmierung, indem du die Pfeiltasten in Scratch drückst.

- 5 Jetzt kann die Katze ganz wundervoll über die Bühne huschen. Allerdings „geht sie rückwärts“, wenn du den Pfeil nach links drückst.

Um dies zu ändern, müssen wir einen „Kostümwechsel“ vornehmen:

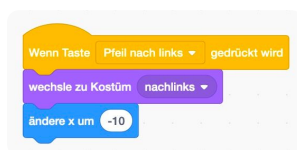
- > Wechsel in den Einstellungen in **Kostüme**
- > Ändere den Namen von *Kostüm1* zu *nachrechts*
- > Rechtsklick auf Kostüm *nachrechts*
- > Bestätige mit Enter
- > Klick *Duplizieren*
- > Wähle „nachrechts2“ aus.
- > Klicke auf „horizontal spiegeln“ (siehe Bild)
- > Ändere den Namen von *nachrechts2* zu *nachlinks*
- > Wechsle in den Einstellungen zu **Skripte**



- 6 Nun soll die Katze die Kostüme anziehen, je nachdem in welche Richtung sie läuft.

Wenn Taste (Pfeil nach links) gedrückt wird
wechsle zu Kostüm (nachlinks)
ändere x um (-10)

Ergänze außerdem den Kostümwechsel nach rechts.



- 7 Teste nun dein Programm und lass die Katze in alle Richtungen laufen.

- 8 Bitte speichere die fertige Aufgabe ab. Dateinamenschema:
JJMMTT-Aufg-Name

⑥ Anwendung:

Die Katze soll sich beim Drücken des Leertaste in die Mitte der Bühne setzen und „Kostüm2“ anziehen.

- Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

Musterlösung



Schnelles Treppenbauen

Starte mit dem Video!

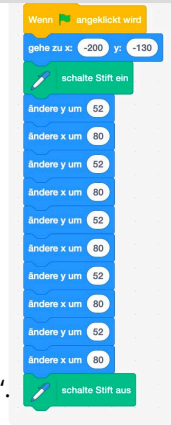


[ogy.de/ST-SCRATCH](https://www.scratch.de/ST-SCRATCH)

Im Abschnitt "Bewegungen üben" solltest du die Katze eine Treppe zeichnen lassen.

Das war aber mit viel Arbeit verbunden. Für jede Stufe musstest du die Katze hoch laufen lassen und dann noch zur Seite. Bei fünf Stufen wird das ganz viel Code.

Sieh ihn dir hier rechts noch einmal an:



Programmierer sind faule Leute und überlegen sich lieber einen schlaun Trick, um sich die Arbeit zu erleichtern. Sie arbeiten mit sogenannten **Schleifen**. Was das bedeutet, wollen wir uns jetzt anschauen.

- ⑦ Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“. Erledige die Schritte in genau dieser Reihenfolge!

- 1 Verkleinere die Katze wieder auf Größe 40.
- 2 Startsignal soll wieder das Klicken auf die grüne Fahne sein.
- 3 Die Katze soll wieder unten Links in der Ecke, Koordinaten x=-215 und y=-155, starten. Füge den korrekten **Bewegungsblock** ein.
- 4 Jetzt benötigen wir wieder den Malstift, die Katze soll ja eine Treppe zeichnen. Aktiviere ggf. wieder die Erweiterung *Malstift* und füge **schalte Stift ein** unter das **Fahnenereignis** ein.
- Tip* Zum Ausprobieren ist es ganz gut, wenn man auf Tastendruck die bisherigen Zeichnungen löschen kann. Programme dafür: **Wenn Taste (Leertaste) gedrückt wird > Lösche alles**
- 5 Die Katze soll 5 Stufen zeichnen. Wir programmieren erstmal eine Stufe. Dafür soll die Katze **5 Punkte nach oben** und **5 Punkte nach rechts** gehen. Füge die beiden Blöcke zunächst frei auf dem Programmierbereich zusammen und verbinde sie erstmal **nicht** mit dem **Fahnenereignis**.
- 6 Auf der orangenen Palette „Steuerung“ findest du mehrere Schleifen zur Auswahl. Ziehe die **Wiederhole (10) mal** Schleife unter das Stift-Einschalten.
- 7 Ändere die Wiederholungszahl. Wir wollen, dass die Katze fünf Stufen zeichnet. Ändere die Zahl also auf 5.
- 8 Ziehe nun die beiden blauen **Bewegungsblöcke** in die Schleife hinein. Füge außerdem den Block **schalte Stift aus** darunter.
- 9 Teste nun dein Programm und schau, ob die Katze die Stufen wieder so zeichnet. Wenn alles geklappt hat, fahre mit dem nächsten Schritt fort.
- 10 Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

💡 Schleifen - Zählschleifen

Eine Schleife verwenden wir, um Wiederholungen im Programm zu realisieren – also um Dinge mehrmals zu machen. Jede Schleife hat eine Durchlauf-Bedingung, damit sie nicht endlos läuft. D.h. vor einem Durchlauf wird eine Bedingung geprüft. Ist die Bedingung erfüllt, werden die Befehle innerhalb der Schleife ausgeführt. Ist die Bedingung nicht (mehr) erfüllt, wird die Schleife verlassen.

<http://www.c-howto.de/tutorial/schleifen/>

In unserem Beispiel benutzen wir eine **Zähl-Schleife**.

Das bedeutet, dass wir angeben, wie oft wir die Sequenz ausführen wollen, der in der Schleife steht. In unserem Beispiel soll die Katze eine Stufe zeichnen. Und das 5 mal. Das sieht dann so aus:

Die Schleife zieht einen klaren Rahmen um die Sequenz, die wiederholt werden soll.



⑧ Anwendung >>> Benutze bei jeder Aufgabe Zählschleifen!

- a) **Verändere deine Programmierung so, dass die Stufen bis ganz nach oben rechts gehen.**

Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

- b) > Erstelle eine neue, leere Oberfläche (Datei > Neu)

> **Lass deine Katze beim Klick auf die grüne Fahne vom Punkt X=0 & Y=0 ein Quadrat mit Seitenlänge 100 zeichnen.**

>> *Tip: Bei einem Quadrat sind alle Seitenlängen gleich, jede Ecke hat 90°*

> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

- c) > Erstelle eine neue, leere Oberfläche (Datei > Neu)

> **Lass deine Katze beim Klick auf die grüne Fahne eine gestrichelte Linie von links (X=-200 & Y=0) nach rechts zeichnen.**

>> *Tip: der Strich bzw. der Abstand soll 10 Schritte betragen*

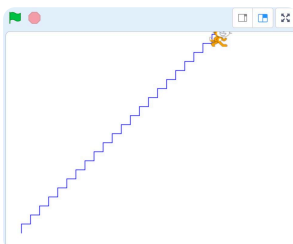
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

- d) > Erstelle eine neue, leere Oberfläche (Datei > Neu)

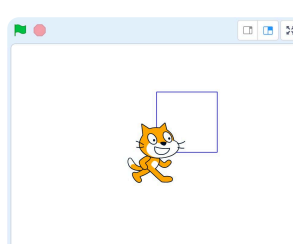
> **Lass deine Katze beim Klick auf die grüne Fahne vom Punkt X=-50 & Y=-50 ein Quadrat, Seitenlänge 200, mit einer gestrichelte Linie zeichnen.**

>> *Tip: Versuche die Teilaufgaben b und c miteinander zu kombinieren.*

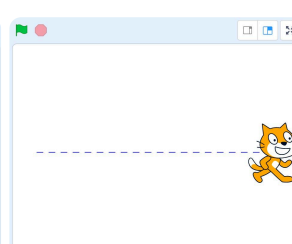
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name



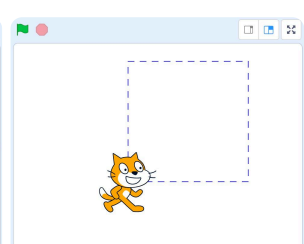
a)



b)



c)



d)

Musterlösung - Zählschleifen

Wenn angeklickt wird

gehe zu x: -215 y: -155

schalte Stift ein

wiederhole 22 mal

ändere y um 15

ändere x um 15

schalte Stift aus

a)

Wenn angeklickt wird

gehe zu x: 0 y: 0

schalte Stift ein

wiederhole 4 mal

gehe 100 er Schritt

drehe dich um 90 Grad

b)

Wenn angeklickt wird

gehe zu x: -200 y: 0

wiederhole 20 mal

gehe 10 er Schritt

schalte Stift ein

gehe 10 er Schritt

schalte Stift aus

c)

Wenn angeklickt wird

gehe zu x: -50 y: -50

wiederhole 4 mal

wiederhole 10 mal

gehe 10 er Schritt

schalte Stift ein

gehe 10 er Schritt

schalte Stift aus

drehe dich um 90 Grad

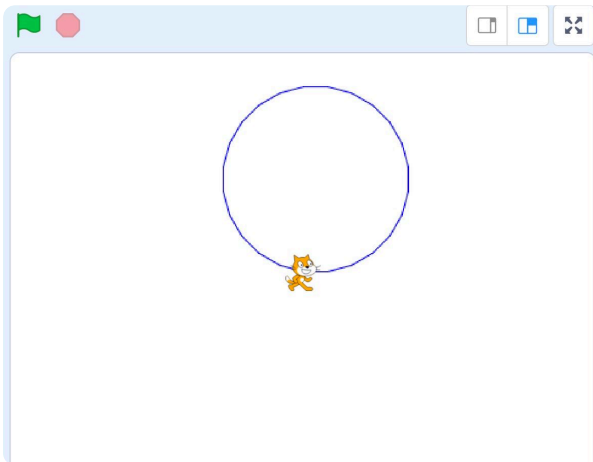
schalte Stift aus

d)

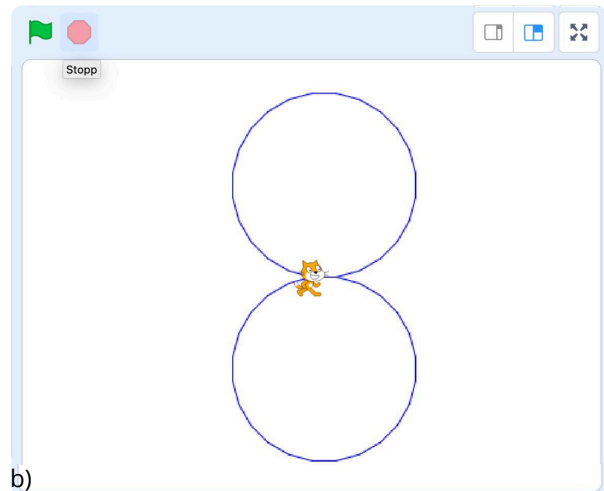
Zählschleife: Weitere Übungen

⑨ **Anwendung** >>> Benutze bei jeder Aufgabe Zählschleifen!

- a) > Erstelle eine neue, leere Oberfläche (Datei > Neu)
> **Lass deine Katze beim Klick auf die grüne Fahne vom Punkt X=0 & Y=0 einen Kreis zeichnen.**
>> *Tip: 15° Drehungen reichen aus.*
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name
-
- b) > Erstelle eine neue, leere Oberfläche (Datei > Neu)
> **Lass deine Katze beim Klick auf die grüne Fahne vom Punkt X=0 & Y=0 eine 8 zeichnen.**
>> *Tip: Orientiere dich an der Aufgabe vorher!*
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name
-
- c) > Erstelle eine neue, leere Oberfläche (Datei > Neu)
> **Lass deine Katze beim Klick auf die grüne Fahne 10 Quadrate (Seitenlänge 20) von links (X=-200 & Y=0) nach rechts zeichnen.**
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name
-
- d) > Erstelle eine neue, leere Oberfläche (Datei > Neu)
> **Lass deine Katze beim Klick auf die grüne Fahne 100 Quadrate in 10er Reihen (Seitenlänge 20) von links (X=-200 & Y=-100) nach rechts zeichnen.**
>> *Tip: Orientiere dich an der Aufgabe vorher. Nutze 4 (!) Zählschleifen!*
> Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

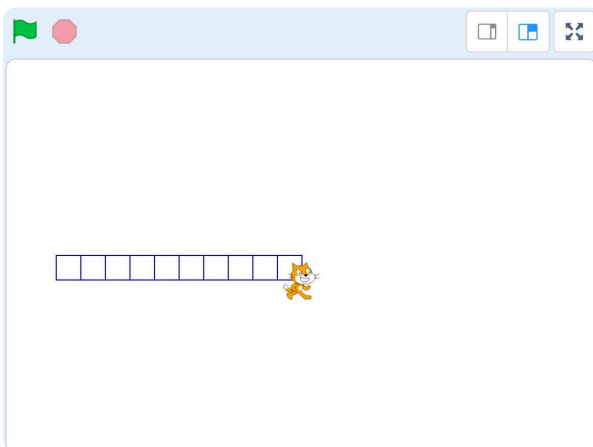


a)

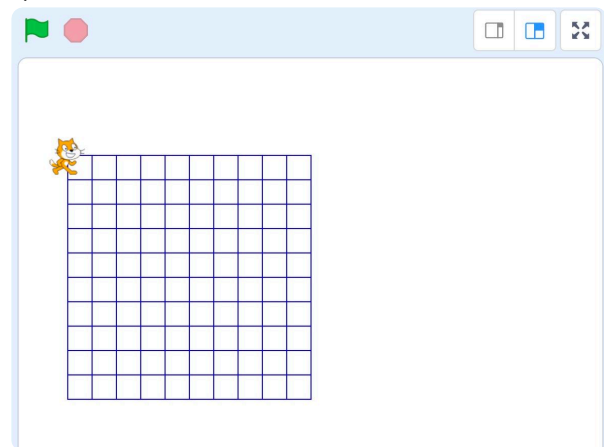


b)

a)



c)



d)

Musterlösung Zählschleife

Ein Kreis hat 360° , also $\Rightarrow 360^\circ$ durch die Drehungsgradzahl = Wiederholungsanzahl

Schrittgröße ist flexibel!

```

    Wenn angeklickt wird
      schalte Stift ein
      wiederhole 24 mal
        gehe 20 er Schritt
        drehe dich um 15 Grad
      schalte Stift aus
  
```

a)

```

    Wenn angeklickt wird
      schalte Stift ein
      wiederhole 24 mal
        gehe 20 er Schritt
        drehe dich um 15 Grad
      wiederhole 24 mal
        gehe 20 er Schritt
        drehe dich um 15 Grad
      schalte Stift aus
  
```

b)

```

    Wenn angeklickt wird
      gleite in 1 Sek. zu x: -200 y: 0
      schalte Stift ein
      wiederhole 10 mal
        wiederhole 4 mal
          gehe 20 er Schritt
          drehe dich um 90 Grad
        gehe 20 er Schritt
      schalte Stift aus
  
```

c)

```

    Wenn angeklickt wird
      gleite in 1 Sek. zu x: -200 y: -100
      schalte Stift ein
      wiederhole 10 mal
        wiederhole 4 mal
          gehe 20 er Schritt
          drehe dich um 90 Grad
        gehe 20 er Schritt
        drehe dich um 180 Grad
        wiederhole 10 mal
          gehe 20 er Schritt
          drehe dich um 180 Grad
          drehe dich um 90 Grad
          gehe 20 er Schritt
          drehe dich um 90 Grad
        schalte Stift aus
  
```

```

    Wenn Taste Leertaste gedrückt wird
      schalte Stift aus
      lösche alles
      gleite in 1 Sek. zu x: 0 y: 0
      setze Richtung auf 90 Grad
  
```

d)

„The stage is yours!“ - Meine eigene Bühne

Starte mit dem Video!



Bisher sind wir entweder auf weißem Hintergrund oder im Koordinatensystem unterwegs gewesen. Das ist doch etwas trist. Deshalb wollen wir uns jetzt die Umgebung verschönern! Dazu können wir unsere eigene Bühne designen.

Das Ziel ist, eine Straße durchs Grüne zu haben, damit dort Autos entlang fahren können.

[ogy.de/EB-SCRATCH](https://www.scratch.de/EB-SCRATCH)

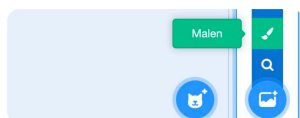
Im Anschluss können wir dann damit weiter arbeiten. Viel Spaß!

- ⑩ Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“.
Erledige die Schritte in genau dieser Reihenfolge!



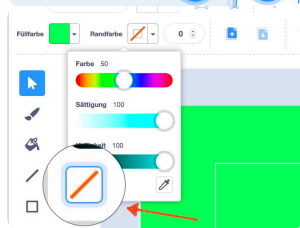
1

Schalte die Katze aus. Klicke dazu im **Figurenbereich** erst die Katze an und dann bei „Zeige dich“ auf das durchgestrichene Auge.



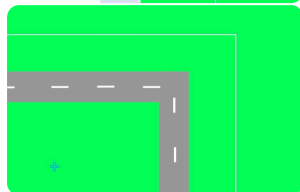
2

Bewege deine Maus im Figurenbereich unten rechts auf das Bühnensymbol. Dort öffnet sich dann automatisch ein kleines Menü. Klicke auf **Malen**.



3

Jetzt bist du im Bühnenbildeditor. Benutze die angebotenen Werkzeuge (Vermutlich bietet dir das **Rechteck** eine schnelle Hilfe) und zeichne einen großes grünes Rechteck als Hintergrund. **Achtung:** den Rahmen solltest du vor der Zeichnung auf **transparent** setzen. (siehe Bild links)



4

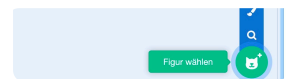
Benutze nun weitere graue Rechtecke, um eine O-förmige Straße zu zeichnen. Wenn du möchtest kannst du mit weißen Rechtecken auch Fahrbahnmarkierungen hinzufügen.

Tip

Grau erhältst du, wenn du in der Farbauswahl die Sättigung auf 0 setzt.

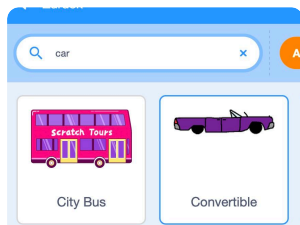
Tip

Rechts oben auf der Bühne, siehst du, wie deine Zeichnung dann später „fertig“ aussieht.



5

Nun fügen wir noch ein Auto hinzu! Klicke dazu im Figurenbereich auf **Figur wählen**. Gib in die Suche das Wort **car** ein und klicke auf das lilane Cabrio. Nun ist es hinzugefügt.



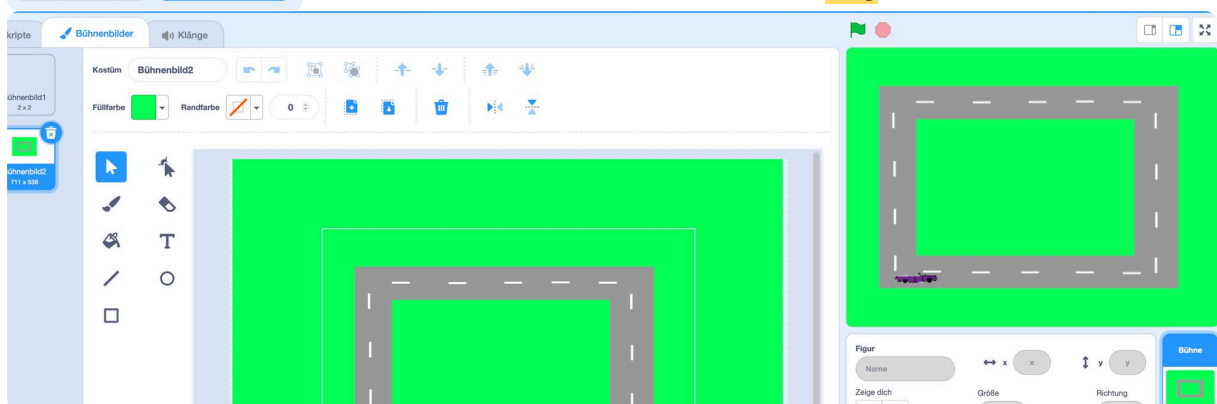
6

Ändere die Größe des Autos auf 30 und setze es auf die Straße.

7

Bitte speichere die fertige Aufgabe ab. Diesmal schreiben wir noch „Straßenkarte“ dazu.

Dateinamenschema: JJMMTT-Aufg-Name-Straßenkarte



Wir fahren hin und her. Automatisch!

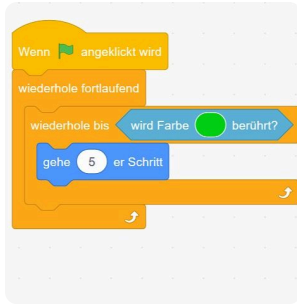
Starte mit dem Video!



Jetzt fährt das Auto wunderbar bis zum Rand. Als nächstes wäre es sinnvoll, wenn es dann umkehren würde und wieder zurück fährt. Bis die Wiese kommt. Und immer so weiter! Hin **und her**. Ganz automatisch, immer weiter *fortlaufend*.

[ogy.de/H2-SCRATCH](https://www.scratch.de/H2-SCRATCH)

- ⑬ Du brauchst deine Programmieroberfläche **aus der Übung Wir fahren hin**.
Erledige die Schritte in genau dieser Reihenfolge!



1 Wähle das Auto im **Figurenbereich** aus. Nun kannst du es programmieren.

2 Löse den Block "**Wiederholen bis Farbe berührt wird**" vom Startereignis **grüne Fahne**.

3 Füge unter die **grüne Fahne** aus **Steuerung** den Block **wiederhole fortlaufend** ein.

4 In die **fortlaufende Schleife** fügst du die **Bedingungsschleife** aus Schritt 2 ein.



6 Aktuell berührt das Auto ja die Wiese, also lassen wir es zurücksetzen mit dem Block **gehe -10er Schritt**

5 Jetzt müssen wir das Auto umkehren lassen. Dazu muss das Auto um 180° nach links drehen. Füge den Block **dreh dich links um 180 Grad** ein.

7 Setze das Auto unten links in die Ecke und probiere es aus. Du wirst feststellen, dass das Auto überhaupt nicht mehr aufhört hin und her zu fahren. Das liegt an der „unendlichen“ Schleife **wiederhole fortlaufend**

8 Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

💡 Schleifen - Endlosschleifen

Neben der Zählschleife und der Bedingungsschleife kennst du nun auch eine **Endlosschleife**. Diese lässt die Programmierung in der Schleife immer wieder ablaufen, bis das Programm beendet wird oder etwas anderes geschieht.



- ⑭ Erstelle für Scratch dieses Skript. Klicke dann die grüne Flagge an und bewege den Mauszeiger über die Bühne. Was passiert?

- ⑮ Ergänze den Lückentext

Der Wiederhole-Fortlaufend-Befehl sorgt dafür, dass die von ihm umschlossene Befehlsfolge immer wieder und ohne ausgeführt wird.

Um den Wiederhole-Fortlaufend-Befehl zu beenden muss man auf das klicken.

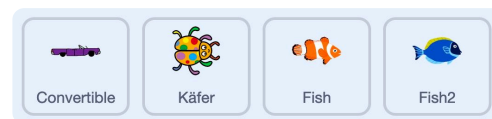
- 16) Bearbeite die Programmierung so, dass das Auto nicht nur hin und her fährt, sondern unendlich viele komplette Runden im Viereck fährt.

Tipp: das Auto muss wahrscheinlich auf der linken Fahrbahn fahren, quasi als Geisterfahrer, da es sehr lang ist.



- 17) Weitere Aufgaben:

- Ändere deine Bühne so, dass ein großer blauer See in der Mitte der Straße ist.
- Lass zwei Fische darin schwimmen, sie sollen von links nach rechts immer weiter ihre Bahnen ziehen.
Extra-knifflig: Schaffst du es, dass der Fisch die Kostüme so wechselt, dass er nicht „auf dem Kopf“ nach links schwimmt?
- Lass einen Käfer außen auf dem grünen Bereich kreuz und quer herumlaufen. (Suchbegriff: Bug)
Tipp: Benutze einen dünnen äußeren Rahmen, der ebenfalls die Farbe der Straße hat.



a)

Ein Auto mit Gaspedal

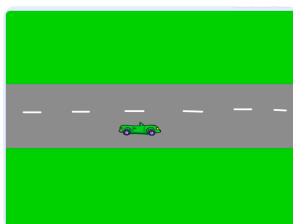
Starte mit dem Video!



ogy.de/GP-SCRATCH

Nun wollen wir das Auto ein bisschen besser steuern. Bisher fährt es immer in derselben Geschwindigkeit. Jedes Mal, wenn die Schleife durchlaufen wird, geht es um 5 Schritte nach vorne. Das können wir variieren, also verändern.

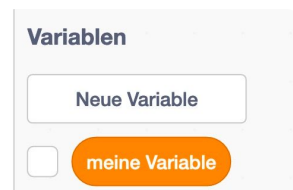
- ⑱ Du brauchst eine neue Programmieroberfläche: Klicke „Datei“ > „Neu“.
Erledige die Schritte in genau dieser Reihenfolge!



1

Erstelle eine neue Bühne. Diesmal wollen wir keine Ringstraße, sondern eine Straße von links nach rechts, die durchs Grüne führt.

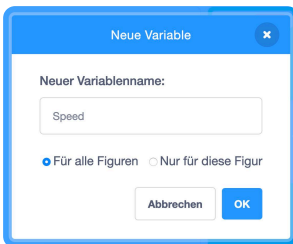
Füge außerdem wieder ein Auto ein (Größe 50) und schalte die Katze unsichtbar.



2

Nun brauchst du einen Code, der das Auto beim Klick auf die **grüne Fahne** von links nach rechts - und wieder zurück - fahren lässt. Denke auch an die Kostüme, damit das Auto nicht „auf dem Kopf“ zurück fährt.

Benutze hier den Block **<wird (Rand) berührt?>**. Orientiere dich ansonsten an dem Code von *Wir fahren hin und her*.

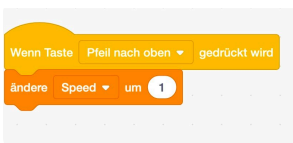


3

In der Palette **Variablen** findest du den benötigten Baustein, um die Geschwindigkeit zu verändern. Klicke auf **Neue Variable**, gib ihr den Namen **Speed** und lasse den Punkt bei **Für alle Figuren**. Klicke auf **OK**. Jetzt ist die Variable **deklariert**.

Tipp

Oben links auf der Bühne erscheint nun die Variable mit aktuellem Wert. Sie beginnt immer mit 0.



4

Wir wollen mit einem Tastendruck auf **↑** den Wert von **Speed** steigen, und mit **↓** fallen lassen.

Ziehe dazu aus der Palette **Ereignisse** den Block **Wenn Taste (Leertaste) gedrückt wird** in den Programmierbereich und verändere **Leertaste** in **Pfeil nach oben**.

Aus der Palette **Variablen** fügst du den Block **ändere (meine Variable) um (1)** darunter. Ändere **meine Variable** in **Speed** um.



5

Erledige Schritt 4 noch einmal, aber für **Pfeil nach unten**. Denke daran, dass sich die Geschwindigkeit verringern soll.

6

Nun müssen wir noch den Code des Autos ändern. In der Bedingungsschleife „geht“ das Auto bisher immer einen **5er Schritt**. Die Zahl (5) ersetzen wir durch die Variable **Speed**. Ziehe diese aus der Palette **Variablen** auf die Zahl.

Setze außerdem beim Klick auf **grüne Fahne** **Speed** auf 0.

Tipp

Einfacher ist es, das Auto immer an der gleichen Position starten zu lassen (beispielsweise an Position X=0 und Y=0) und mit dem Kostüm zum nach rechts fahren.

7

Starte dein Programm mit dem Klick auf die **grüne Fahne** und teste deine beiden Pfeiltasten aus.

8

Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name

- ⑲ Schaffst du es, den Code so zu ändern, dass du schneller als „13“ fahren kannst?

Variablen

Code	Um im Code „flexibel“ zu bleiben, benutzt man in der Informatik Variablen .
Inhalt	In einer Variable kann man einen Wert speichern. Das kann bspw. eine Zahl oder ein Text sein.
<i>Im Beispiel</i>	<i>In unserem Beispiel können wir uns damit den Wert für die Geschwindigkeit merken.</i>
änderbar	Ein anderes deutsches Wort für <i>variabel</i> ist <i>änderbar</i> . Eine <i>Variable</i> ist also ein änderbarer Wert .
<i>Im Beispiel</i>	<i>In unserem Beispiel können wir die Variable Speed aber auch ändern. Die Änderungen bleiben dann gespeichert.</i>
Deklaration	Wenn man eine Variable neu anlegt, nennt man dies das Deklarieren einer Variable.
Initialisierung	Wenn eine Variable zum ersten Mal einen Wert erhält, nennt man dies die Initialisierung der Variable. Man sagt auch: <i>Die Variable wird initialisiert</i> .

20 Setze die richtigen Wörter ein:

deklarieren 1x geändert 1x geänderten 1x initialisieren 1x Speed 1x
 Speicherplatz 1x Text 1x verwenden 1x Zahlen 1x

Wir benutzen Variablen, um Werte (wie oder) unter einem eindeutigen, klaren

Namen (in unserem Beispiel ist der Name) zu speichern und später wieder zu

.

Aus der Sicht eines Computers, ist eine Variable ein mit einem bestimmten

Namen.

Variablen erlauben zusätzlich, dass die Werte während der Ausführung des Programms

werden können und das Programm dann mit den Werten weiter

läuft.

Man kann beim Programmieren mehrere verschiedene Variablen erstellen. Das Fachausdruck dafür

lautet „Variablen “

Im Programm muss man der neuen Variablen dann auch einen Startwert zuweisen. Dies nennt man in

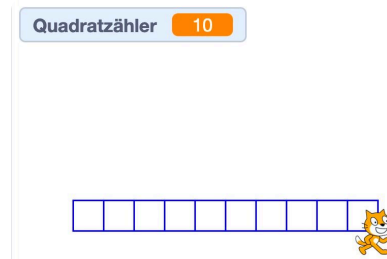
der Informatik „die Variable “.

Weitere Übungen zu Variablen 1

21) Jetzt kennst du einiges Werkzeug, um deine Programmierungen noch zu überarbeiten. In Aufgabe 9c hast du eine Reihe mit Quadraten zeichnen lassen. Lade die Datei und arbeite mit Variablen:

- a) Deklare die Variable „Quadratähler“, die bei jedem neuen Quadrat um 1 erhöht wird.
- b) Erstelle Code, so dass mit der Leertaste die bisherige Zeichnung gelöscht wird und der Quadratähler auf „0“ zurückgesetzt wird.
- c) Bitte speichere die fertige Aufgabe ab.

Dateinamenschema: JJMMTT-Aufg-Name



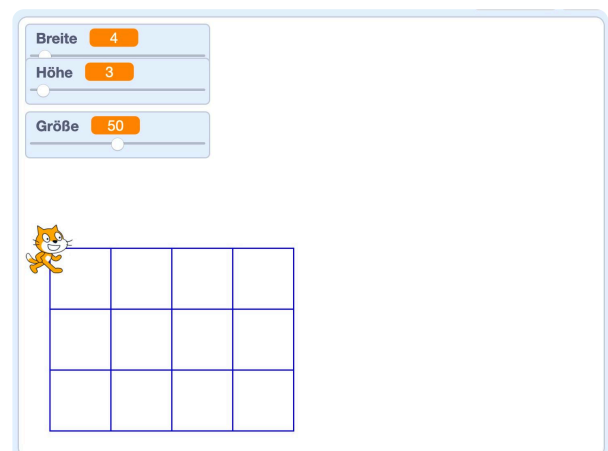
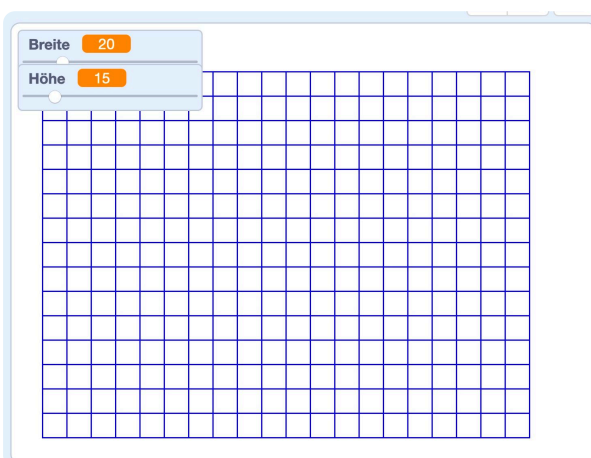
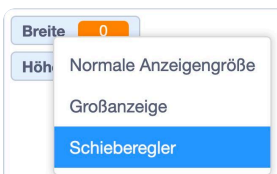
a)



Ursprung

22) Lade die Datei aus Aufgabe 9d. Zum Abgleichen links nochmals die Musterlösung. Diese muss überarbeitet werden. Achtung: Ich gehe von der Musterlösung aus!

- a) Beim Ausprobieren, ist es gut, schnell „reset“ drücken zu können. Erstelle Code, der bei Druck auf die Leertaste gestartet wird:
 - > Stift wird ausgeschaltet
 - > Alle Zeichnungen des Stifts werden gelöscht
 - > Setze die Katze auf X=0 und Y=0
 - > Setze Richtung auf 90 Grad
- b) Es ist doch sehr schade, dass der Code nur „langweilig“ immer hundert quadrate zeichnet. Das sollte flexibler werden. Mit Hilfe von Variablen ist dies kein Problem. Deklare die beiden Variablen „Höhe“ und „Breite“.
- c) Mache je einen Rechtsklick auf die beiden Variablen in der Bühne und wähle aus: **Schieberegler**.
- d) Setze nun die Variable an den richtigen Stellen ein, sodass du mit den beiden Schieberegler die Höhe und die Breite der Kästchenzeichnung einstellen kannst.
- e) Deklare eine weitere Variable „Größe“. Nutze sie, um die Größe der Kästchen mit einem Schieberegler einstellen zu können.
- f) Bitte speichere die fertige Aufgabe ab. Dateinamenschema: JJMMTT-Aufg-Name



Weitere Übungen zu Variablen 1

Scratch code blocks for a square drawing exercise:

- Wenn **angeklickt** wird
- gleite in **1** Sek. zu x: **-200** y: **0**
- schalte Stift ein
- wiederhole **10** mal
 - wiederhole **4** mal
 - gehe **20** er Schritt
 - drehe dich um **90** Grad
 - ändere **Quadratähler** um **1**
 - gehe **20** er Schritt
- schalte Stift aus

Additional code blocks on the right:

- Wenn Taste **Leertaste** gedrückt wird
- lösche alles
- setze **Quadratähler** auf **0**

19

Scratch code blocks for a square drawing exercise using variables:

- Wenn **angeklickt** wird
- gleite in **1** Sek. zu x: **-215** y: **-160**
- schalte Stift ein
- wiederhole **Höhe** mal
 - wiederhole **Breite** mal
 - wiederhole **4** mal
 - gehe **Größe** er Schritt
 - drehe dich um **90** Grad
 - gehe **Größe** er Schritt
 - drehe dich um **180** Grad
 - wiederhole **Breite** mal
 - gehe **Größe** er Schritt
 - drehe dich um **180** Grad
 - drehe dich um **90** Grad
 - gehe **Größe** er Schritt
 - drehe dich um **90** Grad
- schalte Stift aus

Additional code blocks on the right:

- Wenn Taste **Leertaste** gedrückt wird
- schalte Stift aus
- lösche alles
- gleite in **1** Sek. zu x: **0** y: **0**
- setze Richtung auf **90** Grad

20

Ein Katzenrechner kann nicht nur addieren!


Starte mit dem Video!



Ein Taschenrechner, der nur addieren kann? Das taugt nicht viel. Also lass ihn uns verbessern:

- 25 Du brauchst deine Programmieroberfläche **aus der Übung Der Katzenrechner**.
Erledige die Schritte in genau dieser Reihenfolge!

[ogy.de/KR-SCRATCH](https://www.tutory.de/KR-SCRATCH)

- 1 **Wir bearbeiten die Variablen:**
> Deklariere die Variable *Operation*.
> Benenne die Variable *Zahl3* um in *Lösung* mit einem Rechtsklick in der Palette **Variablen**.
- 2 Initialisiere alle vier Variablen mit dem Wert 0.
Ändere außerdem die Ausspruch „Wir addieren“ um in „Wir rechnen!“
- 3 Nun benötigen wir noch eine weitere Abfrage, nämlich nach der Rechenoperation. Lass die Katze also fragen
„Wie möchtest du rechnen? + - /,*“
Der „/“ steht für „geteilt“. Setze die Variable *Operation* auf die Antwort.
Diese Abfrage sollte nach der Abfrage für *Zahl1* erfolgen.
- 4 Nun müssen wir eine Verzweigung starten. In der Palette **Steuerung** findest du den Block **falls <>, dann ...**
Füge ihn ein.
In die Bedingung, also in das Feld <>, setzt du aus der Palette **Operatoren** den Block 
Setze in das erste Feld die Variable *Operation* und in das zweite das Zeichen +.
Vergleiche mit dem Bild links.
- Tip* Du musst wahrscheinlich noch den Block in dem das Ergebnis gesagt wird ans Ende schieben. Achtung: Innerhalb der Endlosschleife!
- 5 Erledige Schritt 4 noch einmal, aber für die restlichen drei Rechenoperationen. Setze sie jeweils untereinander.
- 6 Nun wollen wir noch die Verkündung des Ergebnisses verändern:
Die Katze soll sagen:
„Aufgabe und Ergebnis lauten“ und anschließend die Variablen *Zahl1*, *Operator*, *Zahl3* zeigen. Darauf folgt ein „=“ und die Variable *Lösung*.
- 7 Starte dein Programm mit der Leertaste und teste deinen Taschenrechner!
- 8 Bitte speichere die fertige Aufgabe ab. Dateinamenschema:
JJMMTT-Aufg-Name



Verzweigungen

Bisher ist immer brav eins nach dem anderen passiert. Aber beim Katzenrechner würde dies wenig Sinn ergeben:

Wenn die Katze nicht weiß, ob sie addieren, subtrahieren, multiplizieren oder dividieren soll, und einfach alle Rechenoperationen ausführt, dann braucht man sie auch gar nicht erst fragen.

Die Katze muss also wissen, welchen "Rechenweg" sie einschlagen soll.

Den Weg der Addition oder den Weg der Subtraktion oder ... oder ... Die Auswahl des Weges gibt **das Rechensymbol vor, welches der Nutzer eingeben muss.**

Dort stehen zwar 4 Zeilen Code, **es wird aber immer nur eine Zeile Code ausgeführt.**

Der Code, der Weg "verzweigt" sich an dieser Stelle und die Katze geht nur den Weg, der zu der Bedingung passt.



Verzweigungen für Rechenoperationen

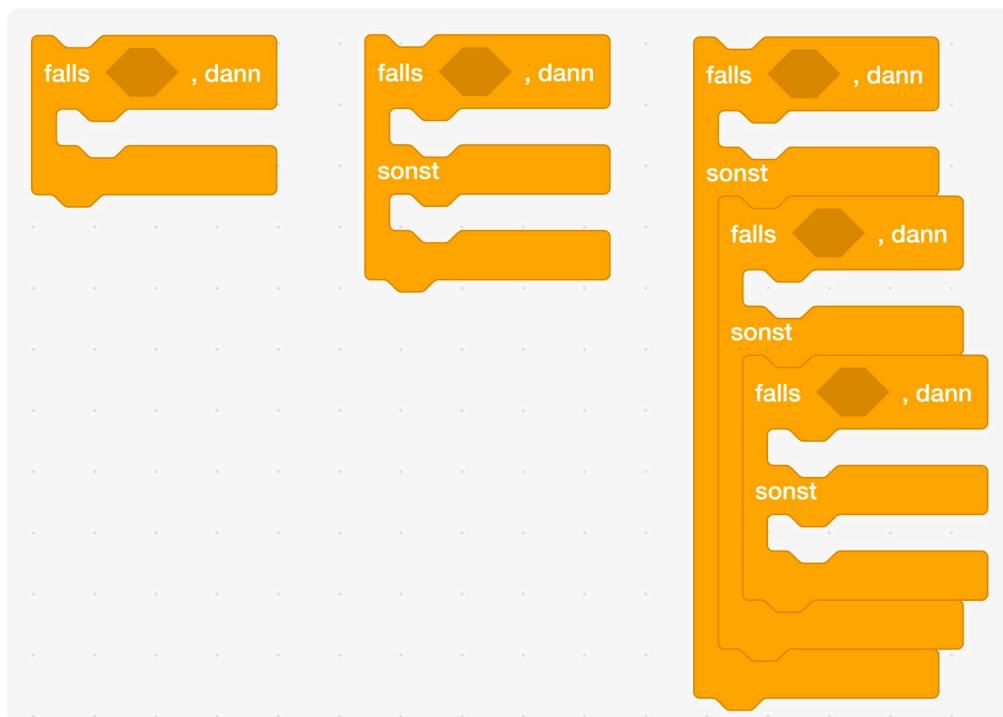
Das ist wie im echten Leben:

- Falls Schnee fällt > zieh einen Mantel an.
- Falls Regen fällt > nimm einen Regenschirm mit.
- Falls die Sonne scheint > benutze Sonnencreme.
- Falls dein Zimmer unordentlich ist > räume es auf.
- usw.



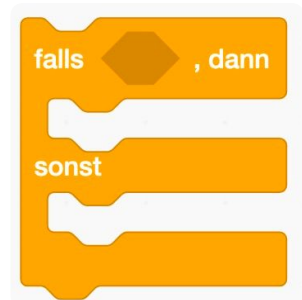
Verzweigungen - IFTTT

sind Programmcode, der ausgeführt wird, wenn eine bestimmte Bedingung erfüllt ist. Man sagt auf "Falls ... dann ..." bzw. "Falls dies dann das" - Verknüpfung dazu. (In Englisch: If ... then ... oder "*If this then that*").



Übungen zu Verzweigungen

- ②6) Bestimmt hast du bereits ausprobiert, was passiert, wenn man der Katze kein Rechensymbol sondern bspw. einen Buchstaben vorgibt. Die Katze macht so weiter, wie sie kann. Sie hat nämlich nur für die vier Rechenoperationen einen Code, den sie ausführen kann. Bei allen anderen Eingaben gibt sie am Ende einfach den Initialisierungswert von „Lösung“ aus. Dies kann man mit dem nebenstehenden Block lösen. Ersetze im Katzenrechnercode die „Falls“-Blöcke durch den nebenstehenden „falls - sonst“-Block. Füge unter „sonst“ den nachfolgenden Code ein. Am Ende sollte die Katze sagen „Keine gültige Rechenoperation. Bitte ignorieren Sie die Ergebnisanzeige!“.



Keine gültige Rechenoperation. Bitte ignorieren Sie die Ergebnisanzeige!



- ②7) Erstelle ein neues Programm und nutze Variablen und Verzweigungen: Die Katze soll sich mit dir unterhalten, Fragen stellen und darauf antworten. Sei kreativ, in dem was sie fragt und antwortet. Am besten stellst du „oder-Fragen“ bzw. „ja oder nein-Fragen“. Die Katze soll aber mindestens die vorgegebenen Fragen stellen. Gib der Mensch eine unbekannte Antwort ein, soll die Katze sagen „Tut mir leid, ich verstehe nicht.“ Du musst wieder die „falls-sonst-Blöcke“ ineinander verschachteln.
- a) Geht es dir heute gut?
(falls ja: „Das freut mich!“;
falls nein: „Das tut mir leid!“)
- b) Magst du lieber Orangensaft oder Apfelsaft?
(falls O: „Wusstest du, dass die Menschen seit mindest 4.500 Jahren Orangen essen?“;
falls A: „Wusstest du, dass Apfelsaft zu 10% aus Zucker besteht? Cola nur aus 9%!“)

- ②8) Überarbeite die Aufgabe zum **Auto mit Gaspedal**

Nutze eine weitere Variable, die die Schrittweite „zurück vom Grünen“ festlegt. Lass diese Variable mit Hilfe von $mal -1$ ($*-1$) berechnen. So kannst du es schaffen, dass jede Geschwindigkeit möglich ist.

Notizen:

Musterlösung

The code is a Scratch script for a calculator. It starts with a 'falls' (if) block where 'Operation' is checked against '+'. If true, it sets 'Lösung' to 'Zahl1 + Zahl2'. If false, it checks for '-', then '*', and finally '/'. If none of these operations are selected, it says 'Keine gültige Rechenoperation. Bitte ignorieren Sie die Ergebnisanzzeige!' for 4 seconds. Finally, it says 'verbinde Aufgabe und Ergebnis lauten: ' followed by 'Zahl1', 'Operation', and 'Zahl2'.

The code is a Scratch script for a character interaction. It starts with a 'Wenn angeklickt wird' (When clicked) event. It then enters a 'wiederhole fortlaufend' (Repeat) loop. Inside the loop, it sets 'Gemüt' to 0 and 'Saft' to 0. It says 'Hallo!' for 2 seconds, asks 'Geht es dir heute gut?' and waits for an answer. It sets 'Gemüt' to the answer. If 'Gemüt' is 'ja', it says 'Das freut mich!' for 2 seconds. If 'Gemüt' is 'nein', it says 'Das tut mir leid!' for 2 seconds. Otherwise, it says 'Tut mir leid, ich verstehe nicht...' for 2 seconds. It then asks 'Magst du lieber Orangensaft oder Apfelsaft?' and waits for an answer. It sets 'Saft' to the answer. If 'Saft' is 'Orangensaft', it says 'Wusstest du, dass die Menschen seit mindest 4.500 Jahren Orangen essen?' for 2 seconds. If 'Saft' is 'Apfelsaft', it says 'Wusstest du, dass Apfelsaft zu 10% aus Zucker besteht? Cola nur aus 9%!' for 2 seconds. Otherwise, it says 'Tut mir leid, ich verstehe nicht...' for 2 seconds. Finally, it says 'Ciao - bis zum nächsten mal!' for 2 seconds.

Exkurs - Aussagenlogik

Sieh dir das Video rechts an. Aussagenlogik ist nicht so ganz einfach, schau es dir einfach ein paar Mal an - du wirst den Dreh sicher herausbekommen.

Erledige außerdem die Scratch-Aufgabe am Ende. Wenn du in dem Thema sicher bist, dann erledige die Aufgaben unten.



[ogy.de/aussagenlogik](https://www.tutary.de/aussagenlogik)

29) Fragen zum Video:

a) Aussagen können eine von zwei Eigenschaften haben. Welche beiden Eigenschaften sind das?

b) Kann sich die Eigenschaft einer Aussage ändern?

c) Zeichne das Symbol der Negation und nenne ein Synonym („ein anderes Wort für“) **Negation**.

d) Nenne beiden Möglichkeiten zur Verknüpfung bzw. Kombination von Aussagen und zeichne die jeweiligen Symbole.

e) Ergänze folgende Wahrheitstafeln.

Aussage 1: Das Kleid ist gelb.	Aussage 2: Die Kinder tanzen.	Aussage 1 \wedge Aussage 2
0 (falsch)	0 (falsch)	
	1 (richtig)	0 (falsch)
1 (richtig)	0 (falsch)	
		1 (richtig)



Nenne die Aussagenkombination, die für das Bild stimmt:

Aussage 1: Der Hase frisst.	Aussage 2: Der Baum hat Blätter.	Aussage 1 \vee Aussage 2
0 (falsch)	0 (falsch)	
0 (falsch)	1 (richtig)	
1 (richtig)	0 (falsch)	
		1 (richtig)



Übungen - Aussagenlogik 1



30) Sieh dir das Bild an und lies dir folgende Aussagen durch. Überprüfe, ob sie wahr (1) oder falsch (0) sind.

- a) Die Brille liegt auf dem Buch.
- b) Der Strohalm steckt in der Kokosnuss.
- c) Die Sonnenbrille liegt unter der Jeans.
- d) Der Schal ist rot, blau und weiß.
- e) Die Kamera liegt auf dem Tisch.
- f) Der Hut liegt auf der Kamera.
- g) Die Kokosnuss ist rot.
- h) Im Buch ist ein Bild abgedruckt.

31) Betrachte das Bild und die Aussagen a-h oben. Sind folgende Aussagen wahr (1) oder falsch (0)?

- | | | | | | | | |
|-------------|--------------------------|-------------|--------------------------|-------------|--------------------------|-------------|--------------------------|
| 1) $\neg a$ | <input type="checkbox"/> | 3) $\neg c$ | <input type="checkbox"/> | 5) $\neg e$ | <input type="checkbox"/> | 7) $\neg g$ | <input type="checkbox"/> |
| 2) $\neg b$ | <input type="checkbox"/> | 4) $\neg d$ | <input type="checkbox"/> | 6) $\neg f$ | <input type="checkbox"/> | 8) $\neg h$ | <input type="checkbox"/> |

32) Formuliere folgende Aussagen:

- 1) $\neg a$
- 2) $\neg c$
- 3) $\neg d$
- 4) $\neg e$

33) Formuliere folgende Aussagen:

- 1) $a \wedge d$ 2) $b \wedge g$ 3) $\neg c \wedge e$ 4) $\neg d \vee f$ 5) $e \vee c$ 6) $\neg f \wedge \neg a$

34) Betrachte das Bild und entscheide, ob die Aussagen der vorherigen Aufgabe wahr (1) oder falsch (0) sind.

- 1) 2) 3) 4) 5) 6)

Übungen - Aussagenlogik 2

35) Notiere zu folgenden Aussagen die Negation:

a) $x \geq 0$

b) $x < 12$

c) Y ist eine positive Zahl

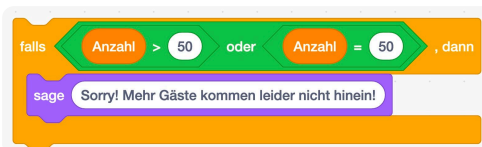
d) In der Variable Operation ist der Wert + gespeichert

e) Auf dem Motorrad sitzt mindestens ein Mensch.

f) Die Schublade ist leer.

g) Bei einem Fußballspiel sind genau 22 Spieler auf dem Feld.

h) Heute wird es höchstens 13°C warm werden.



a)



b)



c)



d)

36) Notiere zu diesen „Falls-und-dann-Bedingungen“ die Negation:

a) falls

dann

b) falls

dann

c) falls

dann

d) falls

dann

Exkurs - Algorithmen darstellen

Häufig lohnt es sich seinen Algorithmus vor der aufwendigen Programmierung zu notieren. So weiß man immer was man vor hatte und muss noch nicht auf die richtige Programmierung achten. Dazu lernst du eine textbasierte und zwei grafische Notationen kennen. Sieht dir die Videos der Reihe nach an. Erledige anschließend die Aufgaben.

Pseudocode



ogy.de/pseudocode

Pseudocode

Pseudo
Code

pseu do | pseudo |
 Umgangssprachlich
 Adjektiv
 HERKUNFT zu pseudo-, Pseudo-
 nicht echt, nur nachgemacht, nachgeahmt

Dient der Veranschaulichung

Ähnlich zu echtem Code

- ★ Einfacher als echter Code
- ★ Formaler als normale Sprache

Programmablaufplan

Programmablaufplan (Flussdiagramm)

Terminator	Verbindung	Anweisung	Verzweigung
Start/Stopp	→	Operation	 Frage? <div style="display: flex; justify-content: space-between; font-size: 0.8em;"> Ja → Nein ↓ </div>

<https://de.wikipedia.org/wiki/Programmablaufplan>



ogy.de/PAP

Struktogramm / Nassi-Shneidermann-Diagramm



ogy.de/Struktogramm

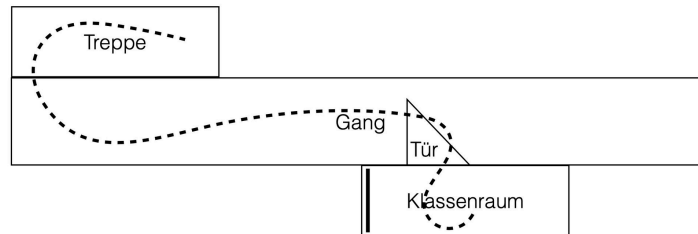
Struktogramm (Nassi-Shneiderman-Diagramm)

Teilprobleme werden sichtbar

Terminator	Anweisung	Verzweigung	Schleife
Start/Stopp	Operation	 Bedingung <div style="display: flex; justify-content: space-between; font-size: 0.8em;"> Ja Nein </div> <div style="display: flex; justify-content: space-between; font-size: 0.8em;"> Anweisung Anweisung </div>	Wiederhole x mal <div style="border: 1px solid white; padding: 2px; font-size: 0.8em; margin-top: 2px;"> Anweisung </div> Anweisung

<https://de.wikipedia.org/wiki/Nassi-Shneiderman-Diagramm>
https://www.inf-schule.de/algorithmen/algorithmen/algorithmusbegriff/exkurs_darstellung

Algorithmen (grafisch) darstellen - Aufgaben



Der Weg zum Pausenhof

Zuerst musst du aufstehen.

Dann musst du dich so drehen, dass du in Richtung Tafel schaust.

Laufe dann nach vorne bis du vor der Tafel stehst, drehe dich nochmal, sodass du zur Tür schaust.

5 Laufe durch die Tür bis auf den Gang.

Dreh dich so, dass du nicht zum Gangende schaust.

Geh ihn entlang bis du neben einer Treppe stehst.

Dreh dich nach rechts, gehe zwei Schritte vor und drehe dich noch 1x nach rechts, sodass du vor dem Treppenabgang stehst.

10 Gehe die Treppe hinunter. Gehe anschließend so lange geradeaus, bis du auf dem Schulhof bis.

37 Sieh dir die Grafik oben, sowie den Text „Der Weg zum Pausenhof an“.

Im Video wurde dieser Text in die verschiedenen Darstellungsformen umgewandelt. Schreibe bzw. zeichne diese ab und ergänze bis zum Ende.

38 Hier siehst du ein Rezept für Pfannkuchen.

Wähle eine der drei Darstellungsformen für Algorithmen aus und wandle das Rezept um.

Achtung: Die Zutaten müssen noch abgewogen bzw. abgezählt werden! Du hast noch nicht die korrekte Menge vorliegen!



Zutaten

2	Eier
200ml	Milch
1 Prise	Zucker
1 Prise	Salz
200g	Mehl
60ml	Mineralwasser
etwas	Speiseöl zum Ausbacken

Anleitung

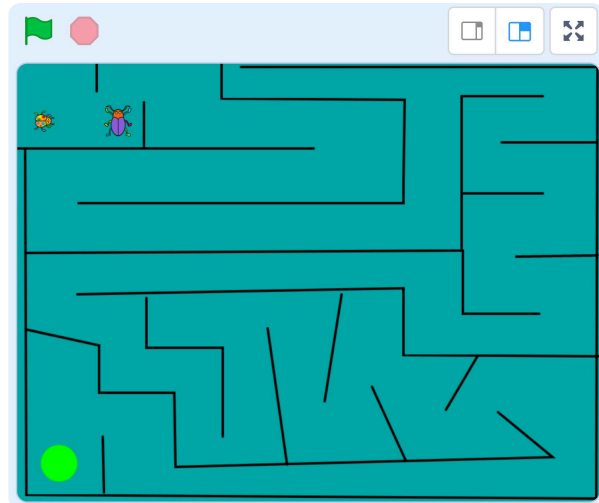
- 1 ↓ Eier mit Milch, Zucker, Salz, Mehl und Mineralwasser zu einem glatten Teig rühren.
- 2 ↓ Bei Bedarf noch etwas Mehl oder Wasser hinzugeben, um die gewünschte Konsistenz zu erreichen.
- 3 ↓ Eine beschichtete Pfanne mit etwas Speiseöl erhitzen.
- 4 ↓ Mit einer Schöpfkelle eine Kelle Teig in die Pfanne geben und die Pfanne kurz in jede Richtung schwenken um den Teig zu verteilen.
- 5 ↓ Den Pfannkuchen von beiden Seiten etwa 1-2 Minuten bräunlich ausbacken. Warm genießen.

Quelle: <https://www.einfachbacken.de/rezepte/pfannkuchen-das-schnelle-grundrezept>

Das erste Spiel

Jetzt bist du ausgestattet, um dein erstes "richtiges" Programm bzw. Spiel zu programmieren!

Rechts siehst du einen Irrgarten mit einem Ziel, dem grünen Punkt und den beiden "Spielern":



Hier sind die beiden Namen aus Scratch genannt, du kannst sie natürlich umbenennen.

Erstelle deinen eigenen Irrgarten - beachte, dass Ränder vorhanden sind.

Beetle (Größe 25) möchte zum grünen Punkt ...

- Du musst also die Steuerung für den Beetle erstellen.
- *Vergiss nicht eine "Boost"-Funktion zum schneller Laufen! (Und-Verknüpfung)*
- Wenn Beetle den grünen Punkt erreicht, so erhält er 10 Punkte.
- Du benötigst du eine Variable für den Punktestand: Deklarier sie und initialisiere sie bei Spielstart mit 10 Punkten. (Ggf. *musst du diesen Wert noch anpassen*)
- Sobald Beetle an den Rand oder eine Wand des Irrgartens stößt, soll er an den Start zurück gesetzt werden (Bedenke die Blickrichtung!).
- Jedes Zurücksetzen kostet 1 Punkt.

... und der Ladybug (Größe 20) verfolgt ihn. - Der Gegner:

- Der Ladybug soll computergesteuert unterwegs sein.
- Er soll nicht auf den Start zurück gesetzt werden, wenn er die Wand berührt.
- *Option: Auf Tastendruck wird die automatische Steuerung ausgesetzt und Ladybug kann von Spieler 2 gesteuert werden. Dann muss ihn aber die Wand auch zurücksetzen!*
- Sein Ziel ist es, Beetle zu fangen. Schafft er dies, werden beide zurück an den Startpunkt gesetzt.
- Jedes erfolgreiche Fangen kostet ebenfalls 1 Punkt.

Die Zeit und das Ziel des Spiels

- Hier musst du etwas herumprobieren. Der Spieler hat also x Minuten Zeit (*das hängt auch von der Schwierigkeit deines Irrgartens ab. Das Beispiel oben ist schon sehr anspruchsvoll!*) um den grünen Punkt so oft wie möglich zu erreichen.
- Ist sein Punktestand beim Ablaufen der Zeit positiv, gewinnt er - ansonsten... gewinnt der fiese Ladybug!

Falls dir noch weitere Verbesserungen oder Anpassungen einfallen, die das Spiel besser machen - dann probiere dies aus! Das ist das Kreative beim Programmieren! **Viel Spaß!**

Weitere Ideen:

Sobald du fertig bist, steht dir die Scratch Welt offen: Du könntest dein komplett eigenes Spiel programmieren! Hier sind ein paar Ideen:

- Pong
- Flappy Bird
- Renn-Spiele
- usw.