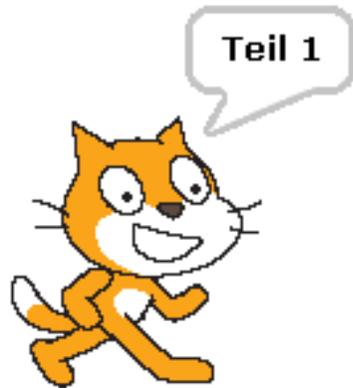


Bernd Gärtner

Name: _____

Handbuch Programmieren für Kinder...



...mit **SCRATCH**



Inhalt

- Thema 0: *Programmieren* 3
- Thema 1: *Erste Schritte* 9
- Thema 2: *Orientierung* 16
- Thema 3: *Wiederholungen* 22
- Thema 4: *Bedingungen* 26

Zeichenerklärung



Hier wird ein Scratch-Programm vorgestellt, das du selbst am Computer ausprobieren solltest.



Hier steht, was passiert, wenn du das korrekt zusammengebaute Programm ausführst.



Hier wird ein wichtiger Begriff oder ein neuer Scratch-Befehl eingeführt.



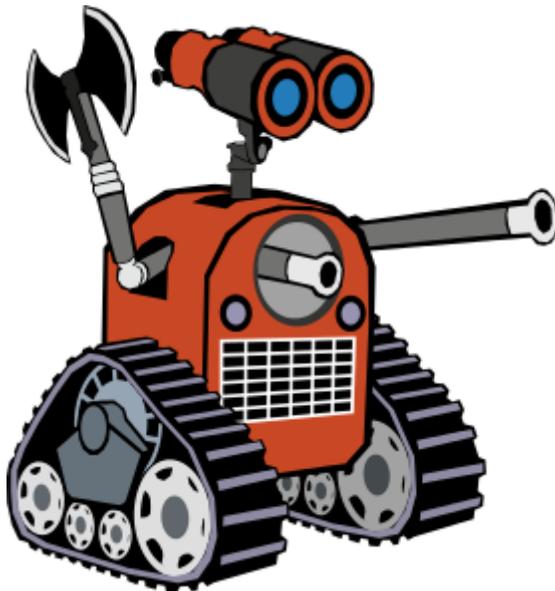
Ein Befehl in Klammern wird nicht im Handbuch, sondern im Arbeitsheft zum Handbuch eingeführt.



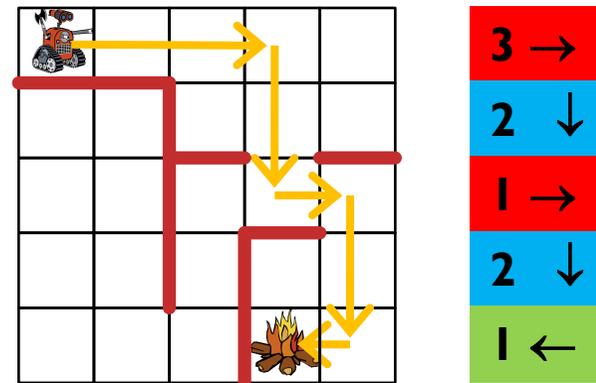
Im Arbeitsheft gibt es viele Aufgaben, die dir beim Scratch-Lernen helfen!

Thema 0*: Programmieren

Hier erfährst du, was Programmieren bedeutet und was Computerprogramme sind. Du lernst, dir aus Legosteinen Programme zur Steuerung eines vorgestellten Feuerlöschroboters zu bauen.



- Die **Begriffe**, die hier neu eingeführt werden:
 - (Computer-)Befehl
 - Programmiersprache
 - (Computer-)Programm



* Informatiker zählen immer ab 0.

Thema 0: Programmieren

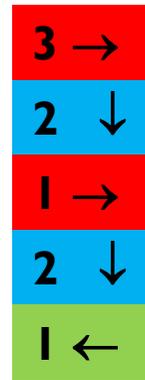
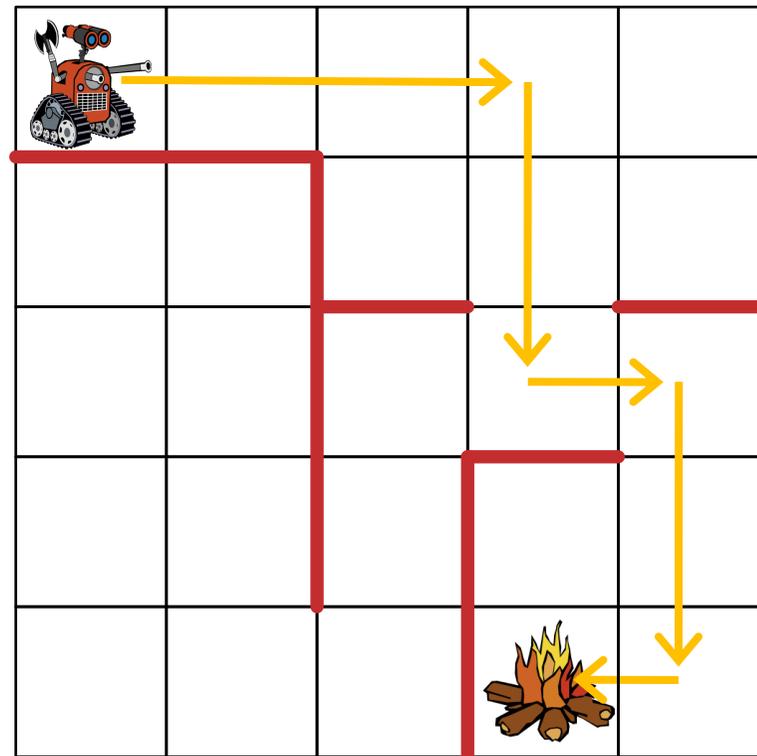
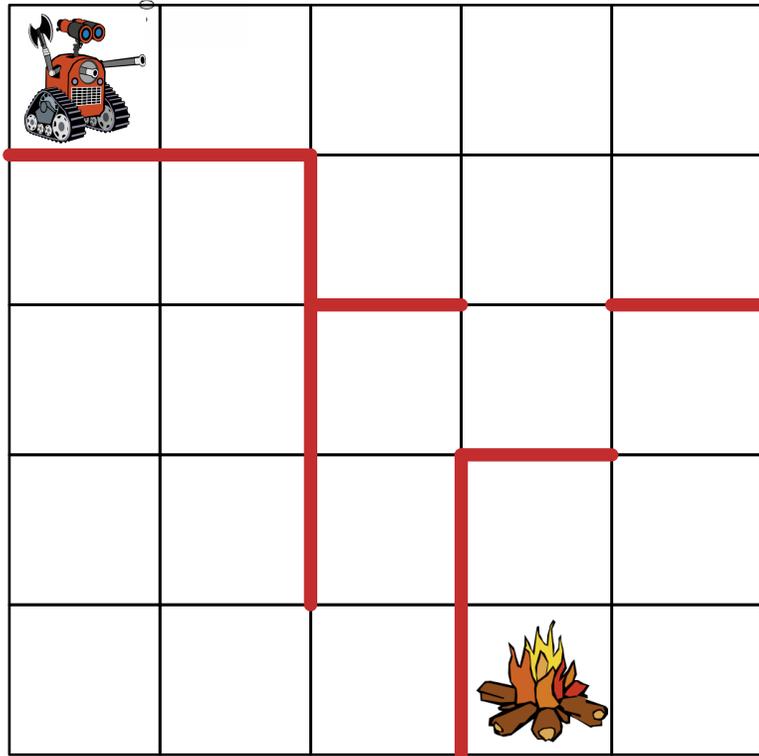
Was ist Programmieren?



Programmieren heisst, einem Computer (z.B. in einem Roboter) eine Folge von Befehlen zu erteilen, damit er genau das macht, was du von ihm willst. Computer verstehen nur ganz bestimmte einfache Befehle, deshalb brauchst du oft viele davon.



Gehe zum Brand!



Thema 0: Programmieren Programmiersprachen



Eine Programmiersprache ist eine Sprache, die der Computer versteht. Sie besteht aus Befehlen, die du im Umgang mit dem Computer benutzen kannst.

Die Programmiersprache, die der Löschroboter versteht, besteht aus den folgenden 17 Befehlen. Der Roboter kann nicht durch Wände gehen, kann sie aber vorher mit seiner Axt zerstören.

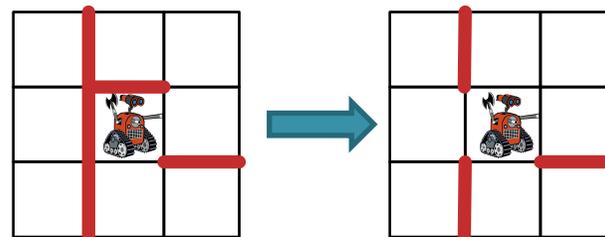
1 → **2 →** **3 →** **4 →** Gehe 1,2,3,4 Kästchen nach rechts

1 ↓ **2 ↓** **3 ↓** **4 ↓** Gehe 1,2,3,4 Kästchen nach unten

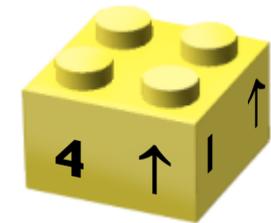
1 ← **2 ←** **3 ←** **4 ←** Gehe 1,2,3,4 Kästchen nach links

1 ↑ **2 ↑** **3 ↑** **4 ↑** Gehe 1,2,3,4 Kästchen nach oben

A Benutze die Axt; alle Wände, die direkt ums aktuelle Kästchen des Löschroboters stehen, werden zerstört.



Du kannst dir die Befehle ganz einfach aus Legosteinen basteln.



Beschrifte jede der vier Seiten eines Steins mit einem der vier Befehle für die Richtung.

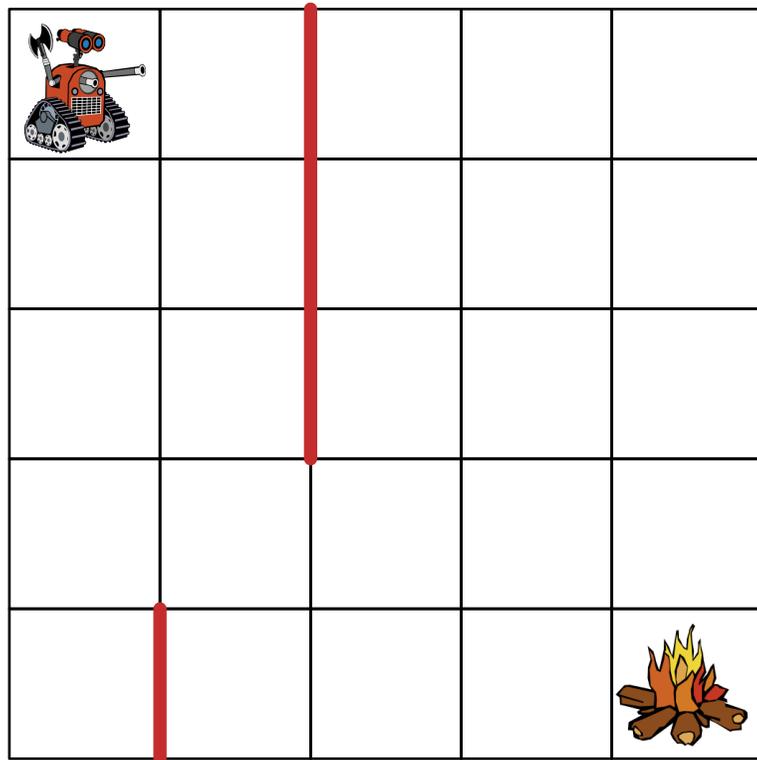


Thema 0: Programmieren Programme

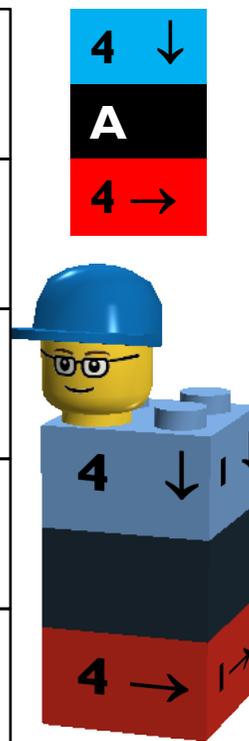
 Ein Programm besteht aus einer oder mehreren Befehlsfolgen. Programme werden in einer Programmiersprache aufgeschrieben und danach vom Computer ausgeführt.

Programm in der Löschroboter- Programmiersprache

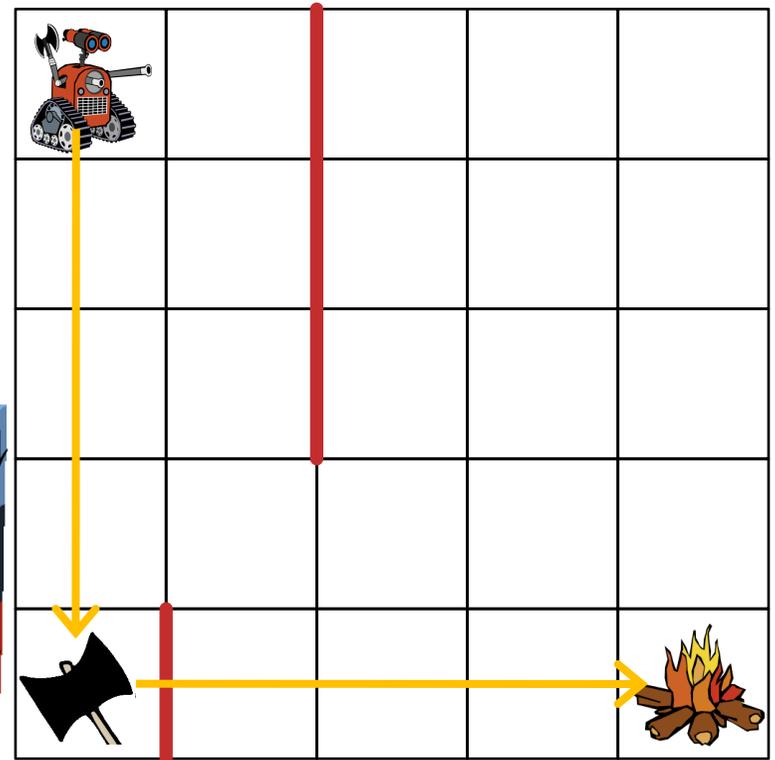
Ausgangssituation



Programmiersprache



Ausführung des Programms



SCRATCH ...

- ...ist eine neue Programmiersprache, die es dir ermöglicht, deine eigenen interaktiven Geschichten, Animationen, Spiele, Musik- und Kunstwerke zu erstellen und sie als *Scratch-Projekte* anderen über das Internet mitzuteilen.
- Bei Scratch programmierst du mit *Blöcken*, die du wie Legosteine stapelst, um dein Programm zusammenzubauen.

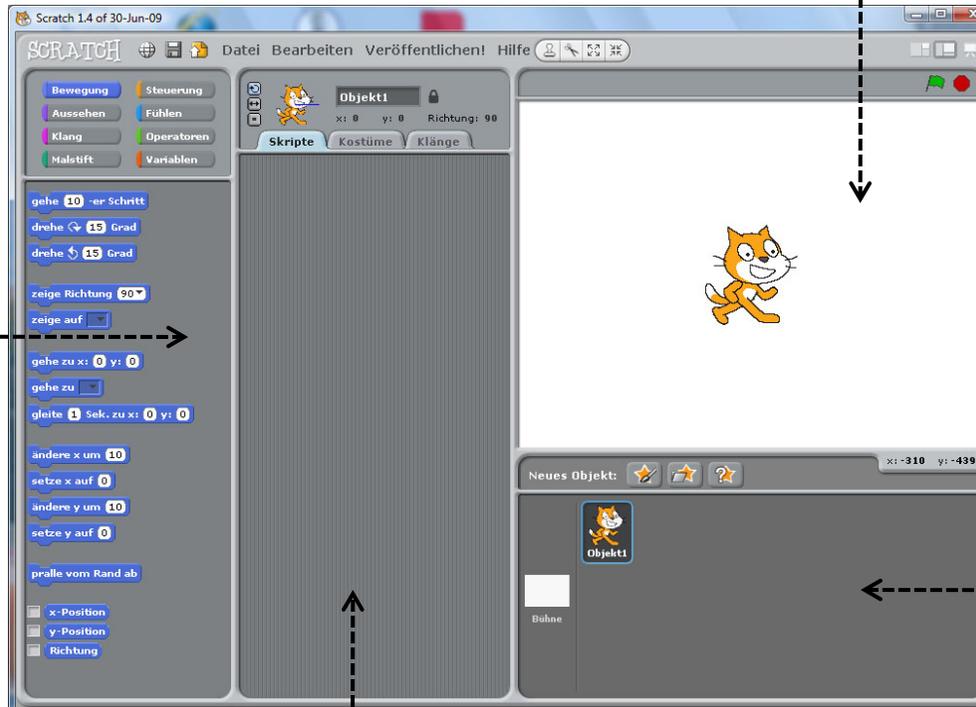


- Für weitere Informationen und zum Herunterladen (gratis) gehe zu <http://scratch.mit.edu/>

Die Benutzeroberfläche

Bühne: Hier läuft dein *Projekt* (Geschichte, Animation, Spiel, Musik...) ab. Die handelnden Lebewesen und Gegenstände in deinem Projekt heissen *Objekte*.

Blockpalette: Von hier holst du die Blöcke, aus denen du dann im Programmierbereich dein Programm zusammenbaust.

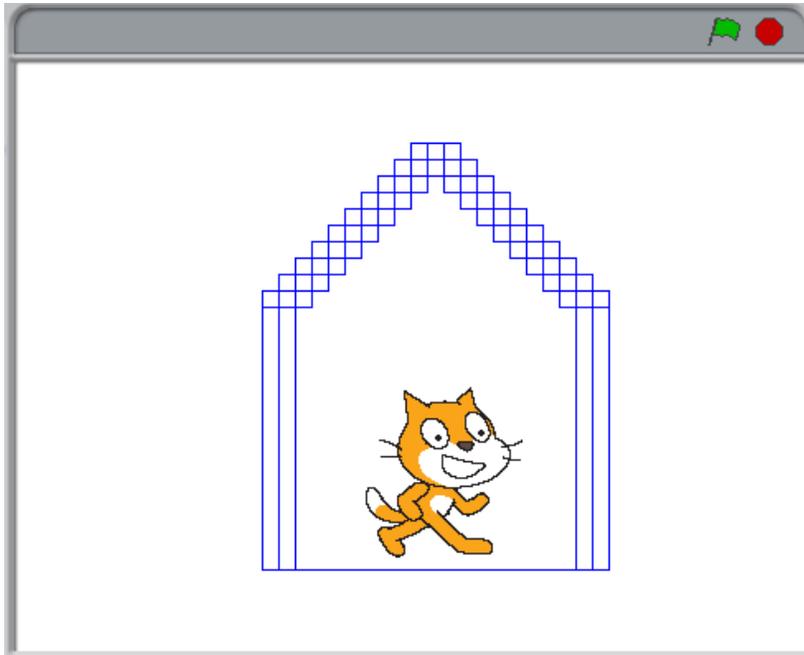


Objektliste: Um ein Objekt zu programmieren, wählst du es hier aus. Am Anfang gibt es nur ein Objekt, die Katze Scratch.

Programmierbereich: Hier baust du Blockstapel, die den Objekten sagen, was sie tun sollen. Die Stapel aller Objekte zusammen bilden dein *Programm*.

Thema 1: Erste Schritte

Hier realisierst du dein erstes interaktives Projekt. Du bringst einem Objekt das Laufen und Malen bei und steuerst es über die Tastatur.



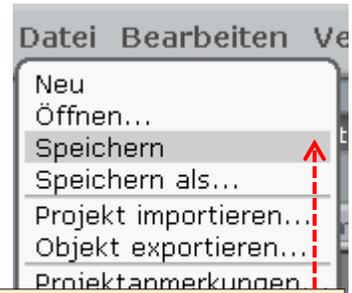
- Die **Blöcke**, die du dabei neu kennlernst:



- Die **Begriffe**, die hier neu eingeführt werden:
 - (Scratch-)Befehl
 - Skript (Befehlsfolge)
 - Attribut (Merkmal)
 - Hut (Ereignisbehandlung)

Thema I: Erste Schritte Geradeaus gehen

Hier kannst du dem Objekt einen Namen geben, zum Beispiel „Scratch“.



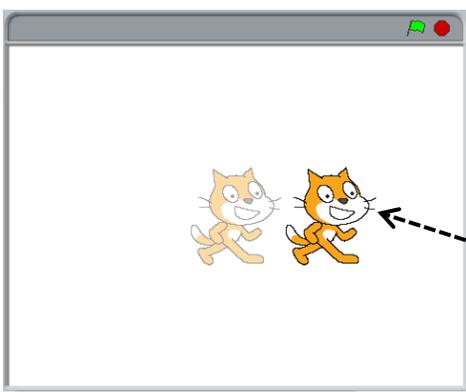
Wichtig: Dein Projekt regelmässig speichern!



Die 10 kannst du auch durch eine beliebige andere Zahl ersetzen.

Ziehe den Block **gehe 10 -er Schritt** aus der Blockpalette in den Programmierbereich. Klicke ihn dann ein paarmal an und beobachte die Bühne!

Jeder Block mit einer Vertiefung  oder Ausbuchtung  ist ein *Befehl*. Der Text im Block beschreibt den Befehl. Bei jedem Anklicken des Blocks führt dein Objekt den Befehl einmal aus.



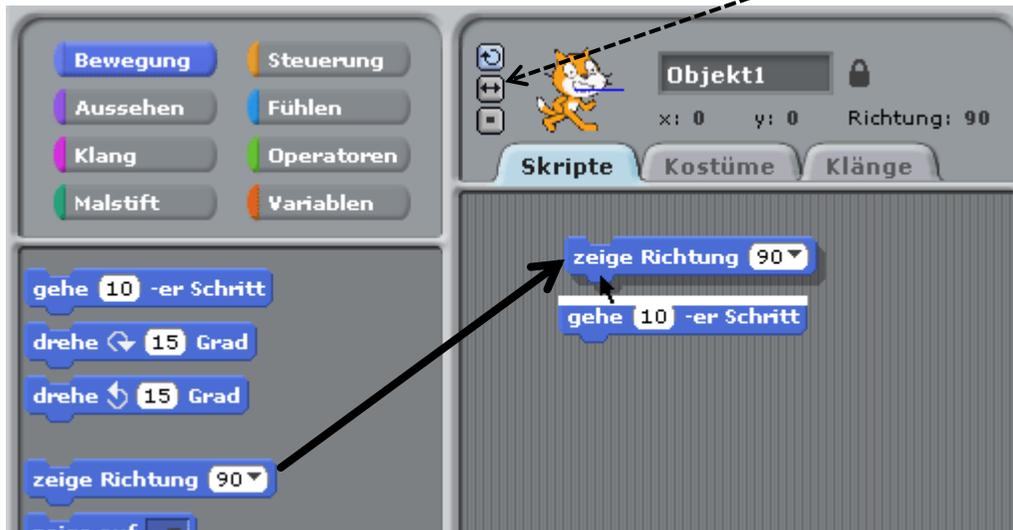
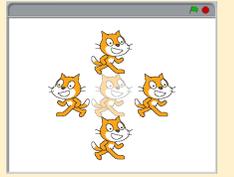
Die Katze Scratch geht bei jedem Klick 10 Schritte (= Bildpunkte oder Pixel) geradeaus. Nach zehnmal Klicken und 100 Schritten ist Scratch hier gelandet.

Thema I: Erste Schritte

Die Richtung wählen

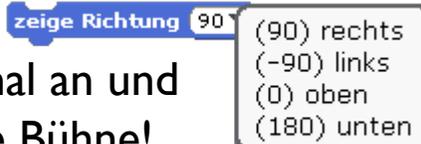


Klicke hier für dieses Verhalten:



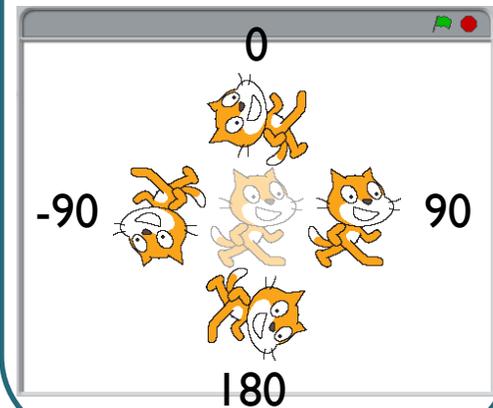
Halte den Block **zeige Richtung 90** über den ersten Block und lasse beide zu einem Stapel zusammenschnappen. Wähle im Blockmenü eine Richtung aus:

Klicke den Stapel ein paarmal an und beobachte dabei wieder die Bühne!



Ein Stapel von Befehlen ist ein  **Skript**. Bei jedem Anklicken des Stapels führt dein Objekt alle Befehle des Skripts einmal aus (von oben nach unten).

 Scratch zeigt bei jedem Klick zuerst in die ausgewählte Richtung und geht in dieser Richtung dann 10 Schritte geradeaus.

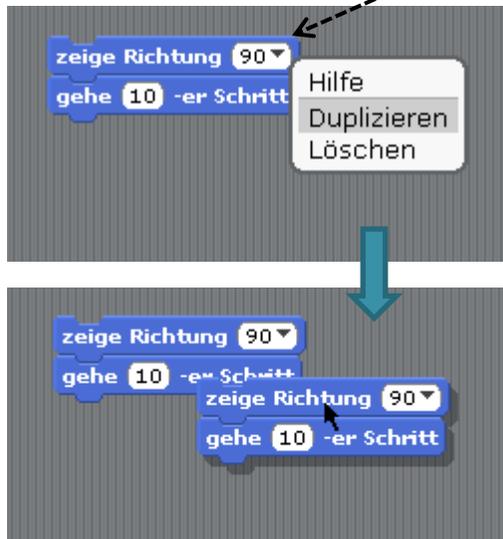


SCRATCH -Technik

Stapelbehandlung

- Kopieren, Löschen, Hilfe

Menü durch Rechtsklicken eines Blocks öffnen!

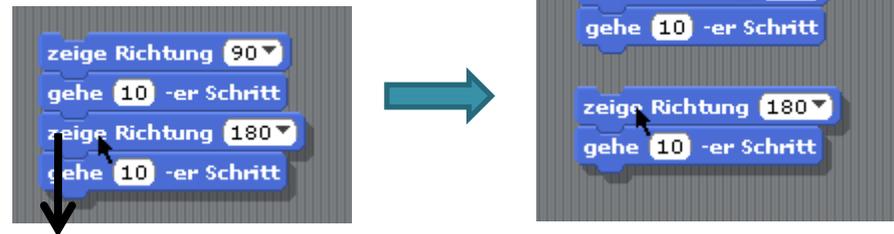


- Anbauen



Beim Loslassen wird der gezogene Block oder Stapel an der weissen Linie eingebaut.

- Auftrennen



Thema I: Erste Schritte

Das Kostüm wechseln

Die violetten Blöcke findest du im Menü „Aussehen“, die blauen bei „Bewegung“.



Baue dir im Programmierbereich diese beiden Skripte zusammen:
Klicke beide abwechselnd ein paarmal an. Was passiert dabei auf der Bühne?

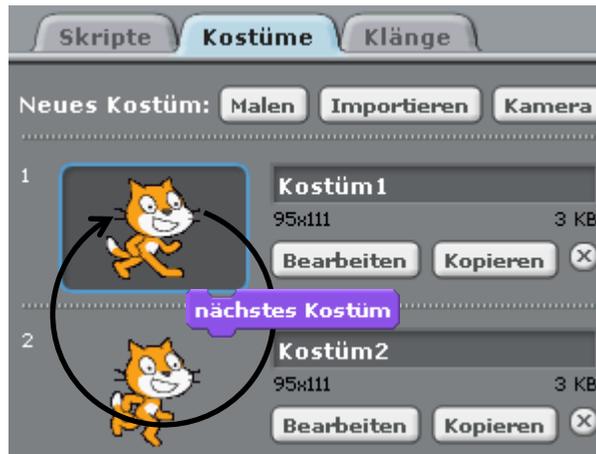
```

zeige Richtung 90
nächstes Kostüm
gehe 10 -er Schritt

zeige Richtung -90
nächstes Kostüm
gehe 10 -er Schritt
    
```



Scratch macht „echte“ Laufbewegungen in die jeweilige Richtung. Erreicht wird das durch Kostümwechsel. Scratch hat am Anfang zwei Kostüme zur Auswahl. Du kannst neue Kostüme malen, laden (*importieren*) oder per Kamera aufnehmen. Du kannst bestehende Kostüme bearbeiten und kopieren.



Das Kostüm ist ein *Attribut* (Merkmal) des Objekts. Andere Attribute sind Standort, Richtung und Verhalten des Malstifts (Handbuch S.15). Alle Attribute zusammen beschreiben den aktuellen Zustand des Objekts, der beim Speichern des Projekts mitgespeichert wird.



Thema I: Erste Schritte

Auf Tasten reagieren

Die gelben Blöcke findest du im Menü „Steuerung“.



Baue dir im Programmierbereich diese beiden Skripte zusammen:

```

Wenn Taste Pfeil nach links gedrückt
  zeige Richtung -90
  nächstes Kostüm
  gehe 10 -er Schritt
  
```

```

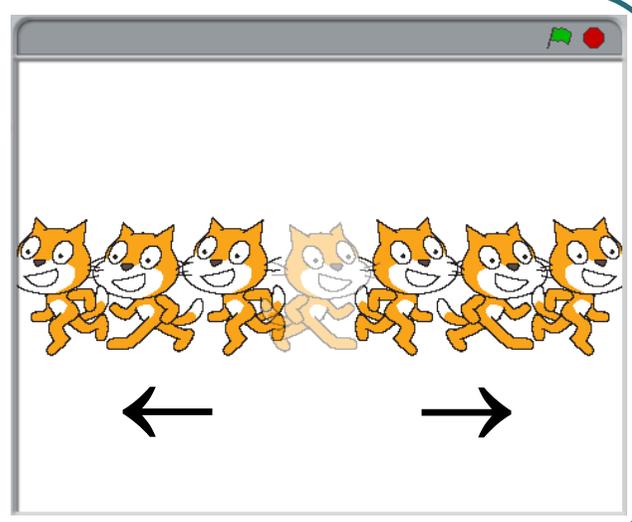
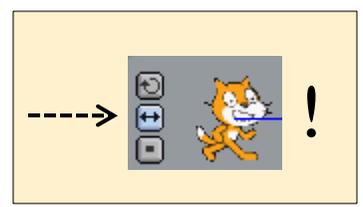
Wenn Taste Pfeil nach rechts gedrückt
  zeige Richtung 90
  nächstes Kostüm
  gehe 10 -er Schritt
  
```

Wähle die gewünschte Taste jeweils aus dem Menü des Blocks aus!

Drücke dann die Pfeiltasten (← und →) und beobachte, was Scratch auf der Bühne macht!

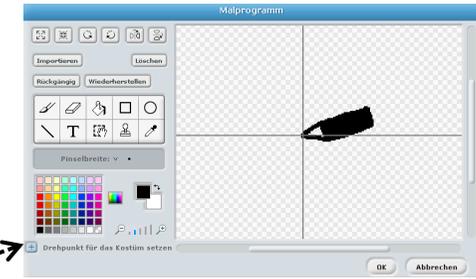
Jeder Block von der Form  ist ein *Hut*. Wenn er einem Skript „aufgesetzt“ wurde, führt das Objekt das Skript jedes Mal aus, wenn das im Hut angegebene Ereignis eintritt. Hüte erlauben dir, Objekte gezielt zu steuern.

Wenn du eine der Pfeiltasten gedrückt hältst, läuft Scratch flüssig in die zugehörige Richtung.



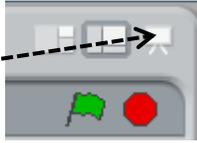
Thema I: Erste Schritte

Den Malstift benutzen



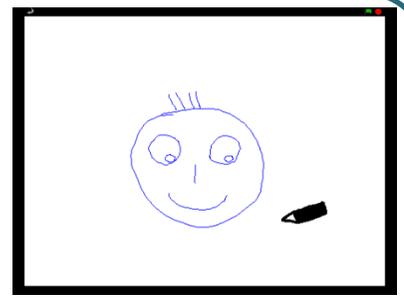
Male für Scratch ein Stift-Kostüm, klicke auf **+ Drehpunkt für das Kostüm setzen** und verschiebe den Kostüm-Mittelpunkt auf die Stiftspitze. Klicke dann auf das Schloss neben dem Objektnamen, um es zu öffnen. Baue dir im Programmierbereich diese drei Skripte zusammen:



Wechsle durch Klick auf das Leinwandsymbol  in den Präsentationsmodus. Ziehe den Stift mit der Maus herum (das geht nur bei offenem Schloss!) und beobachte, was nach dem Drücken von ↓, ↑ und der Leertaste passiert!



Du kannst mit dem Stift auf der Bühne ein Bild malen.

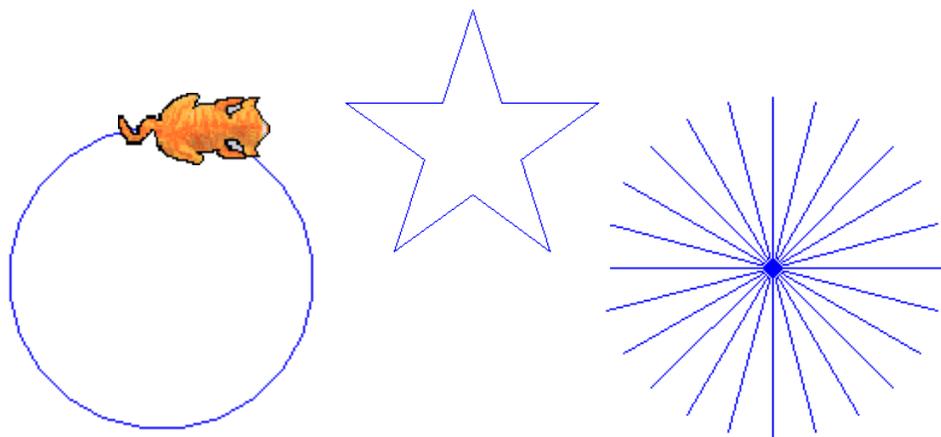


Jedes Objekt hat einen Malstift dabei. Wenn er abgesenkt ist, zeichnet er die Bewegungen des Objekts nach. Es gibt Befehle zum Wählen und Ändern der Stiftstärke, Stiftfarbe und Farbstärke.



Thema 2: Orientierung

Hier lernst du, wie du dich auf der Bühne orientieren kannst: Du bringst einem Objekt bei, seine Position und Richtung zu wählen und zu ändern. Damit kannst du per Tastendruck leicht geometrische Muster malen.



- Die **Blöcke**, die du dabei neu kennenlernst:



- Die **Begriffe**, die hier neu eingeführt werden:
 - Aufräumskript* (Initialisierung)
 - Punkt*, *Richtung*, *Drehwinkel* (Geometrie)

Thema 2: Orientierung

Die Position wählen



Erstelle dieses Skript und klicke die grüne Flagge über der Bühne an!

Wenn  angeklickt

gehe zu x: -240 y: 0

senke Stift ab

gehe zu x: 240 y: 0

hebe Stift an

gehe zu x: 0 y: -180

senke Stift ab

gehe zu x: 0 y: 180

hebe Stift an

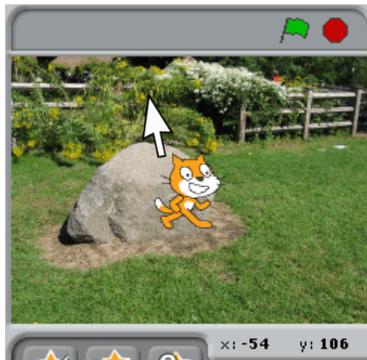
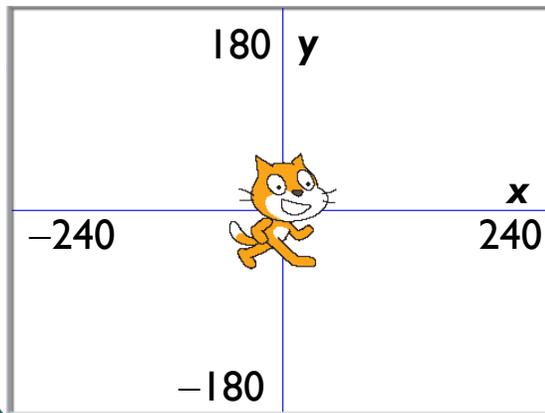
gehe zu x: 0 y: 0

x-Achse

y-Achse



Scratch malt das Koordinatenkreuz seiner Welt (der Bühne) und geht danach wieder zum Punkt (0,0), der Mitte der Bühne.



gehe zu x: -54 y: 106



Standort des Mauszeigers

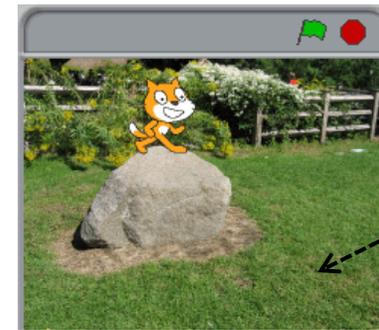
Ein Punkt wird mit x - und y -Koordinaten beschrieben (x : links \rightarrow rechts, y : unten \rightarrow oben).



Beispiel: Punkt $(-54, 106)$ ist 54 Schritte links und 106 Schritte oberhalb der Mitte.



Wechsel des Hintergrunds:
Auswahl Bühne \rightarrow Hintergründe



Nature

Thema 2: Orientierung Aufräumen



Lasse Scratch auf der Bühne herumlaufen oder ein Bild malen. Erstelle dann dieses Skript und drücke die Taste ‚a‘!
Was passiert?

```
Wenn Taste a gedrückt
  hebe Stift an
  wische Malspuren weg
  zeige Richtung 90
  gehe zu x: 0 y: 0
```

Ein Skript, das ein Objekt und/oder die Bühne in einen ganz bestimmten Zustand bringt, heisst *Aufräumskript*. Aufräumskripte sorgen dafür, dass dein Projekt (Geschichte, Animation, Spiel, ...) immer „richtig“ anfängt oder weitergeht.

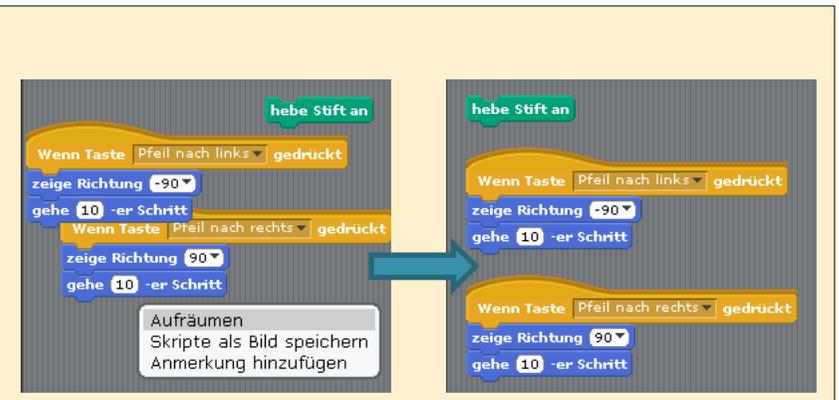
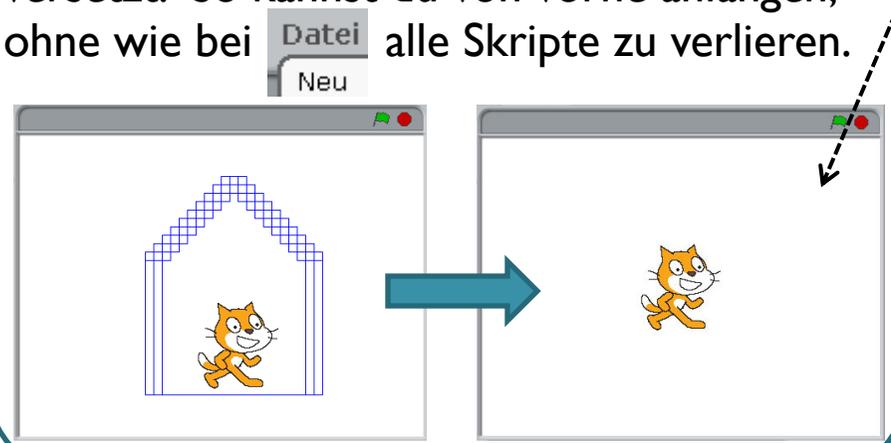


Gehe zur Mitte der Bühne (siehe auch *Orientierung / Die Position wählen*)

Der „aufgeräumte“ Zustand kann so, aber auch ganz anders aussehen.



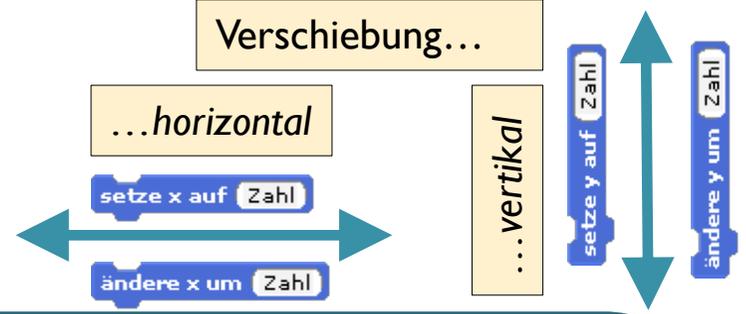
Scratch und die Bühne werden wieder in den Ausgangszustand versetzt. So kannst du von vorne anfangen, ohne wie bei **Datei** alle Skripte zu verlieren.



Du kannst auch den Programmierbereich automatisch aufräumen lassen.

Thema 2: Orientierung

Die Position ändern



Erstelle die drei Skripte unten und benutze dann die Pfeiltasten →, ←, ↑. Was beobachtest Du?

Skript 1: Wenn Taste Pfeil nach rechts gedrückt → nächstes Kostüm → gehe 10 -er Schritt

Skript 2: Wenn Taste Pfeil nach links gedrückt → setze x auf -190

Skript 3: Wenn Taste Pfeil nach oben gedrückt → nächstes Kostüm → ändere y um 50 → warte 1 Sek. → nächstes Kostüm → ändere y um -50

auf gleicher Höhe zu x=-190

50 nach oben und kurz warten...

...dann wieder 50 nach unten

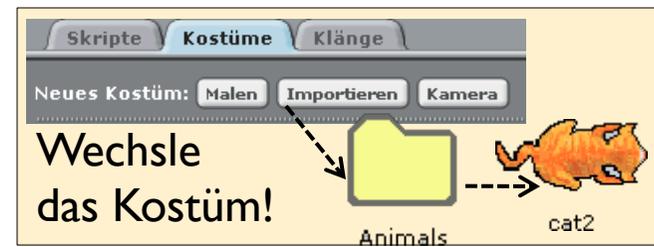
Mit diesen Befehlen verschiebst du ein Objekt ohne Richtungsänderung *horizontal* (setze und ändere x) oder *vertikal* (setze und ändere y).

Befehl	Effekt auf Position
setze x auf Zahl	$(x, y) \rightarrow (Zahl, y)$
ändere x um Zahl	$(x, y) \rightarrow (x + Zahl, y)$
setze y auf Zahl	$(x, y) \rightarrow (x, Zahl)$
ändere y um Zahl	$(x, y) \rightarrow (x, y + Zahl)$
gehe zu x: Zahl1 y: Zahl2	$(x, y) \rightarrow (Zahl1, Zahl2)$

Bei → läuft Scratch 10 Schritte nach rechts, bei ← geht sie an den linken Rand, und bei ↑ macht Scratch einen kleinen Hüpf.

Thema 2: Orientierung

Die Richtung ändern



Gib Scratch das Aufrämskript und die drei Tastenskripte unten. Was passiert beim Drücken der Pfeiltasten →, ← und der Leertaste?

```

Wenn Taste a gedrückt
  hebe Stift an
  wische Malspuren weg
  zeige Richtung 90
  gehe zu x: 0 y: 0
  
```

```

Wenn Taste Pfeil nach rechts gedrückt
  drehe 30 Grad
  
```

„+ 5 Min“

```

Wenn Taste Pfeil nach links gedrückt
  drehe -30 Grad
  
```

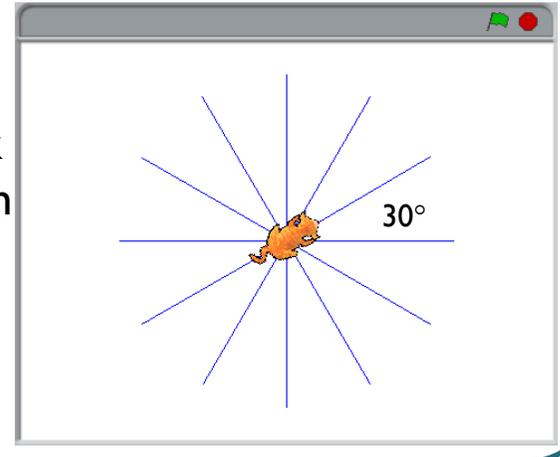
„- 5 Min“

```

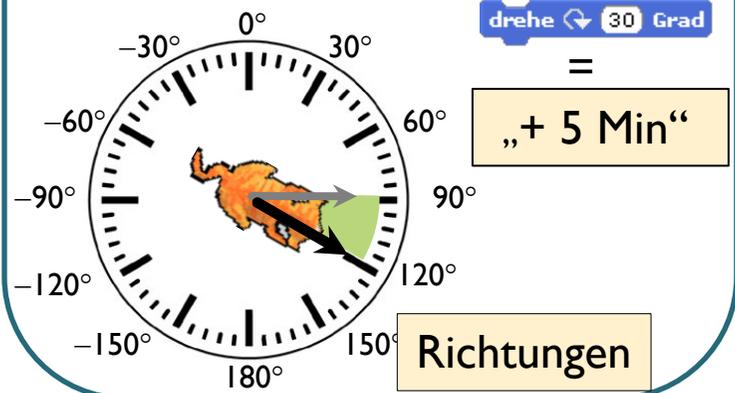
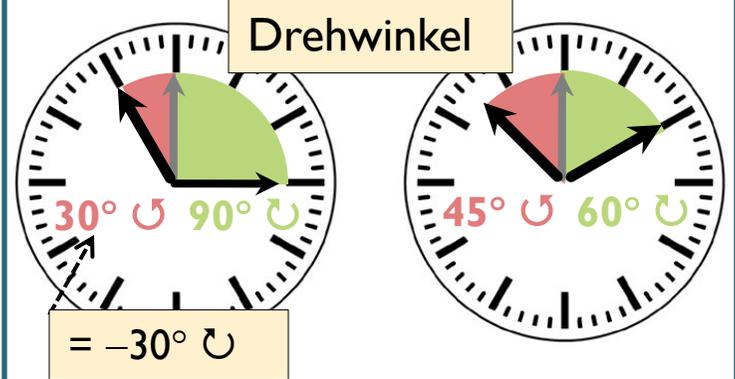
Wenn Taste Leertaste gedrückt
  senke Stift ab
  gehe 150 -er Schritt
  gehe zu x: 0 y: 0
  
```



Bei den Pfeiltasten dreht sich Scratch ein Stück nach rechts oder nach links. Die Leertaste „feuert“ eine Linie in die aktuelle Richtung ab.



Drehwinkel und Richtungen werden in Grad (°) gemessen. 360° ist eine volle Drehung. Man dreht *im* (↻) oder *gegen* (↺) den Uhrzeigersinn.



Thema 2: Orientierung

Vielecke malen



Gib Scratch diese zwei Skripte und probiere im unteren Skript die Winkel 30° , 45° , 60° , 72° , 90° , 120° . Wie oft musst du jeweils die Taste „Pfeil nach rechts“ drücken, bis Scratch wieder am Ausgangspunkt ist, und wie sehen die Bilder aus?

```

Wenn Taste a gedrückt
  hebe Stift an
  wische Malspuren weg
  zeige Richtung 90
  gehe zu x: 0 y: 150
  
```

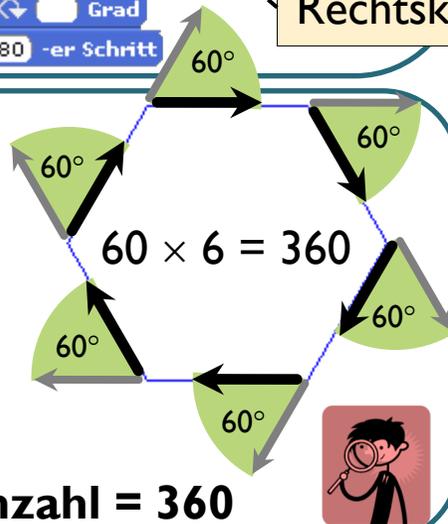
```

Wenn Taste Pfeil nach rechts gedrückt
  senke Stift ab
  drehe Grad
  gehe 80 -er Schritt
  
```

Rechtskurve

- $30 \times 12 = 360$
- $45 \times 8 = 360$
- $72 \times 5 = 360$
- $60 \times 6 = 360$
- $90 \times 4 = 360$
- $120 \times 3 = 360$

Winkel \times Eckenzahl = 360

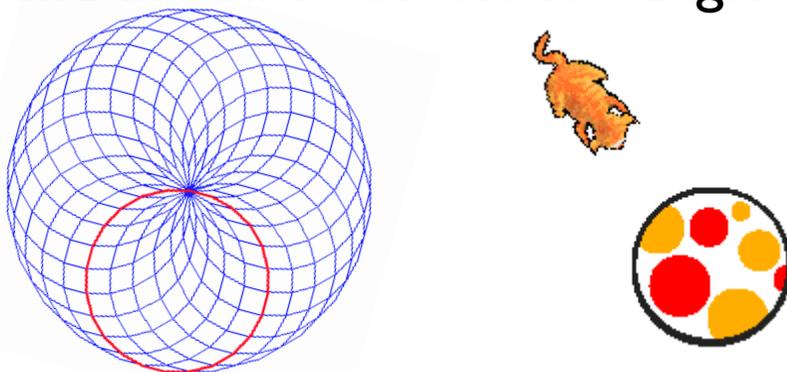


Scratch malt Vielecke. Für jede Ecke musst du einmal \rightarrow drücken.

<p>30° Zwölfeck</p>	<p>45° Achteck</p>
<p>60° Sechseck</p>	<p>72° Fünfeck</p>
<p>90° Viereck</p>	<p>120° Dreieck</p>

Thema 3: Wiederholungen

Hier erfährst du, wie du Befehle und Befehlsfolgen mehrfach ausführen kannst. Dadurch entstehen auf einfache Weise „komplizierte“ Abläufe, die du aber gut planen musst. Du erfährst, wie ein Objekt seine Position ändern kann, und wie du neue Objekte lädst und auf ihre Positionen reagierst.



- Die **Blöcke**, die du dabei neu kennlernst:



- Die **Begriffe**, die hier neu eingeführt werden:
 - Wiederholung (Schleife)
 - Ablauf (Kontrollfluss)

Thema 3: Wiederholungen Dinge mehrmals tun

Bewegung **Steuerung**

Aussehen Fühlen

Klang Operatoren

Malstift Variablen



Erstelle dieses Skript und klicke auf die grüne Flagge über der Bühne. Was passiert?

```

Wenn grüne Flagge angeklickt
  senke Stift ab
  wiederhole 24 mal
    drehe 15 Grad
    gehe 20 -er Schritt
  
```

```

wiederhole 24 mal
  drehe 15 Grad
  gehe 20 -er Schritt
  
```

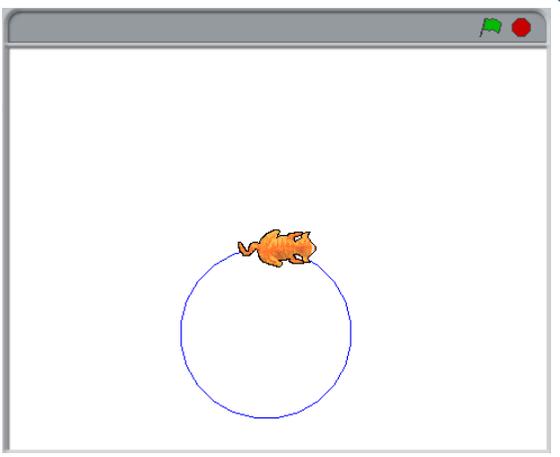
```

wiederhole 24 mal
  drehe 15 Grad
  gehe 20 -er Schritt
  
```

Lasse den Wiederhole-Mehrmals-Block um ein Skript „herumschnappen“ oder baue das Skript in den Block ein!



Beim Ereignis „Grüne Flagge angeklickt“ malt Scratch „von selbst“ einen Kreis (genauer: ein 24-Eck).



Wenn es dir zu schnell geht: warte Sek.

```

senke Stift ab
wiederhole 4 mal
  drehe 90 Grad
  gehe 100 -er Schritt
} 1
  drehe 90 Grad
  gehe 100 -er Schritt
} 2
  drehe 90 Grad
  gehe 100 -er Schritt
} 3
  drehe 90 Grad
  gehe 100 -er Schritt
} 4
  
```

Der *Wiederhole-Mehrmals-Befehl* führt die von ihm umschlossene Befehlsfolge so oft hintereinander aus, wie es im Zahlenfeld steht.

wiederhole mal

Thema 3: Wiederholungen

Abläufe planen

Wenn du ein „kompliziertes“ Muster malen willst, musst du den Ablauf sorgfältig planen!

Erstelle dieses Skript und klicke auf die grüne Flagge! Was passiert?

```

    Wenn grüne Flagge angeklickt
    wiederhole 3 mal
    senke Stift ab
    wiederhole 4 mal
    drehe 90 Grad
    gehe 100 -er Schritt
    hebe Stift an
    gehe 100 -er Schritt
    
```

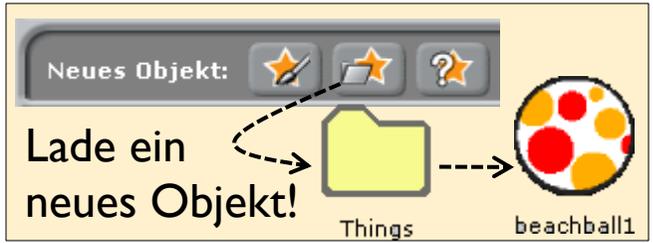
Scratch malt dieses Muster aus 3 Quadraten.

Ablauf: Wiederhole dreimal die Schritte 1 und 2:

1. Quadrat mit Seitenlänge 100 malen
2. Stift heben, 100 Schritte gehen
1. Quadrat mit Seitenlänge 100 malen
2. Stift heben, 100 Schritte gehen
1. Quadrat mit Seitenlänge 100 malen
2. Stift heben, 100 Schritte gehen

Thema 3: Wiederholungen

Dinge endlos tun



Lade eine zweites Objekt (zum Beispiel einen Ball), gib ihm einen passenden Namen und erstelle für Scratch dieses Skript. Klicke dann die grüne Flagge an und ziehe den Ball mit der Maus herum. Was passiert?

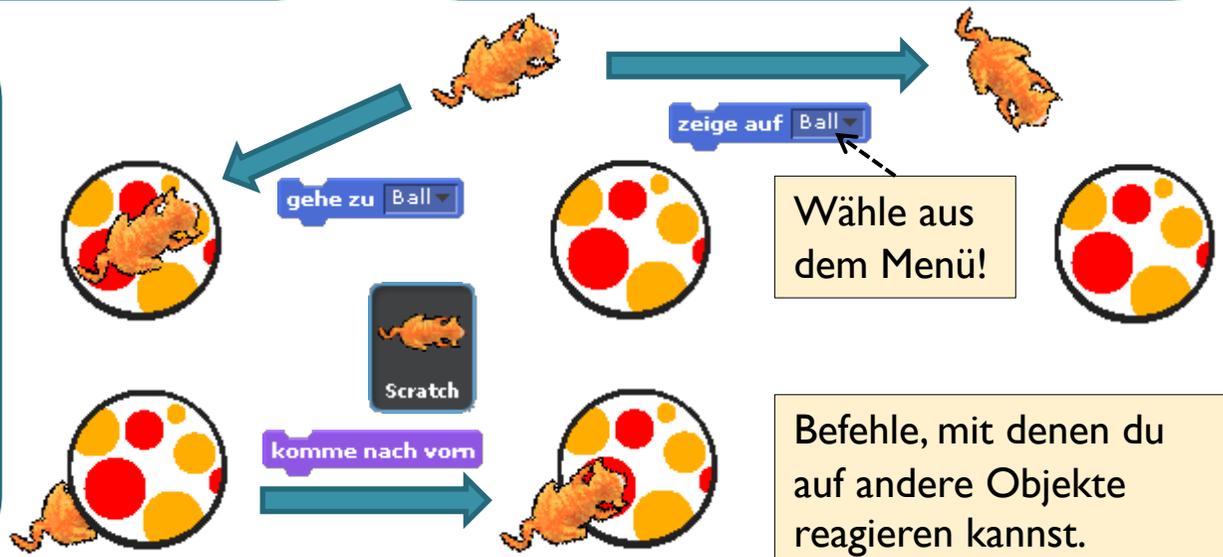
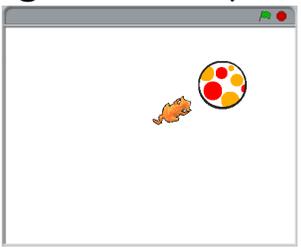
```
Wenn [grüne Flagge] angeklickt
  wiederhole fortlaufend
    zeige auf Ball
    gehe 5 -er Schritt
```

Der Wiederhole-Fortlaufend-Befehl sorgt dafür, dass die von ihm umschlossene Befehlsfolge immer wieder und ohne Ende ausgeführt wird. Klicken auf das Skript oder den roten Punkt über der Bühne (stoppe alle Skripte) stoppt auch den Wiederhole-Fortlaufend-Befehl.



```
wiederhole fortlaufend
```

Scratch versucht, den Ball zu fangen: sie dreht sich fortlaufend in Richtung des Balls und geht dabei jedes Mal fünf Schritte auf den Ball zu.

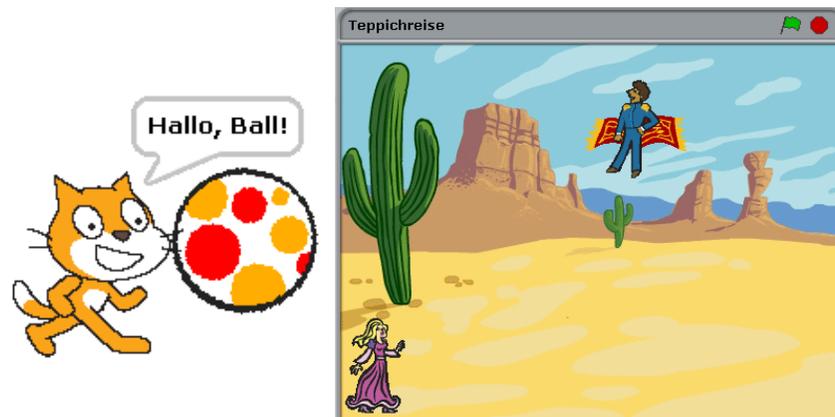


Wähle aus dem Menü!

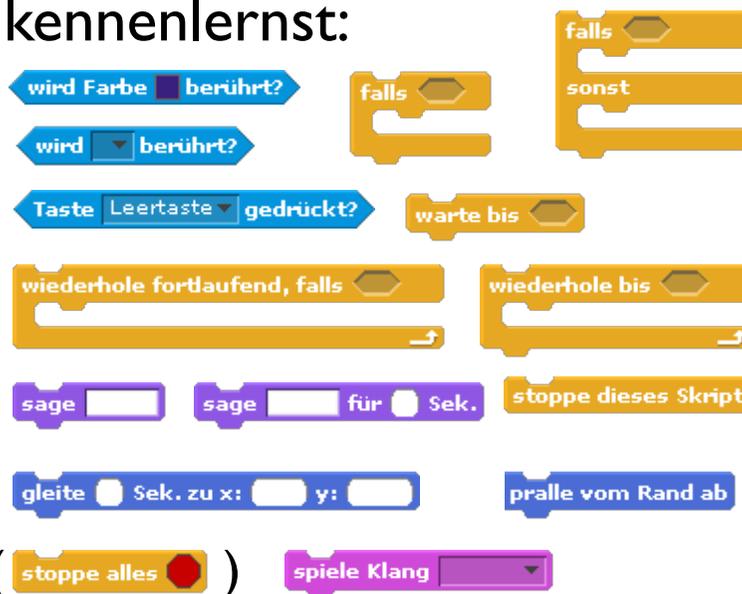
Befehle, mit denen du auf andere Objekte reagieren kannst.

Thema 4: Bedingungen

Hier lernst du, wie du etwas tun kannst, falls eine bestimmte Bedingung gilt, und wie du darauf warten kannst, dass sie gilt. Damit bist du schon in der Lage, Spiele zu programmieren, in denen Objekte auf Berührungen von Farben und anderen Objekte reagieren.



Die **Blöcke**, die du dabei neu kennenlernst:



- Die **Begriffe**, die hier neu eingeführt werden:
 - Bedingung (Prädikat)
 - Befehl mit Bedingung (Verzweigung)

Thema 4: Bedingungen

Bedingungen prüfen



Ziehe diese drei Blöcke in den Programmierbereich von Scratch und klicke sie an, bis du eine Antwort bekommst. Versuche bei allen drei Blöcken, zwei verschiedene Antworten zu bekommen!



wird Ball berührt?

Taste Leertaste gedrückt?

wird Farbe berührt?

Gehe ins Hilfe-Menü des Blocks, um herauszufinden, wie du die rote Farbe aus dem Ball in dieses Feld bekommst!



Die Antwort ist „wahr“ oder „falsch“, abhängig von der Situation, die beim Klicken auf den jeweiligen Block vorliegt.

wird Ball berührt?

falsch



wird Farbe berührt?

falsch

wird Ball berührt?

wahr



wird Farbe berührt?

falsch

wird Ball berührt?

wahr



wird Farbe berührt?

wahr

Jeder Block der Form



ist eine *Bedingung*. Der Text in der Bedingung ist eine Ja-Nein-Frage. Bei jedem Anklicken der Bedingung wird die Ja-Nein-Frage der Situation entsprechend beantwortet. Antwort **wahr** bedeutet „Ja, das stimmt, die Bedingung gilt!“. Antwort **falsch** heisst „Nein, das stimmt nicht, die Bedingung gilt nicht!“

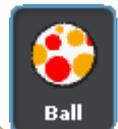
Thema 4: Bedingungen Dinge tun, falls...



Erstelle für den Ball dieses Skript und klicke die grüne Flagge an! Was passiert, wenn du mit dem Mauszeiger den Ball berührst?

```

Wenn  angeklickt
wiederhole fortlaufend
  falls  wird Mauszeiger berührt?
    zeige auf Mauszeiger
    gehe -5 -er Schritt
  
```

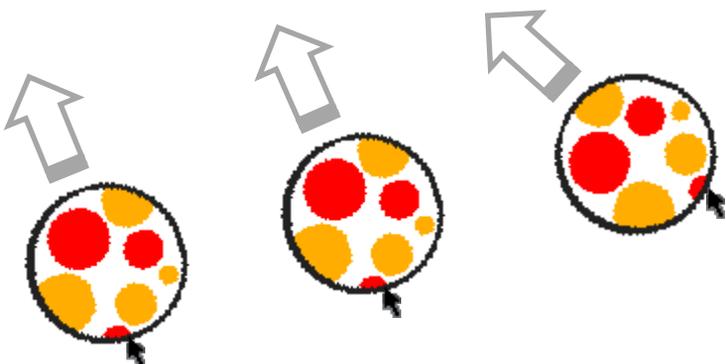


```

Wenn  angeklickt
wiederhole fortlaufend, falls  wird Mauszeiger berührt?
  zeige auf Mauszeiger
  gehe -5 -er Schritt

```

Noch kürzer!



Jedes Mal, wenn er vom Mauszeiger berührt wird, springt der Ball ein kleines Stück vom Mauszeiger weg. Auf diese Weise kannst du den Ball mit der Maus auf der Bühne herumschieben.

Jeder Block mit einer Ausparung dieser Art () ist ein *Befehl mit Bedingung*. Der *Falls-Befehl* sorgt dafür, dass die von ihm umschlossene Befehlsfolge ausgeführt wird, falls seine Bedingung gilt. Gilt die Bedingung nicht, wird die Befehlsfolge nicht ausgeführt, es passiert also nichts. In beiden Fällen geht es danach mit dem nächsten Befehl im Skript weiter. Willst du auch etwas tun, wenn die Bedingung nicht gilt, dann nimm den *Falls-Sonst-Befehl*!



Thema 4: Bedingungen Dinge tun, bis...



Gib Scratch und dem Ball diese beiden Skripte. Klicke dann die grüne Flagge an und schiebe den Ball zu Scratch! Was passiert?

Scratch Skript:

- Wenn angeklickt
- Warte bis **wird Ball berührt?**
- Warte **1 Sek.**
- sage **Hallo, Ball!** für **1 Sek.**

Ball Skript:

- Wenn angeklickt
- wiederhole bis **wird Scratch berührt?**
 - falls **wird Mauszeiger berührt?**
 - zeige auf **Mauszeiger**
 - gehe **-5** -er Schritt
- sage **Hallo, Scratch!** für **1 Sek.**

...erst ist der Ball dran!

Der *Wiederhole-Bis-Befehl* sorgt dafür, dass die von ihm umschlossene Befehlsfolge solange wiederholt ausgeführt wird, bis seine Bedingung gilt. Danach geht es mit dem nächsten Befehl im Skript weiter. Vor jeder Wiederholung wird die Bedingung neu geprüft. Es kann also auch passieren, dass die Befehlsfolge gar nicht oder endlos ausgeführt wird.



Scratch wartet auf den Ball, während der Ball zu ihr geht. Ist der Ball angekommen, begrüßen sich die beiden.

Nimm den *Warte-Bis-Befehl*, wenn du *nichts* tun willst, bis...

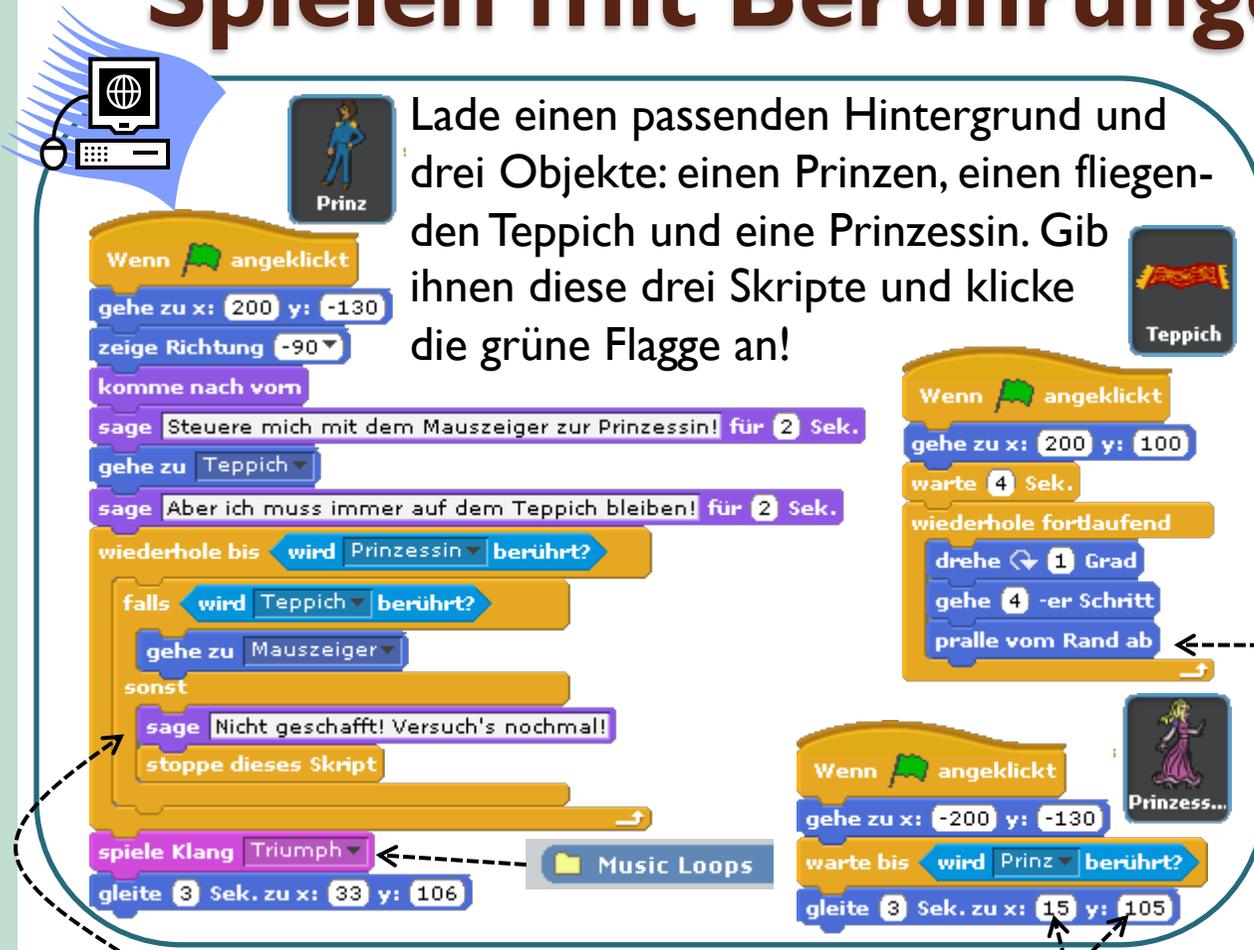
warte bis **wird Ball berührt?**

=

wiederhole bis **wird Ball berührt?**

Thema 4: Bedingungen Spielen mit Berührungen

Hier lernst du zusätzlich noch einige weitere praktische Befehle kennen!



Prinz

- Wenn **angeklickt**
- gehe zu x: 200 y: -130
- zeige Richtung -90
- komme nach vorn
- sage **Steuere mich mit dem Mauszeiger zur Prinzessin!** für 2 Sek.
- gehe zu **Teppich**
- sage **Aber ich muss immer auf dem Teppich bleiben!** für 2 Sek.
- wiederhole bis **wird Prinzessin berührt?**
 - falls **wird Teppich berührt?**
 - gehe zu **Mauszeiger**
 - sonst
 - sage **Nicht geschafft! Versuch's nochmal!**
 - stoppe dieses Skript
- spiele Klang **Triumph**
- gleite 3 Sek. zu x: 33 y: 106

Teppich

- Wenn **angeklickt**
- gehe zu x: 200 y: 100
- warte 4 Sek.
- wiederhole fortlaufend
 - drehe 1 Grad
 - gehe 4 -er Schritt
 - pralle vom Rand ab

Prinzessin...

- Wenn **angeklickt**
- gehe zu x: -200 y: -130
- warte bis **wird Prinz berührt?**
- gleite 3 Sek. zu x: 15 y: 105

Music Loops

 Der Prinz erklärt zuerst das Spiel und folgt dann fortlaufend dem Mauszeiger, bis er die Prinzessin berührt (und mit ihr entschwebt). Dabei musst du den Prinzen immer auf dem Teppich halten, der in Kreisbögen über die Bühne fliegt und jeweils am Rand abprallt.



sage , um eine solche Sprechblase zu löschen!

Setze das Objekt an den Zielort und wechsle zum Menü „Bewegung“: Die Koordinaten sind schon eingetragen!