

9TE / PO Informatique

Logique Informatique

Programmieren mit
ScRATCH



Version 2.0 du 23 mars 2015



ÉDUCATION NATIONALE
Luxembourg

9TE / PO Informatique
Logique Informatique

Programmieren mit
The Scratch logo consists of the word "SCRATCH" in a stylized, blocky font. The letters are primarily orange with a white outline, and they have a slight shadow or glow effect.

ScRATCH

SCRATCH

Einleitung	6
A. Die Scratch-Umgebung	7
A.1. Das Scratch Programmfenster	7
A.2. Hauptelemente eines Scratchprojektes	8
A.2.1 Die Bühne	8
A.2.2 Sprites	8
A.2.3 Befehle	9
A.2.4 Programme	9
A.3. Sprites bearbeiten	12
A.4. Sprites haben Kostüme	14
A.5. Die Bühne	17
A.6. Projekt dokumentieren	18
B. Scratch-Projekte mittels Sequenzen entwickeln	19
C. Auf Ereignisse reagieren	21
D. Die Endlosschleife	23
E. Entscheidungen treffen	26
F. Bedingte Schleifen	29
G. Nachrichten senden und empfangen	31
H. Steuerung anhand einer Schleife	34
H.1. Zeichnen eines Quadrats	35
H.2. Regelmäßige Vielecke (Polygone)	36
H.3. Einfache Mandalas	37
H.4. Einige einfache, zusammengesetzte Mandalas	40
H.5. Erstellen von Kreisen	41
H.6. Erstellen von Kreisen (für Fortgeschrittene)	42
I. Variablen und mathematische Ausdrücke	43
I.1. Variablen	43
I.1.1 Variablen erzeugen und anzeigen	43
I.1.2 Mit Variablen arbeiten	44
I.1.3 Vordefinierte Variablen	47
I.2. Ausdrücke	47
I.2.1 Wahr oder falsch	50
I.2.2 Für Fortgeschrittene: Zusammengesetzte Bedingungen	52
J. Prozeduren	53
J.1. Prozeduren definieren	53
J.2. Prozeduren aufrufen	54
J.3. Argumente und Parameter	55



Einleitung

Scratch ist eine neue Programmiersprache, die es leicht macht, die Programmierung eines Computers zu erlernen. Mit ihr kannst du kreative interaktive Geschichten, Spiele und Animationen erstellen.

Das Programm Scratch kannst du gratis auf der Webseite scratch.mit.edu herunterladen. Auf dieser Webseite findest du auch eine Fülle von Informationen über Scratch (Handbücher, Videoanleitungen, Scratchkarten, Frequently Asked Questions (FAQs) usw), sowie eine online Version mit der du Scratch-Projekte erstellen kannst. Du kannst dich auch auf der Scratchseite anmelden und dort deine Projekte veröffentlichen.

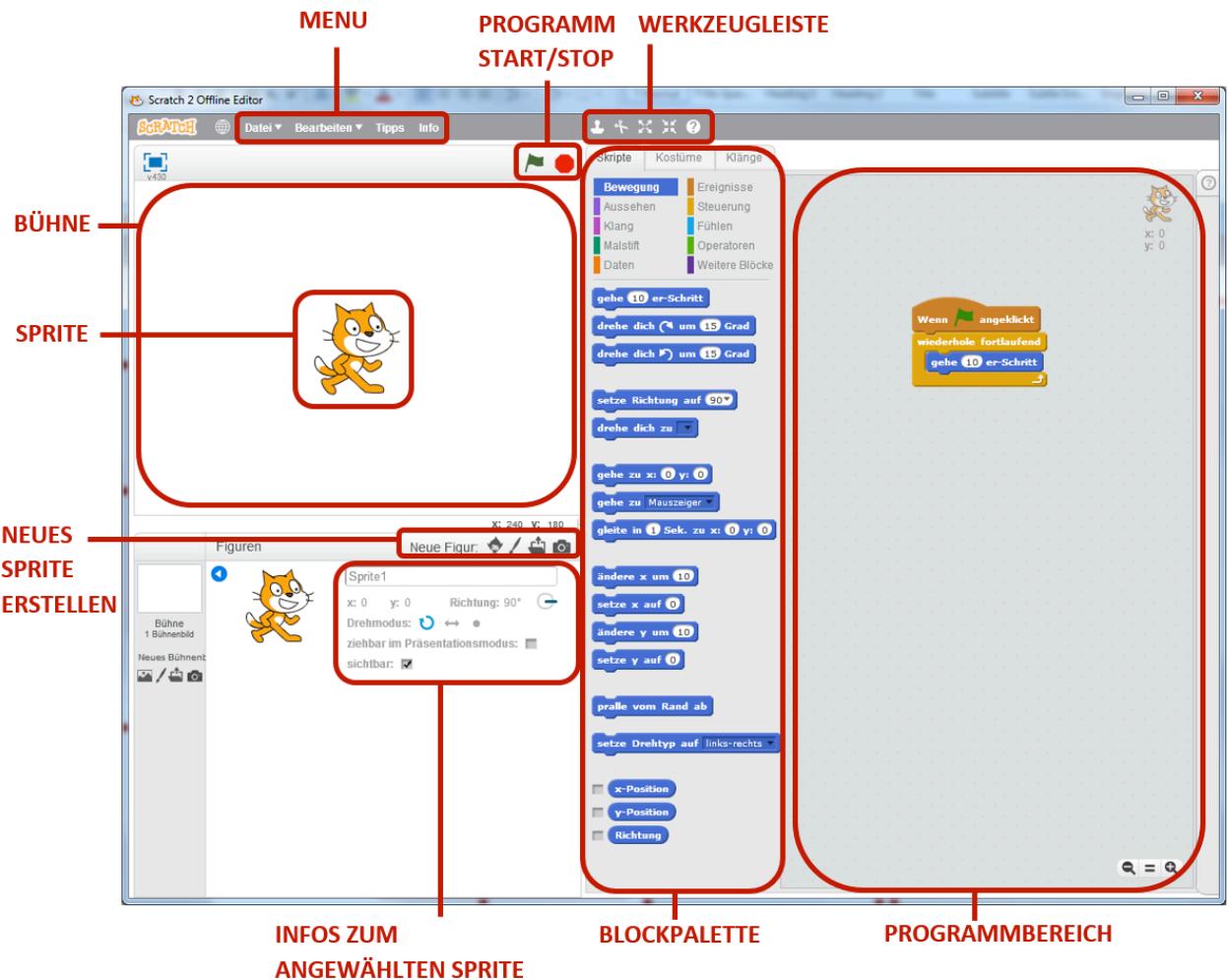
Dieses Skript führt dich schrittweise in die Programmierung mit Scratch ein.

SCRATCH

A. Die Scratch-Umgebung

In diesem Kapitel lernst du die Umgebung kennen mit der du Scratch-Projekte erstellen kannst.

A.1. Das Scratch Programmfenster





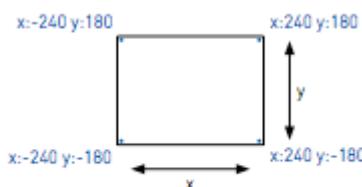
A.2. Hauptelemente eines Scratchprojektes

A.2.1 Die Bühne

Die Bühne ist der Platz, wo Geschichten, Spiele und Animationen zum Leben erwachen.

Sprites bewegen sich und interagieren miteinander auf der Bühne.

Die Bühne ist 480 Einheiten breit und 360 Einheiten hoch. Sie ist in einem XY-Koordinatensystem eingeteilt. Der Mittelpunkt der Bühne hat eine X-Koordinate von 0 und eine Y-Koordinate von 0.



A.2.2 Sprites

Scratch Projekte setzen sich aus Objekten, so genannten Sprites (A.d.U.: ein Sprite ist wortwörtlich ein Kobold, bedeutet aber hier einfach eine Computergrafik) zusammen. Sprites werden mitunter auch **Objekte** genannt.

Du kannst das Aussehen eines Sprites verändern und zum Beispiel ein Sprite wie eine Person aussehen lassen, wie einen Zug, einen Schmetterling oder irgendetwas anderes.

Sprites haben einen Namen. Scratch gibt den Sprites automatisch Namen wie Objekt1, Objekt2, usw. Um dich besser in deinem Projekt zurechtzufinden, sollst du den Sprites **aussagekräftige Namen** geben.





A.2.3 Befehle

Du kannst das Sprite Befehle ausführen lassen, zum Beispiel: ihm sagen sich zu bewegen, Musik abzuspielen oder ihn mit anderen Sprites interagieren lassen. Um dem Sprite die gewünschten **Befehle** mitzuteilen, musst du graphische **Blöcke** in den Programmreich ziehen.

Diese Befehlsblöcke befinden sich in den Blockpaletten. Ein Befehlsblock hat, je nach Art des Befehls, eine andere Farbe und befindet sich in einer anderen Blockpalette. So sind beispielsweise alle Befehle die einen Sprite bewegen dunkelblau und befinden sich in der Palette **Bewegung**.

Hier die verschiedenen Blockpaletten :



Außerdem unterscheiden sich die Befehlsblöcke durch ihre Form.

A.2.4 Programme

Ein Programm ist eine Folge von Befehlen welche nacheinander ausgeführt werden. In Scratch erstellst du ein Programm indem du die benötigten Befehlsblöcke, ähnlich wie bei einem Puzzle, zusammenfügst.

Ein Sprite kann mehrere Programme enthalten, welche auch als Skript(e) bezeichnet werden.

Wenn du ein Programm doppelklickst, führt Scratch die Blöcke von oben bis unten aus.

Um einen Befehlsblock, welchen du nicht mehr benötigst, zu entfernen, ziehe ihn in eine Blockpalette zurück.

SCRATCH

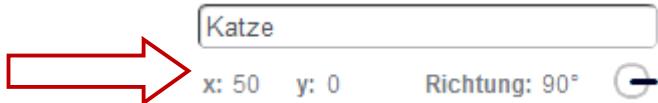
Aufgabe A-01 Mein erstes Programm

Ziel:

Erstellen eines Programms in Scratch.

Beschreibung

- Starte das Programm Scratch.
- Benenne das Sprite um in **Katze**.
- Ziehe den Block  in den Programmablaufbereich. Doppelklicke auf diesen Block. Was stellst du fest? Beobachte die Position des Sprites auf der Bühne! Erkläre!



- Erstelle folgendes Programm :



- Starte das Programm indem du es doppelklickst.
- Stoppe das Programm.
- Wie kannst du das Programm noch starten?

ScRATCH

- h) Ändere den Drehmodus des Sprites und probiere die verschiedenen Möglichkeiten aus:

Erkläre die verschiedenen Drehmodi:







Welcher Drehmodus ist für dieses Projekt am besten geeignet?

- i) Speichere das Projekt unter dem Namen **Aufgabe A-01** ab.



A.3. Sprites bearbeiten

Um ein Sprite auf eine andere Stelle auf der Bühne zu setzen, kannst du es mit gedrückter Maustaste **bewegen**.

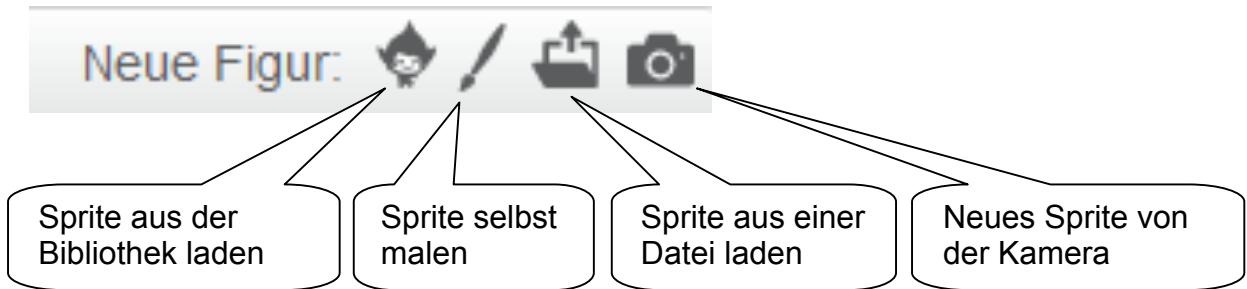
Du kannst Sprites **duplicieren**, **löschen**, **vergrößern** und **verkleinern**: in der Werkzeugeiste die gewünschte Option auswählen und auf das Sprite klicken.



Du kannst ein Sprite auch **drehen** indem du mit gedrückter Maustaste die Richtung des Sprites änderst.



Um weitere Sprites in deinem Projekt zu **erstellen** gibt es mehrere Möglichkeiten:





Aufgabe A-02 Weitere Sprites

Ziel:

Erstellen und bearbeiten neuer Sprites in einem Projekt.

Beschreibung

- a) Öffne das Projekt aus der **Aufgabe A-01** und speichere es unter dem Namen **Aufgabe A-02** ab.
- b) Erstelle neue Sprites und platziere sie auf der Bühne wie unten abgebildet. Den Affen kannst du aus der Bibliothek laden, den Smiley sollst du selbst zeichnen.
- c) Gib den neuen Sprites Namen.
- d) Speichere das Projekt.

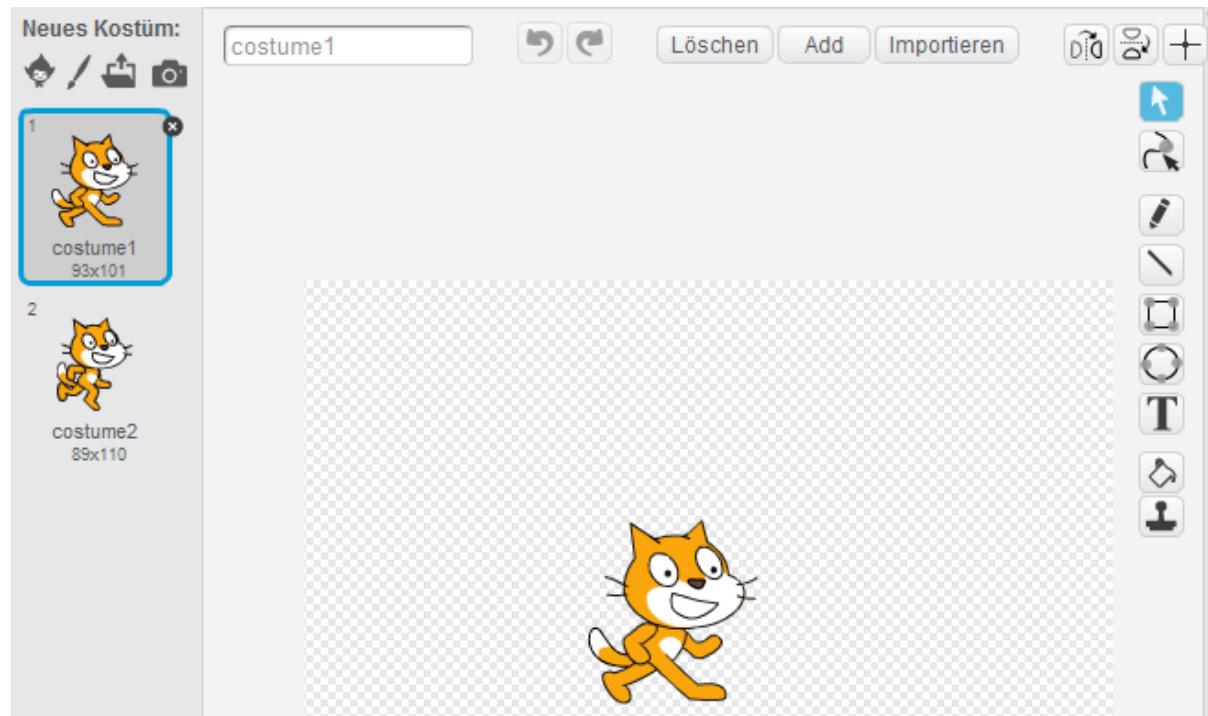




A.4. Sprites haben Kostüme

Sprites können verschiedene Kostüme haben und deshalb ihr Aussehen verändern.

Um dir die Kostüme eines Sprites anzusehen klicke in der Spriteliste auf das gewünschte Sprite und klicke anschließend auf den Reiter **Kostüme**.



Die Katze hat beispielsweise zwei Kostüme. Das aktuelle Kleid ist hervorgehoben. Um zu einem anderen Kostüm zu wechseln, klicke einfach auf das Vorschaubild des Kostüms.

SCRATCH

Es gibt vier Möglichkeiten um neue Kostüme zu erstellen :

- Klicke , um ein neues Kostüm mit dem Paint Editor zu entwerfen.
- Klicke , um eine Bilddatei, die sich auf deiner Festplatte befindet, zu importieren.
- Klicke , um eine Bilddatei, die sich in der Scratch-Bibliothek befindet, zu importieren.
- Falls dein Rechner mit einer Webcam ausgestattet ist, klicke  um neue Kostüme zu fotografieren.

Scratch kann viele verschiedene Bildformate verarbeiten, zum Beispiel: JPG, GIF, BMP und PNG.

Du kannst die Reihenfolge der Kleider ändern, indem du die Vorschaubilder bewegst.

Rechts im Bild befindet sich eine Werkzeugleiste mit der du ein angewähltes Kostüm im Vektor-Modus bearbeiten kannst.

Vektorgrafiken setzen sich aus geometrischen Figuren zusammen. Sie wirken schärfer als Rastergrafiken und können ohne Qualitätsverlust vergrößert werden.



Falls du das Kostüm in eine Rastergrafik umwandelst, erscheint auf der linken Seite eine Werkzeugleiste für Rastergrafiken.

Rastergrafiken bestehen aus einzelnen Bildpunkten und eignen sich zur Darstellung komplexerer Bilder wie Fotos.



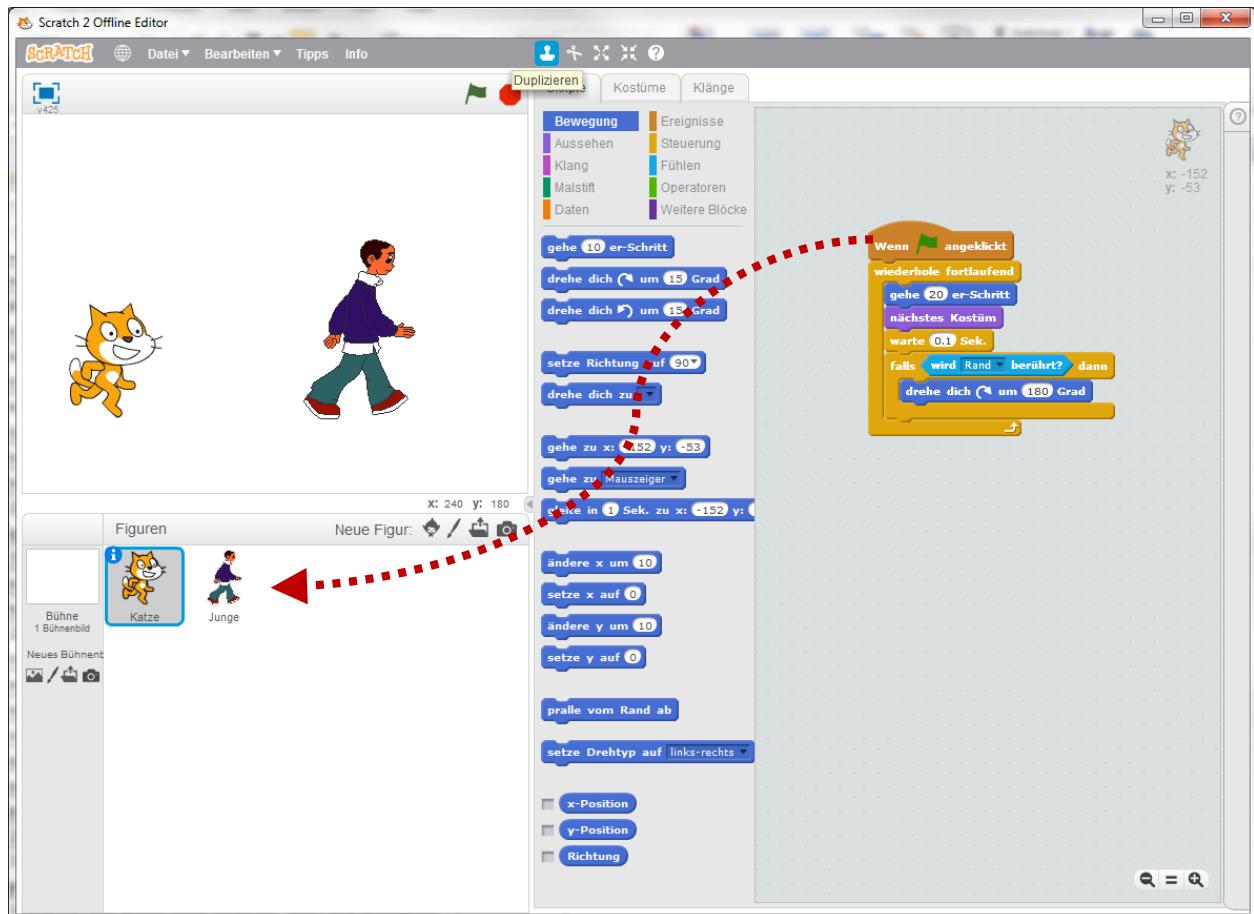


Aufgabe A-03 Kostüme

Ziel: Kostüme bearbeiten.

Beschreibung

- Öffne das Projekt aus der **Aufgabe A-01** und speichere es unter dem Namen **Aufgabe A-03** ab.
 - Lade das Sprite **Boy3 Walking** von der Festplatte und platziere es auf der Bühne wie unten abgebildet.
 - Nenne das Sprite **Junge**.
 - Wie viele Kostüme hat der Junge?
-
- Kopiere das Programm des Sprites **Katze** in das Sprite **Junge** (Programm aus dem Programmablaufbereich in das Sprite **Junge** in der Spriteliste ziehen (siehe Abbildung)).
 - Starte nun beide Programme (Klick auf die grüne Flagge).
 - Speichere das Projekt.





A.5. Die Bühne

Genau wie die Sprites mit einem Kostümwechsel ihr Aussehen verändern können, kann die Bühne ihr Aussehen verändern, indem man die **Hintergründe** wechselt.

Außerdem kannst du die Bühne auch programmieren.

Um Hintergründe und Programm der Bühne zu verändern, klicke auf das Bühnensymbol, das sich links von der Spriteliste befindet.

Aufgabe A-04 Bühne – Hintergründe

Ziel:

Hintergründe der Bühne bearbeiten.

Beschreibung

- Öffne das Projekt aus der **Aufgabe A-03** und speichere es unter dem Namen **Aufgabe A-04** ab.
- Ändere den Hintergrund der Bühne wie unten abgebildet.
- Speichere das Projekt.





A.6. Projekt dokumentieren

Es ist immer eine gute Idee, seine Projekte ausreichend zu dokumentieren. Wenn du im Nachhinein ein Projekt ändern oder erweitern möchtest, hilft dir eine gute Dokumentation dich besser zurecht zu finden.

Mit den **Kommentaren** kannst du einzelne Programmteile dokumentieren. Dies ist besonders bei komplexeren Programmen interessant, um nicht die Übersicht zu verlieren.

Klicke mit der rechten Maustaste auf den Block im Programmablauf welchem du eine Anmerkung hinzufügen möchtest. Wähle im Kontextmenü **Kommentar hinzufügen** und gib einen Kommentar ein.



Aufgabe A-05 Projektdokumentation

Ziel:

Projekt dokumentieren

Beschreibung

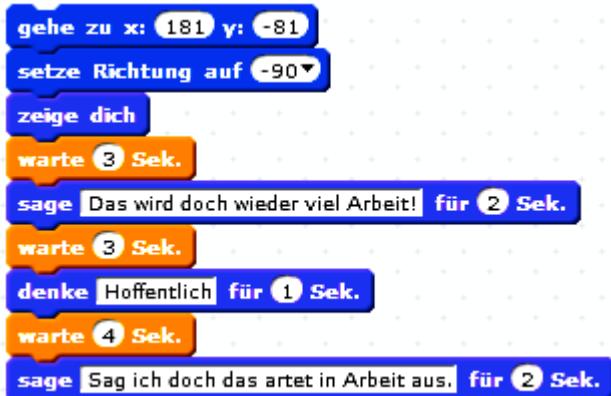
- Öffne das Projekt aus der **Aufgabe A-04** und speichere es unter dem Namen **Aufgabe A-05** ab.
- Versuche herauszufinden wie das Programm des Sprites **Katze** funktioniert und kommentiere jeden Befehlsblock.
- Speichere das Projekt.



B. Scratch-Projekte mittels Sequenzen entwickeln

Die einfachste Form der Programmstruktur ist die Sequenz. Mehrere Anweisungen werden nacheinander ausgeführt.

Beispiel einer Sequenz in Scratch:



Aufgabe B-01

Ziel:

Projekte mittels Sequenzen erstellen.

Beschreibung

- Erstelle ein neues Projekt und speichere es unter dem Namen **Aufgabe B-01** ab.
- Die Katze soll:
 - sich zur Position x=0, y= -100 begeben
 - nach links schauen
 - während 2 Sekunden "Hallo" sagen.
- Speichere das Projekt.

SCRATCH

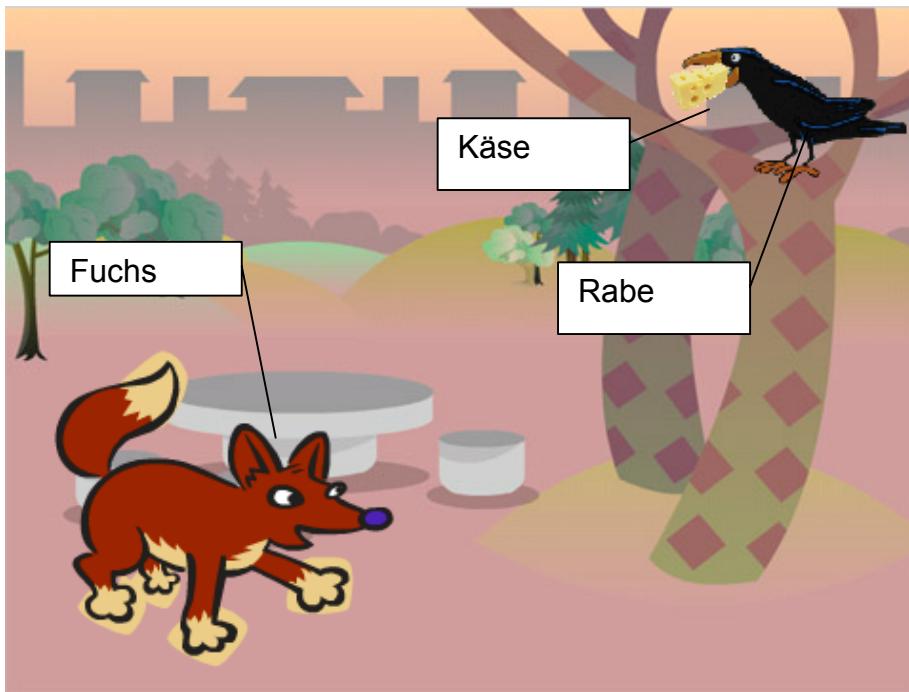
Aufgabe B-02 "Le corbeau et le renard!"

Ziel: Projekte mittels Sequenzen erstellen.

Beschreibung

- Erstelle ein Projekt mit folgenden Sprites (siehe Bild): Käse, Rabe und Fuchs. Käse und Rabe sind jedoch in der Bibliothek nicht vorhanden, zeichne sie selbst oder lade sie aus dem Internet herunter.
- Der Fuchs soll nacheinander folgende Sätze aufsagen (ein Befehl pro Satz):

Et Bonjour Monsieur du Corbeau,
Que vous êtes joli! Que vous me semblez beau!
Sans mentir, si votre ramage se rapporte à votre plumage
Vous êtes le phénix des hôtes de ces bois



- Anschließend soll der Rabe den Text „Merci“ sagen, wonach das Käsestück auf den Boden fällt.
- Startbedingungen: Wenn man erneut auf die grüne Flagge klickt, muss der Käse wieder in seiner ursprünglichen Position, d.h. im Mund des Raben erscheinen. Anders ausgedrückt muss am Anfang des Programms der Käse mit dem Befehl „gehe zu“ in diese Position gebracht werden. Man spricht in diesem Fall von Initialisierung. Diese Voreinstellungen werden ab jetzt in allen Aufgaben erfordert.

Bemerkungen:

- Überlege dir vorher, wie du die zeitliche Reihenfolge der Aktionen der 3 Sprites aufeinander abstimmen kannst.
 - Der Käse fällt am schönsten mit einer „gleite zu ...“ Bewegung.
- Speichere das Projekt unter dem Namen **Aufgabe B-02** ab.



C. Auf Ereignisse reagieren

Ereignisse werden mit Blöcken symbolisiert, die man als „Hüte“ bezeichnet. Diese Blöcke haben abgerundete Oberkanten, wie zum Beispiel:



Die angedockten Befehle werden ausgeführt, wenn die grüne Flagge angeklickt wird.



Die angedockten Befehle werden ausgeführt, wenn das angegebene Sprite angeklickt wird.



Die angedockten Befehle werden ausgeführt, wenn die angegebene Taste gedrückt wird.

Diese Blöcke werden als oberste Blöcke in Programmen platziert. Sie warten darauf, dass ein Ereignis geschieht (z.B. wenn eine bestimmte Taste gedrückt wird). Daraufhin werden dann die Befehlsblöcke unterhalb vom Hutblock ausgeführt.

Aufgabe C-01

Ziel:

Auf Mausklick reagieren

Beschreibung

- Erstelle ein Projekt mit einem Sprite, das eine volle Flasche und ein leeres Glas darstellt. Da dieses Sprite nicht als Datei existiert, musst du es entweder selbst zeichnen oder aus dem Internet herunterladen. TIPP: Flasche und Glas sollen **in einem einzigen Sprite** zusammengefasst werden. Dieses Sprite soll mindestens drei Kostüme haben.
- Wenn man auf das Sprite klickt, wird die Flasche nach und nach leer und das Glas nach und nach voll. Wenn die Flasche leer ist, fängt das Ganze von vorne an.



- Speichere das Projekt unter dem Namen **Aufgabe C-01** ab.



Aufgabe C-02

Ziel:

Auf Tastaturereignisse reagieren

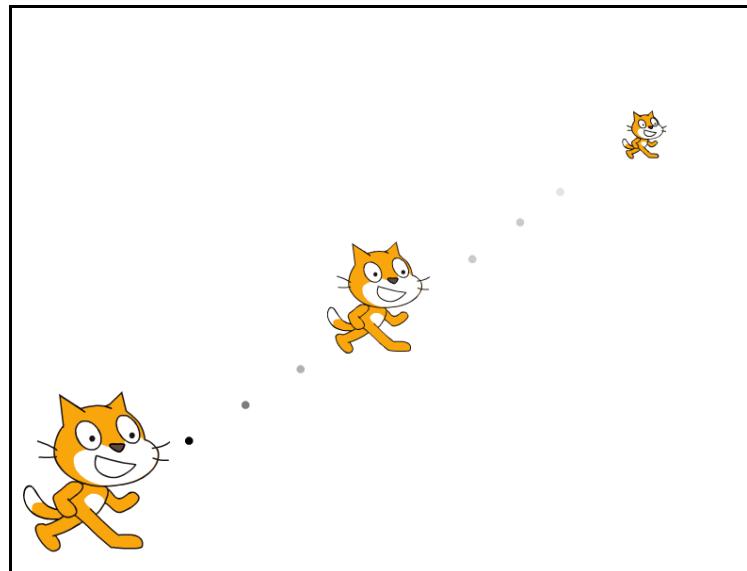
Beschreibung

- a) Erstelle ein Projekt, in dem sich die Katze beim Drücken der vier Pfeiltasten in die jeweilige Richtung bewegt.
- b) Speichere das Projekt unter dem Namen **Aufgabe C-02** ab.

Erweiterung für Fortgeschrittene

Taste « H » : Die Katze bewegt sich quer über den Bildschirm (von unten links nach rechts oben). Die Katze wird immer kleiner. Damit erzeugt man den Anschein von Tiefe.

Taste « V » : Die Katze bewegt sich quer über den Bildschirm (von rechts oben nach unten links). Die Katze wird immer größer.





D. Die Endlosschleife

In dieser Einheit werden wir uns mit dem Steuerungsblock „**wiederhole fortlaufend**“ beschäftigen, welcher es uns erlaubt, Aktionen endlos zu wiederholen. In der Programmierung nennen wir dies eine „*Endlosschleife*“.



Wenn wir die Programmblöcke aus früheren Aufgaben in diese Endlosschleife einfügen, stellen wir fest, dass diese Blöcke ausgeführt werden bis wir das rote Stop-Icon drücken.

Aufgabe D-01 Die unermüdliche Fledermaus

Ziel:

Benutzen der Endlosschleife.

Beschreibung

- a) Erstelle ein Projekt mit folgendem Sprite und folgender Bühne.



- b) Einstellungen beim Start des Programms:

Die Fledermaus wird in die Mitte des Bildschirmes gesetzt und sie schaut nach rechts.

Der Drehmodus soll über einen Befehl auf „links-rechts“ eingestellt werden.

- c) Die Fledermaus hat zwei Kostüme damit man sie fliegen sieht. Wenn die Startflagge angeklickt wird, soll die Fledermaus auf der Bühne hin- und her fliegen.
- d) Speichere das Projekt unter dem Namen **Aufgabe D-01** ab.

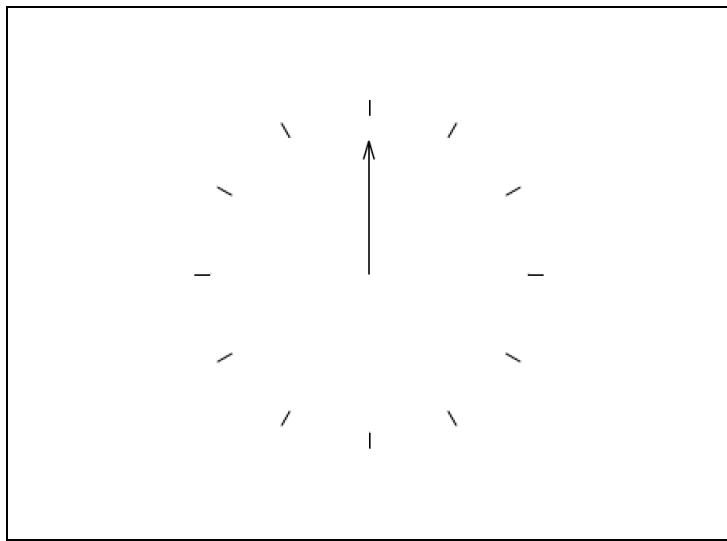
Aufgabe D-02 Eine einfache Uhr

Ziel:

Benutzen der Endlosschleife.

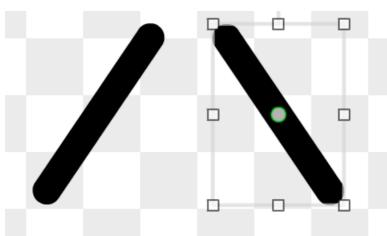
Beschreibung

- Erstelle ein Projekt mit einem neuen Sprite, das einen Uhrzeiger darstellt. Das neue Objekt „Zeiger“ wird mit dem integrierten Graphikprogramm im Vektormodus hergestellt (siehe Vorgehensweise, weiter unten).



Der Dreh- oder Rotationspunkt des Zeigers (seines Kostüms) muss an dessen unteren Ende genau gesetzt werden. Um den Zeigerkopf symmetrisch herzustellen, wird geraten, mit den Werkzeugen „Duplizieren“ und „Links und rechts vertauschen“ zu arbeiten.

Vorgehensweise beim Aufbau der Zeigerspitze



Original

dupliziert und
gedreht



zusammengesetzt

- Einstellungen beim Start des Programms:

Der Zeiger wird in die Mitte des Bildschirmes gesetzt und schaut nach oben.

SCRATCH

- c) Der Uhrzeiger wird animiert. Er macht eine vollständige Drehung (360°) in 60 Sekunden. Der Drehwinkel pro Sekunde beträgt logischerweise $360 : 60 = 6^\circ$. Der Zeiger soll andauernd drehen, sobald die grüne Flagge berührt wird.
- d) Zur besseren Darstellung der Uhr muss das Ziffernblatt noch gezeichnet werden. Folgende Befehlssequenz zeichnet dieses. Die Einteilung des Ziffernblattes wird vom Zentrum, *Punkt (0; 0)* aus gesteuert.



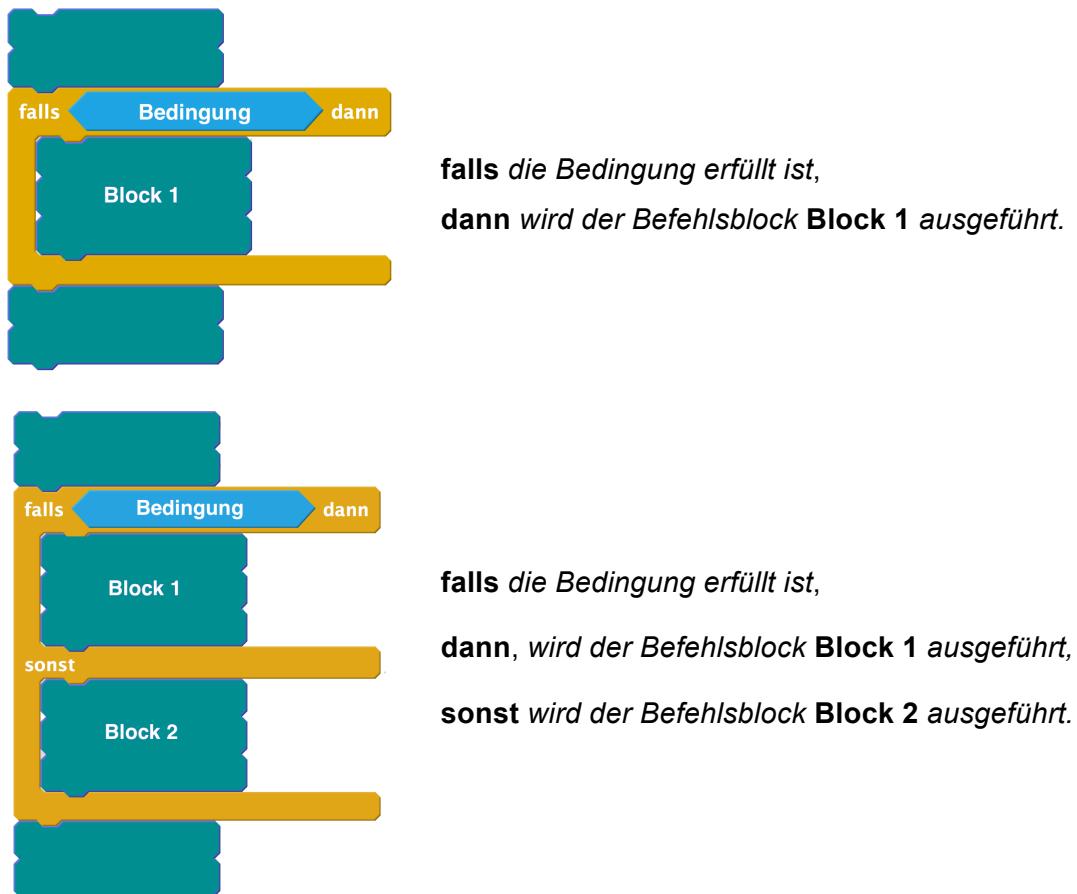
- e) Speichere das Projekt unter dem Namen **Aufgabe D-02** ab

SCRATCH

E. Entscheidungen treffen

Du hast sicher schon bemerkt, dass es in der Programmierung öfters vorkommt, dass man Entscheidungen treffen muss (dies ist im „richtigen Leben“ auch nicht anders). So kannst du dein Sprite veranlassen, dies oder jenes zu tun, je nachdem welche Situation sich stellt.

Die Blöcke, welche hierzu eingesetzt werden können, sind folgende:



Die Bedingung muss jeweils „richtig“ oder „wahr“ sein um die Befehle ausführen zu können, die unter „falls“ stehen.

Es gibt mehrere Arten von Bedingungen, hier sind zwei davon:



so kannst du testen, ob z.B. der Rand oder ein Sprite berührt wurde



so kannst du testen, ob eine Farbe berührt wurde.

Diese Blöcke findest du in der Blockpalette „Fühlen“.

ScRATCH

Aufgabe E-01 Such das Futter!

Ziel: Die Katze sucht das Futter.



Beschreibung:

- a) Einstellungen beim Start des Programms:

Katze (ist auch der Name des Sprites):

- sie befindet sich im linken unteren Bereich des Bildschirms
- sie schaut nach rechts
- ihr Drehmodus ist auf links-rechts eingestellt

Futter (ist auch der Name des Sprites) :

- es befindet sich im rechten unteren Bereich des Bildschirms
- seine Größe wird auf die Hälfte der ursprünglichen Größe gesetzt
- das Futter ist versteckt

- b) Die Katze ist sehr hungrig (und verfressen). Während sie im Zimmer hin- und herläuft, denkt sie „Wo ist das Futter bloß?“. Das Futter erscheint erst beim Betätigen der Leertaste.
- c) Findet sie das Futter, sagt die Katze „Hab ich dich!“ und das Programm endet.
- d) Wenn die Katze das Futter gefunden hat, wird dessen Kostüm (Futter-voll) durch ein zweites Kostüm (Futter-leer), welches du selbst hergestellt hast, ersetzt.
- e) Speichere die Lösung als **Aufgabe E-01** ab.

SCRATCH

Aufgabe E-02 Free the crab!

Ziel: Die Krabbe soll sich aus dem Käfig befreien.

Beschreibung:

- a) Einstellungen beim Start des Programms:

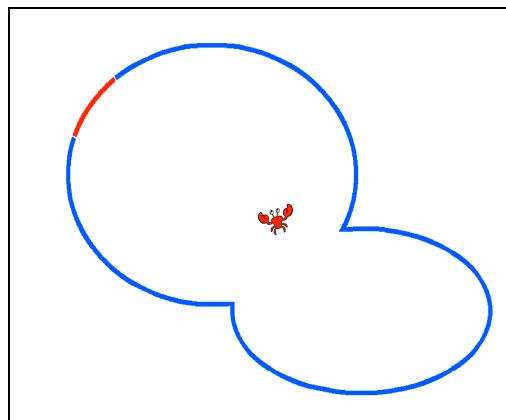
Krabbe (ist auch der Name des Sprites):

- sie befindet sich in der Mitte des Bildschirms
- sie schaut in eine zufällig gewählte Richtung

- b) Benutze das Sprite **Krabbe** aus den Vorlagen



- c) Erstelle einen Bühnenhintergrund („Käfig“) welcher aus einer Farbe besteht. Der Käfig hat einen Ausgang, welcher durch eine andere Farbe dargestellt wird, wie zum Beispiel:



- d) Die Krabbe versucht nun, aus dem Käfig auszubrechen. Trifft sie auf die Randfarbe (blau), so denkt sie „Hmmm...“, dreht sich etwas und versucht es auf ein Neues. Trifft sie auf die Öffnung (rot), so sagt sie „Endlich frei“ und das Programm stoppt 2 Sekunden später.
- e) Speichere dein Programm als **Aufgabe E-02** ab.

Erweiterungen für Fortgeschrittene:

- f) Die 2 Kostüme der Krabbe unterscheiden sich nur wenig. Zeichne ein lustigeres Kostüm, zum Beispiel mit längeren Augen.



- g) Du möchtest mogeln und die Richtung der Krabbe beeinflussen können.

Das Betätigen der „Pfeil nach rechts“-Taste erlaubt dir, die Krabbe nach rechts zu drehen.

Das Betätigen der „Pfeil nach links“-Taste erlaubt dir, die Krabbe nach links zu drehen.

- h) Speichere dein Programm als **Aufgabe E-03** ab.

F. Bedingte Schleifen

Bis heute hast du schon vieles in Scratch gelernt und geübt. Du kannst Befehlsfolgen programmieren, Endlosschleifen und Bedingungen einsetzen.

Beides, Schleifen und Bedingungen kann man auch manchmal kombinieren – du erhältst so eine bedingte Schleife.



Bei diesem Befehl wird überprüft ob die Bedingung wahr ist. Falls ja, geht es mit dem nächsten Befehl hinter der Schleife weiter. Falls nein, wird die Befehlsfolge Befehl 1, Befehl 2, ... innerhalb der Schleife ausgeführt und die Bedingung erneut überprüft. Die Befehlsfolge in der Schleife wird solange ausgeführt bis die Bedingung zutrifft, danach geht es mit dem nächsten Befehl hinter der Schleife weiter.

Aufgabe F-01 Jogging am Strand

Ziel:

Die Katze soll joggen, bis die Leertaste gedrückt wird.



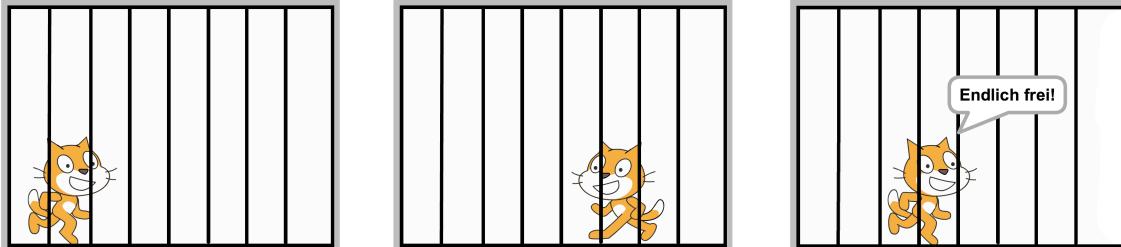
Beschreibung:

- Die Katze joggt am Strand hin und **her**, bis die Leertaste gedrückt wird. Dann bleibt sie stehen und sagt „Uff!“. Benutze die **wiederhole bis** Schleife!
 - Speichere das Projekt als **Aufgabe F-01** ab.
 - Versuche dieselbe Aufgabe mit der Endlosschleife **wiederhole fortlaufend** zu programmieren. Welche Lösung findest du am besten? Begründe!
-
-

SCRATCH

Aufgabe F-02 Freiheit für die Katze

Ziel: Die Katze will aus dem Käfig befreit werden.



Beschreibung:

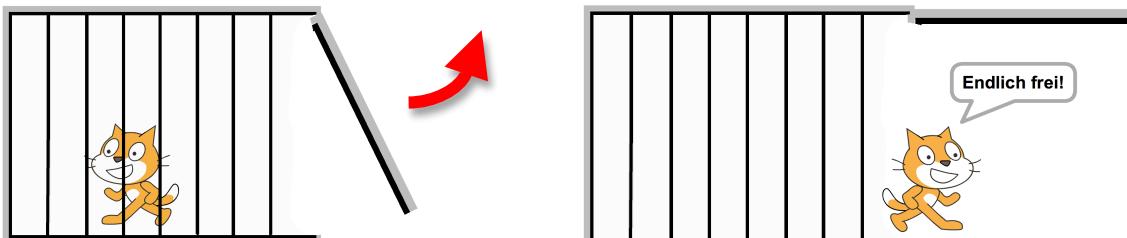
- Die Katze läuft im Käfig hin und her. Am grauen Rand des Käfigs wechselt sie die Richtung. Das Sprite für den Käfig musst du selbst zeichnen, es darf kein Bühnenbild sein.
- Falls die Leertaste gedrückt wird, öffnet sich der Käfig und die Katze dankt für ihre Freiheit.
- Stelle sicher, dass der Käfig sich beim Start des Programms im Vordergrund befindet, dass die Katze sich im Käfig befindet und dass der Käfig geschlossen ist.
- Speichere als **Aufgabe F-02** ab.

Erweiterung:



- Während die Katze im Käfig hin- und herläuft denkt sie alle 10 Sekunden: „Wie komm ich bloß hier raus?“

Erweiterung für Fortgeschrittene:



- Wenn die Leertaste gedrückt wird, öffnet sich die Tür des Käfigs langsam.
- Erst wenn die Katze den Käfig verlassen hat, dankt sie für ihre Freiheit.

Hinweis: Zum Lösen dieser Erweiterung kannst du die zusammengesetzten Bedingungen benutzen (siehe Kapitel I.2.2).



G. Nachrichten senden und empfangen

Es gibt Situationen in denen ein Sprite ein anderes Sprite (oder die Bühne) beauftragen möchte eine Aktion (Programm) auszuführen.

Scratch ermöglicht dies durch Senden und Empfangen von Nachrichten.

Es gibt zwei Befehle um eine Nachricht zu senden:



Sendet eine Nachricht an alle Sprites sowie die Bühne, was diese dann veranlasst Aktionen auszuführen. Der Sender führt anschließend den nächsten Block aus, ohne auf die Fertigstellung der Aktionen zu warten.

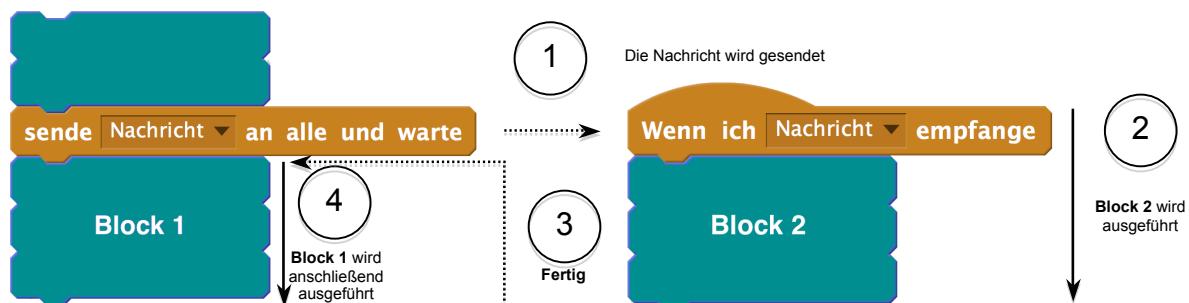
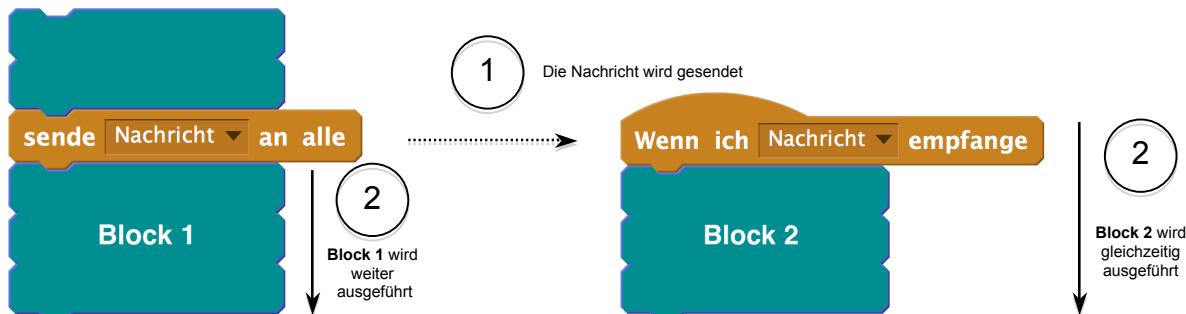


Sendet eine Nachricht an alle Sprites sowie die Bühne, was diese dann veranlasst Aktionen auszuführen. Der Sender wartet bis diese die Aktionen ausgeführt haben, ehe der nächste Block ausgeführt wird.



Wenn ein Sprite oder die Bühne die Nachricht erhält wird das Programm unter diesem Block ausgeführt.

Die folgenden Schemas sollen den Unterschied zwischen den zwei Sende-Befehlen verdeutlichen:



ScRATCH

Aufgabe G-01 „Husch auf den Teppich!“

Ziel:

Senden und Empfangen von Nachrichten.

Beschreibung

- a) Erstelle ein Projekt mit folgenden Sprites : *Oma, Hund und Teppich*



Einstellungen beim Start des Programms:

- Der Hund befindet sich im linken unteren Bereich des Bildschirms
- b) Struppi soll wenigstens zwei Kostüme haben damit man ihn so richtig watscheln sieht.
- c) Wenn die Oma angeklickt wird, sagt sie „Husch auf den Teppich!“ und Struppi soll sich dann sofort auf den Teppich begeben.

Programmiere die Oma und Struppi indem du folgende Befehle benutzt.



- d) Speichere das Projekt unter dem Namen **Aufgabe G-01** ab.



- e) Erweitere das Projekt folgendermaßen: Wenn Struppi auf dem Teppich angelangt ist, sagt Oma „Braver Struppi!“

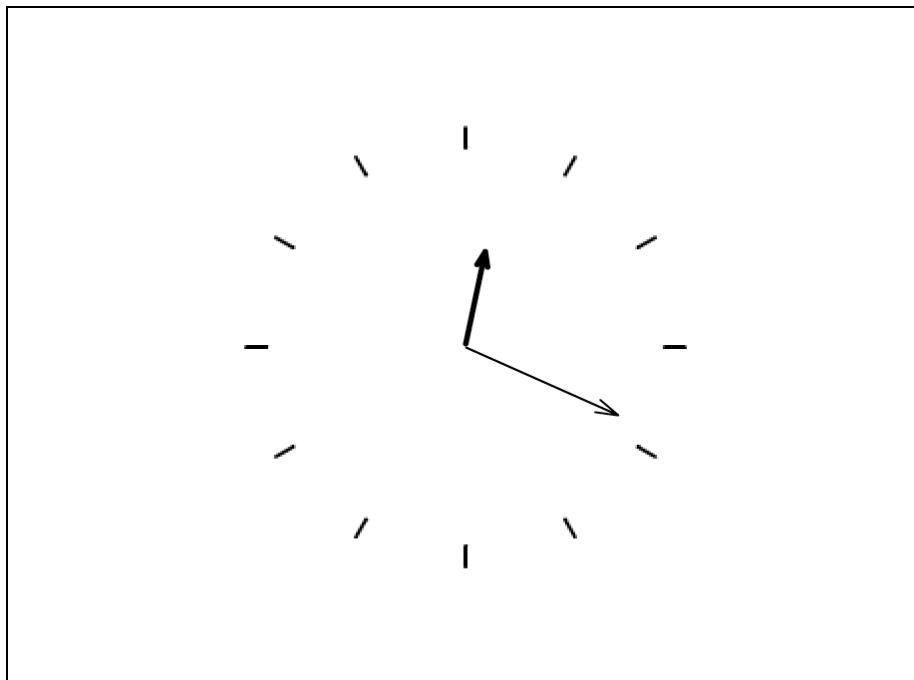
Aufgabe G-02 Eine Uhr mit einem Minutenzeiger

Ziel:

Senden und Empfangen von Nachrichten.

Beschreibung

- a) Öffne das Projekt aus der **Aufgabe D-02** und speichere es unter dem Namen **Aufgabe G-02** ab.
- b) Zeichne einen Minutenzeiger und füge diesen zur Uhr hinzu.
- c) **Funktionsweise** : wenn der Sekundenzeiger wieder auf 12 Uhr zeigt, sendet er eine Nachricht an den Minutenzeiger, dass dieser sich drehen soll.



- d) Speichere das Projekt.

Für Fortgeschrittene:

- e) Speichere die **Aufgabe G-02** unter dem Namen **Aufgabe G-03** ab. Füge einen Stundenzeiger hinzu. Speichere anschließend.

H. Steuerung anhand einer Schleife

In diesem Kapitel lernst du, wie Anweisungen eines Programms kontrolliert mehrmals ausgeführt werden können.

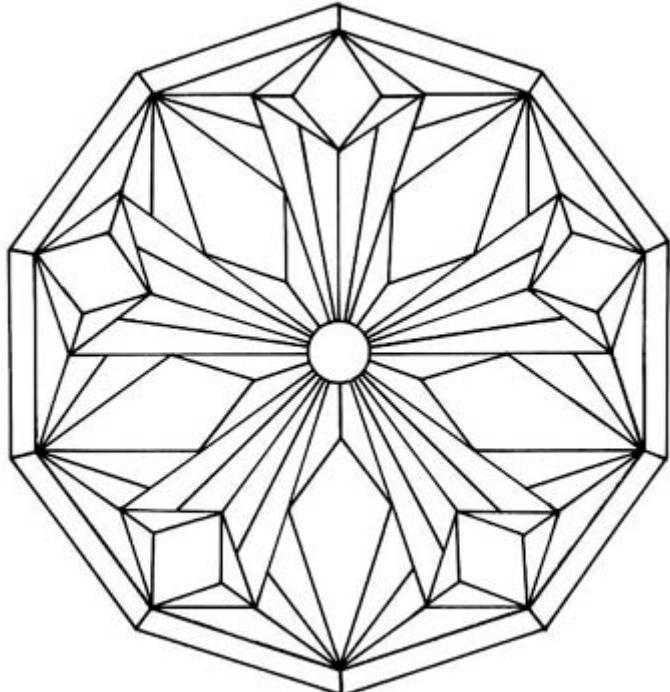
Um diese Einheit zu bewältigen, werden deine Kenntnisse in der Geometrie aufgefrischt (Winkelberechnung: *calcul d'angles* ; Punktsymmetrie / Zentralsymmetrie: *symétrie centrale* ; Drehsymmetrie: *rotation* ; Koordinatensystem: *système de coordonnées, quatre quadrants*).

Mandala

Das Wort Mandala (aus dem Sanskrit) bedeutet so viel wie Kreis und bezeichnet ein kreisförmiges oder quadratisches symbolisches Gebilde mit einem Zentrum, das ursprünglich im religiösen Kontext verwendet wurde (Wikipedia).

Heute denkt man an Ausmalbilder für Kinder und Jugendliche, wenn man von Mandalas redet.

Du wirst in dieser Einheit ein weitaus einfacheres Mandala programmieren.

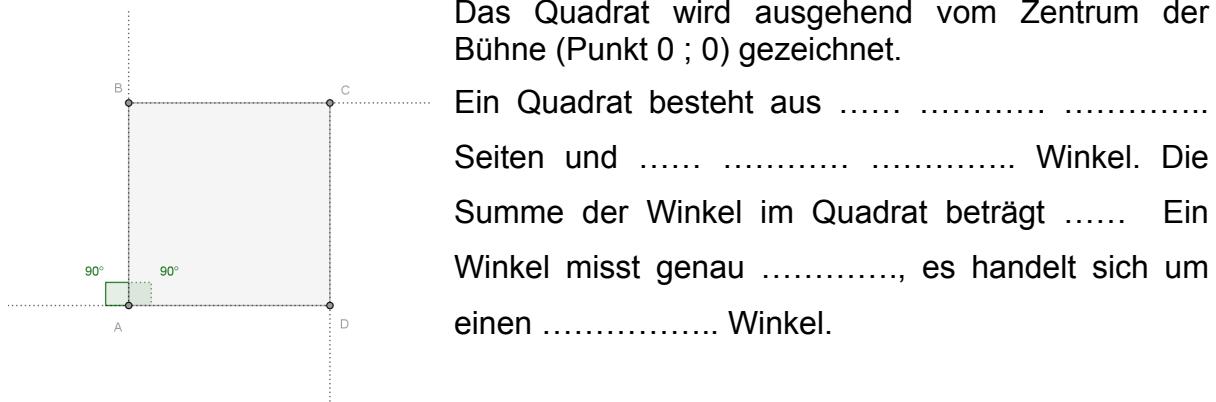


Das Mandala ist um das Zentrum (zentraler Punkt) aus ähnlichen Teilstücken (hier 5, oder genauer 2 x 5) zusammengesetzt.

SCRATCH

H.1. Zeichnen eines Quadrats

Das Quadrat wird ausgehend vom Zentrum der Bühne (Punkt 0 ; 0) gezeichnet.



Ein Quadrat besteht aus Seiten und Winkel. Die Summe der Winkel im Quadrat beträgt Ein Winkel misst genau, es handelt sich um einen Winkel.

Aufgabe H-01 Ein Quadrat

- a) Zeichne ein Quadrat mit einer Seitenlänge von 100 Schritten. Am Ende des Skripts muß das Sprite wieder in die ursprüngliche Richtung zeigen.

Folgende Programmblöcke sind dabei erlaubt:

schalte Stift ein

Wenn grüne Flagge angeklickt

gehe 1 -er Schritt

wische Malspuren weg

drehe 15 Grad

- b) Was stellst du fest?

- c) Speichere als **Aufgabe H-01** ab.

Aufgabe H-02 Ein Quadrat 2

- a) Zeichne erneut ein Quadrat mit einer Seitenlänge von 100 Schritten. Benutze

wiederhole 1 mal

dieses Mal eine **- Schleife**.

Startbedingungen:

- Das Quadrat wird gezeichnet, wenn die Taste „4“ gedrückt wird.
- Der Malstift muss auf Position (0 ; 0) gebracht werden.
- Alle Malspuren werden vorher entfernt.

- b) Speichere als **Aufgabe H-02** ab.

H.2. Regelmäßige Vielecke (Polygone)

„Vielecke können gleichseitig oder gleichwinklig sein. Hat ein Vieleck gleiche Seiten und gleiche Innenwinkel, dann wird es als reguläres oder regelmäßiges Vieleck bezeichnet.“¹

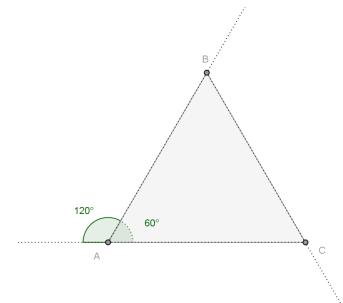
Zeichne regelmäßige Polygone, nachdem du ihre Eigenschaften analysiert und verstanden hast.

Aufgabe H-03 Gleichseitiges Dreieck (triangle équilatéral)

- Öffne das Projekt aus der **Aufgabe H-02** und speichere es als **Aufgabe H-03** ab.
- Dupliziere das vorhandene Skript und ändere es, um ein gleichseitiges Dreieck (Seitenlänge 100) zu zeichnen. Analysiere zuerst, welche Winkel für die Aufgabe wichtig sind.

Startbedingungen:

- Das Dreieck wird gezeichnet, wenn die Taste „3“ gedrückt wird.
 - Der Malstift muss auf Position (0 ; 0) gebracht werden.
 - Alle Malspuren werden vorher entfernt.
- Speichere das Projekt ab.



Aufgabe H-04 n-Eck

Überlege anhand der Kenntnisse, die du in den vorangegangenen Aufgaben gesammelt hast, wie du die beiden leeren Felder ausfüllen musstest, um ein n-eck zu erstellen. Du brauchst diese Aufgabe nicht mit Hilfe von Scratch zu erstellen.

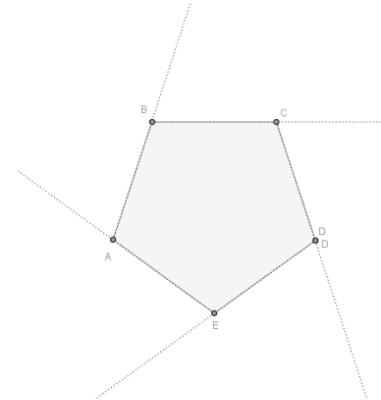
Figur	Anzahl Ecken	Winkel (°)	Gesamt Winkel (°)
Dreieck			
Quadrat			
Pentagon			
n-Eck			

¹ <http://de.wikipedia.org/wiki/Polygon>



Aufgabe H-05 Pentagon

- Öffne das Projekt aus der **Aufgabe H-03** und speichere es als **Aufgabe H-05** ab.
 - Füge ein Skript hinzu, um ein Pentagon (Seitenlänge 100) zu zeichnen.
- Startbedingungen:**
- Das Pentagon wird gezeichnet, wenn die Taste „5“ gedrückt wird.
 - Der Malstift muss auf Position (0 ; 0) gebracht werden.
 - Alle Malspuren werden vorher entfernt.
- Speichere das Projekt ab.



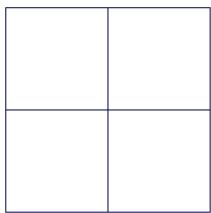
Aufgabe H-06 Regelmäßiges Sechseck (hexagone)

- Öffne das Projekt aus der **Aufgabe H-05** und speichere es als **Aufgabe H-06** ab.
 - Füge ein Skript hinzu, um ein regelmäßiges Sechseck zu zeichnen.
- Startbedingungen:**
- Das Sechseck wird gezeichnet, wenn die Taste „6“ gedrückt wird.
 - Der Malstift muss auf Position (0 ; 0) gebracht werden.
 - Alle Malspuren werden vorher entfernt.
- Speichere das Projekt ab.

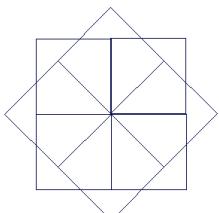
H.3. Einfache Mandalas

Das Mandala, welches jetzt realisiert wird, besteht zunächst aus einer Vielzahl von Quadraten, welche dann um einen zu berechnenden Winkel um das Zentrum rotierend (drehend) gezeichnet werden.

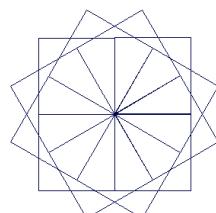
Beispiel:



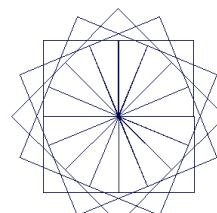
4 Quadrate



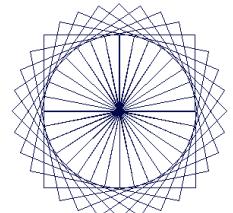
8 Quadrate



12 Quadrate



16 Quadrate



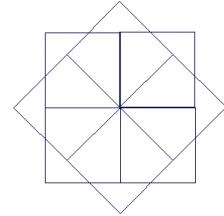
32 Quadrate



Aufgabe H-07 Mein erstes Mandala

Ziel: Ein Mandala aus 8 Quadranten zusammensetzen.

- Erstelle ein neues Projekt und speichere es unter **Aufgabe H-07** ab.
- Zunächst muss der Winkel zwischen 2 benachbarten Quadranten berechnet werden, damit das Mandala symmetrisch aussieht.
- Berechne jetzt den benötigten Winkel für dein Mandala.
Winkel =° / =°
- Füge ein Skript hinzu, um das Mandala mit den 8 Quadranten zu zeichnen.



Startbedingungen:

- Das Mandala wird gezeichnet, wenn die Taste „8“ gedrückt wird.
 - Der Malstift muss auf Position (0 ; 0) gebracht werden.
 - Alle Malspuren werden entfernt.
- Speichere das Projekt ab.

Aufgabe H-08 Zeichnen eines Rahmens

- Öffne das Projekt aus der **Aufgabe H-07** und speichere es als **Aufgabe H-08** ab
- Das Mandala wird hervorgehoben, wenn du es mit einem doppelten Rahmen versiehst.
 - Oberer rechter Punkt des äußeren Rahmens : 230 ; 170
 - Stiftdicke 1
 - Oberer rechter Punkt des inneren Rahmens : 220 ; 160
 - Stiftdicke 2

Startbedingung:

- Der Rahmen wird gezeichnet, wenn die Taste „b“ gedrückt wird.
- Speichere das Projekt ab.

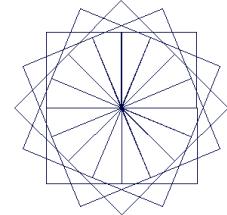


Aufgabe H-09 Mein zweites Mandala

- Öffne das Projekt aus der **Aufgabe H-08** und speichere es als **Aufgabe H-09** ab.
- Füge ein Skript hinzu, um das Mandala mit 16 Quadranten zu zeichnen. Die Berechnung des Winkels erfolgt mit den vorhandenen Operatoren (siehe **Kapitel I.2**).

Startbedingung:

- Dieses Mandala wird gezeichnet, wenn die Taste „m“ gedrückt wird.
- Speichere das Projekt ab.



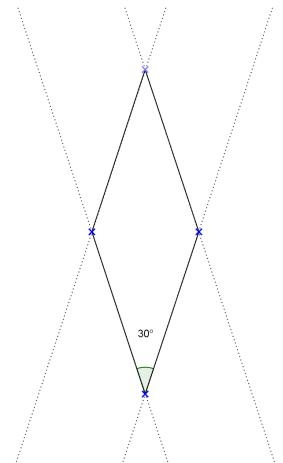
Aufgabe H-10 Ein schöneres Mandala

Das Mandala wird zusehends schöner, wenn es mit Rauten gezeichnet wird.

- Öffne das Projekt aus der **Aufgabe H-09** und speichere es als **Aufgabe H-10** ab.
- Erstelle ein Mandala das aus 12 identischen Rauten besteht. Beginne zuerst mit einer Raute, bestimme die Winkel durch Berechnung und Eintrag auf der beiliegenden Zeichnung.

Startbedingung:

- Dieses Mandala wird gezeichnet, wenn die Taste „r“ gedrückt wird.
- Speichere das Projekt ab.





H.4. Einige einfache, zusammengesetzte Mandalas

Ein Mandala wird aus 2 oder mehreren Mustern zusammengesetzt (gezeichnet).

Aufgabe H-11 Zusammengesetzte Mandalas

- Öffne das Projekt aus der **Aufgabe H-10** und speichere es als **Aufgabe H-11** ab.
- Erstelle das nebenstehende Mandala.

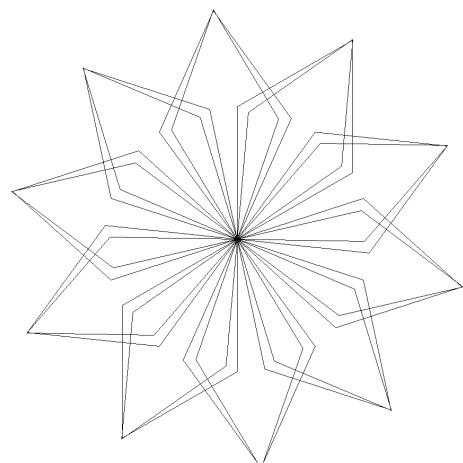
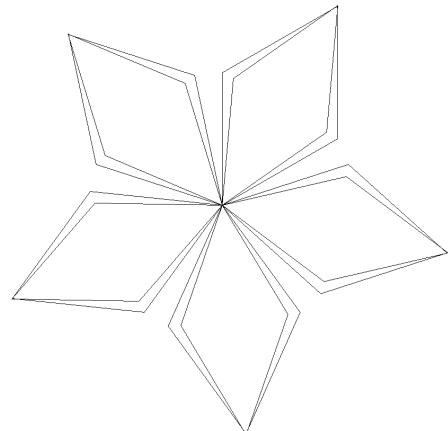
Startbedingung:

- Dieses Mandala wird gezeichnet, wenn die Taste „1“ gedrückt wird.
- Die große Raute hat eine Seitenlänge von 82 und Winkel von 130 beziehungsweise 50.
- Die kleine Raute hat eine Seitenlänge von 80 und Winkel von 140 beziehungsweise 40.
- Das Mandala besteht aus 5 Teilstücken.

- Erstelle anschließend das nebenstehende Mandala.

Startbedingung:

- Dieses Mandala wird gezeichnet, wenn die Taste „2“ gedrückt wird.
- Das Mandala besteht aus 10 Teilstücken. Die Teilstücke sind identisch mit denen aus dem vorigen Mandala.





H.5. Erstellen von Kreisen

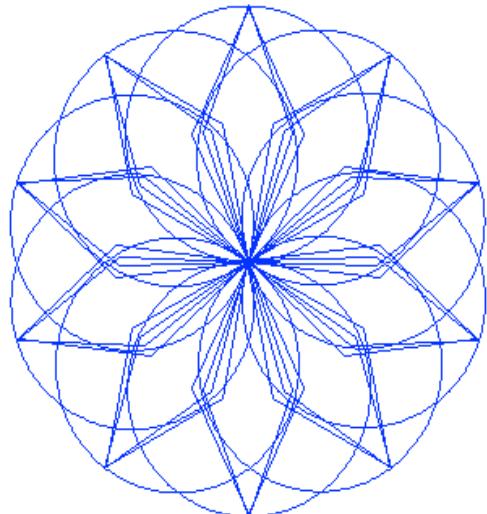
Man kann die Zeichnung von einem Kreis simulieren, indem man eine Figur mit sehr vielen Ecken zeichnet (beispielsweise ein 360-eck, in dem du jedes Mal um 1° drehst).

Aufgabe H-12 Zusammengesetzte Mandalas 2

- Öffne das Projekt aus der **Aufgabe H-11** und speichere es als **Aufgabe H-12** ab.
- Kopiere das Skript des Mandalas, das mit der Taste „a“ gezeichnet wird.
- Modifiziere das Skript, um das nebenstehende Mandala zu erhalten.

Startbedingung:

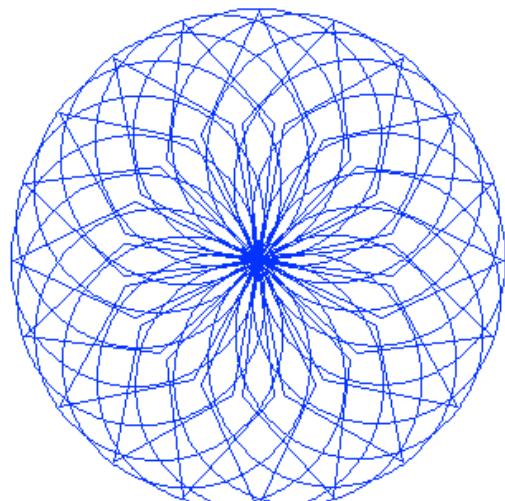
- Dieses Mandala wird gezeichnet, wenn die Taste „c“ gedrückt wird.
- Der Kreis ist ein 60-Eck mit einer Seitenlänge von 8.
- Das Mandala besteht aus 10 Teilstücken.



- Erstelle anschließend das nebenstehende Mandala.

Startbedingung:

- Dieses Mandala wird gezeichnet, wenn die Taste „d“ gedrückt wird.
- Das Mandala besteht aus 20 Teilstücken.





H.6. Erstellen von Kreisen (für Fortgeschrittene)

Das Schwierige beim Zeichnen eines Kreises ist, die Länge ℓ der Teilstücke so zu bemessen, damit dieser mit einem vorgegebenen Radius r entsteht.

Gesamtlänge der 360 Teilstücke (Länge ℓ): _____ (1)

Umfang des Kreises (Radius r): _____ (2)

Wenn du jetzt (1) und (2) gleich setzt, kannst du eine Formel für die Länge ℓ finden:

$$\ell = \text{_____}$$

Aufgabe H-13 Berechnung der Seitenlänge (für Fortgeschrittene)

- a) Öffne ein neues Projekt und speichere es als **Aufgabe H-13** ab.
- b) Berechne die Seitenlänge eines Teilstückes wenn der Kreis einen ein Radius von 50 haben soll.
- c) Prüfe zeichnerisch, ob der Radius stimmt.
- d) Speichere dein Projekt ab.



I. Variablen und mathematische Ausdrücke

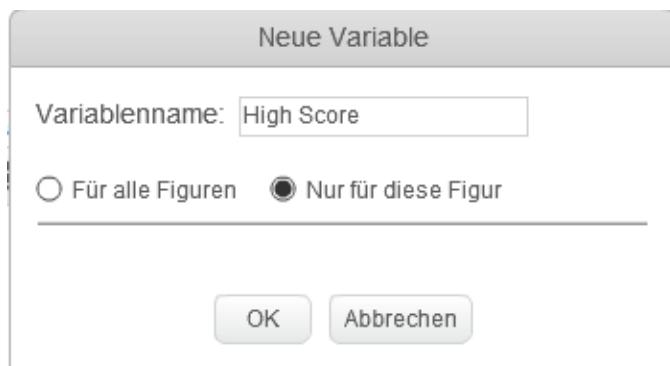
I.1. Variablen

Sehr oft muss man sich irgendwelche Dinge merken: Eine Hausnummer, den Pin-Code eines Telefons, ein Passwort, den Namen eines Telefonpartners, wie viel Geld man noch in der Tasche hat, was man am Nachmittag noch alles erledigen muss, usw. Anstelle von Notizzettelchen oder Knoten im Taschentuch benutzt man in der Programmierung zu diesem Zweck so genannte „Variablen“.

Variablen ermöglichen es uns Werte zu speichern, zu lesen und zu verändern. In Scratch können Variablen verschiedener Art benutzt werden: ganze Zahlen, Dezimalzahlen oder Texte.

I.1.1 Variablen erzeugen und anzeigen

In der Blockpalette können unter anderem neue Variablen erzeugt, verändert oder gelöscht werden. Beim Erzeugen einer neuen Variablen müssen wir der Variablen zuerst einen aussagekräftigen Namen geben:



Wir entscheiden hier auch, welche Objekte in unserem Programm die Variable benutzen und verändern dürfen.

Prinzip:

Im Allgemeinen soll eine Variable immer so privat wie möglich bleiben (Option „Nur für diese Figur“). Kein anderes Objekt soll die Möglichkeit haben unsere Variable absichtlich oder unbeabsichtigt zu verändern. (Wir lassen unsere Adress- und Telefonlisten auch nicht offen in der Mitte des Klassensaales liegen.)

Nur in Ausnahmefällen müssen wir Daten „für alle Figuren“ zur Verfügung stellen.

SCRATCH

Sobald eine Variable erstellt wurde, sehen wir die Variable selbst als Block:



(das Häkchen zeigt an, dass die Variable auf der Bühne sichtbar ist)

Auf der Bühne erscheinen die Variablen als



Normalanzeige



Großanzeige



Regler

Das Aussehen der Variablen kann auf der Bühne durch einen Doppelklick oder einen Klick mit der rechten Maustaste auf die Variable verändert werden.

Spezialfall Regler:

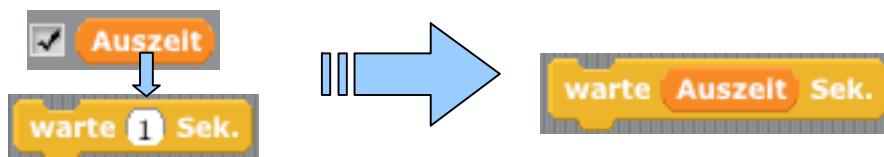
- Regler funktionieren natürlich nur für Zahlen.
- Bei einem Regler kann man mit einem Rechtsklick auch den „Reglerbereich festlegen“, d.h. man kann das Maximum und das Minimum für den Regler einstellen.

I.1.2 Mit Variablen arbeiten

Nachdem die erste Variable erzeugt wurde erscheinen einige Blöcke, die es uns erlauben die Variablen mit unseren Skripten zu ändern:

	Variable auswählen und Wert eingeben (es können auch negative Zahlen, Kommazahlen oder Texte eingegeben werden)
	Variable auswählen und Wert eingeben (es können auch negative Zahlen oder Kommazahlen eingegeben werden – KEINE Texte)
	Variable auswählen: mit diesen beiden Blöcken kann man während des Programmablaufs entscheiden, ob die Variable auf der Bühne sichtbar ist, oder nicht.

In Programmblöcken können Variablen überall eingesetzt werden, wo man bisher Zahlen oder Texte direkt eingetippt hat. Man muss den Variablenblock nur mit der Maus an die richtige Stelle in einen Programmblöck hineinziehen.





Achtung: Texte und Zahlen können nicht beliebig vermischt werden!

- Zahlenvariablen können überall verwendet werden.
- Textvariablen sollte man in den dafür vorgesehenen Stellen einsetzen
(sage [...]... , denke [...]... , frage [...]...)

Aufgabe I-01 Free the crab - reloaded!

Ziel:

Die Krabbe soll mitzählen wie oft sie an die Wände stößt bevor sie sich befreit.

Beschreibung:

- a) Öffne deine **Aufgabe E-02** („Free the crab“) und speichere sie als **Aufgabe I-01** ab.
- b) Füge eine Variable „Zusammenstösse“ hinzu, die jedes Mal erhöht wird, wenn die Krabbe an eine blaue Wand stößt.
- c) Starte das Programm erneut, nachdem die Krabbe den Ausgang gefunden hat und neu positioniert wurde. Wasstellst du fest? Behebe dieses Problem!

Aufgabe I-02 Free the crab - revisited!

Ziel:

Die Krabbe soll sich aus dem Käfig befreien und mitzählen, wie oft sie an die Wände stößt. Die Geschwindigkeit der Krabbe ist einstellbar.

Beschreibung:

- a) Öffne deine **Aufgabe I-01** („Free the crab - reloaded!“) und speichere sie als **Aufgabe I-02** ab.
- b) Füge eine Variable „Auszeit“ hinzu, und stelle sie als Regler dar. Die Werte des Reglers können zwischen 0 und 0.5 verändert werden. Baue die Variable „Auszeit“ so in dein Programm ein, dass sie die Bewegungsgeschwindigkeit der Krabbe bestimmt.

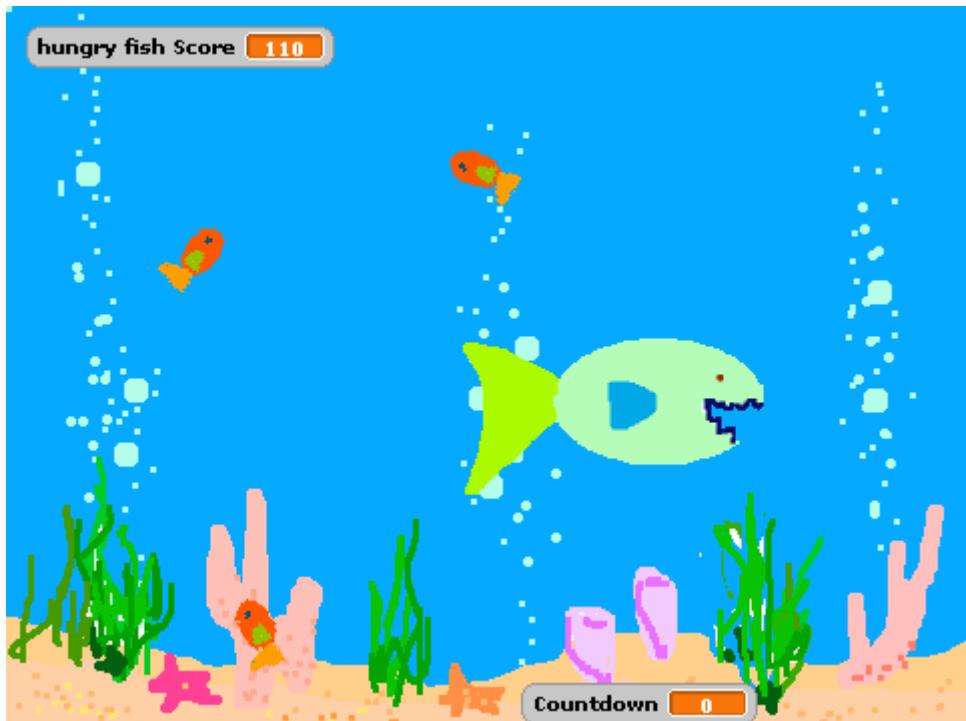
ScRATCH

Aufgabe I-03 Fishchomp - II

Ziel: Ein existierendes Spiel mit Punktestand und Zeitlimit ausstatten.

Beschreibung:

- a) Öffne das Spiel **FishChomp**, das dir vom Lehrer zur Verfügung gestellt wurde und speichere es als **Aufgabe I-03** ab.
Prinzip: der große Fisch folgt der aktuellen Mausposition um die roten Fische zu fressen.
 Siehe dir die Skripte der einzelnen Objekte an und versuche die Funktionsweise des Programms nachzuvollziehen.
- b) Erstelle eine neue Variable „Score“, die bei jedem Verschlucken eines roten Fisches um 10 Punkte erhöht wird. Bei Programmstart muss Score natürlich auf 0 zurückgesetzt werden.
- c) Das Spiel soll zeitlich auf 30 Sekunden begrenzt werden. Erstelle eine Variable „Countdown“, die du beim Programmstart auf 30 setzt. Benutze eine Schleife, um den „Countdown“ jede Sekunde um 1 herabzusetzen. Sind 30 Sekunden abgelaufen, wird das Spiel beendet.
- d) Zeige die Spielanleitung nur während den ersten 5 Sekunden des Spielablaufs.



SCRATCH

I.1.3 Vordefinierte Variablen

Für jedes Objekt gibt es eine Reihe von vordefinierten Variablen, die man genau so verwenden kann, wie die selbst definierten Variablen:

- Blockpalette **Bewegung** : x,y-Position, Richtung (Winkel)
- Blockpalette **Aussehen** : Kostüm Nr. , Bühnenbildname, Größe
- Blockpalette **Klang** : Lautstärke, Tempo
- Blockpalette **Fühlen** : Antwort, Stoppuhr, ... im Moment

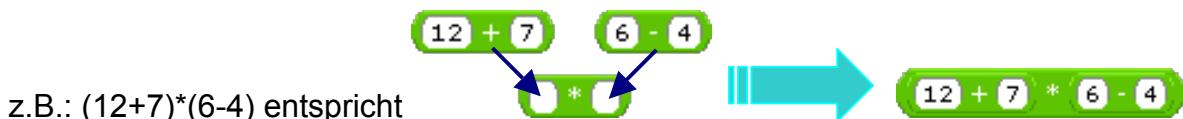
I.2. Ausdrücke

Wie in jeder Programmiersprache kann man in Scratch auch Werte (Zahlen und Texte) vergleichen und mit ihnen rechnen. Man benötigt nicht viele verschiedene Operationsblöcke um eine sehr mächtige Programmiersprache zu erhalten.

Die meisten dieser Blöcke befinden sich in der Blockpalette „**Operatoren**“ und behandeln Zahlen:

	Addition, Subtraktion Multiplikation, Division	für ganze Zahlen und Dezimalzahlen
	Zufallszahl liefern	für ganze Zahlen und Dezimalzahlen
	Rest einer Division	nur für ganze Zahlen sinnvoll
	auf/abrunden einer Zahl	für Dezimalzahlen Resultat: ganze Zahl
	verschiedene mathematische Funktionen	Wichtigste Funktionen: Wurzel: (fr. racine carrée) Betrag: (fr. valeur absolue) Weitere: sin, cos, tan, asin, acos, atan, ln, log, e^, 10^

Man kann Rechenblöcke ineinander einsetzen um komplexere Berechnungen zu erhalten. Dies ist vergleichbar mit den Klammern in der Mathematik:

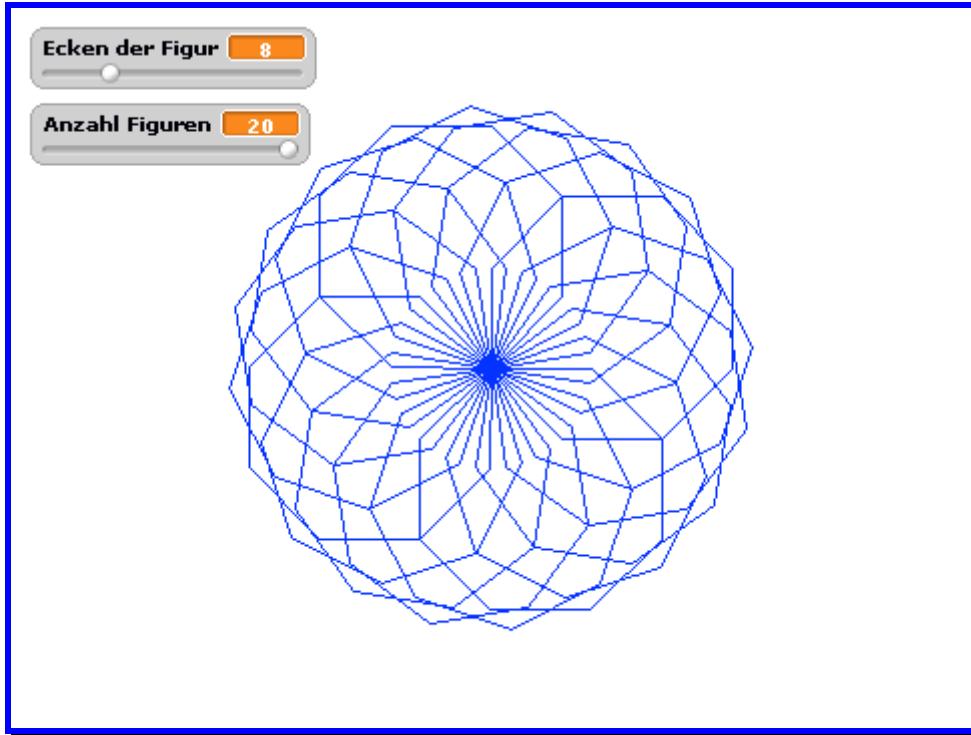


Dabei muss man jedoch darauf achten, dass die Blöcke in der richtigen Reihenfolge eingesetzt werden.

Prinzip: Berechnung von oben nach unten: d.h. der oberste Block wird zuerst berechnet, der unterste zuletzt.

**Aufgabe I-04 Mandala Generator****Ziel:**

Mandalas automatisch berechnen lassen

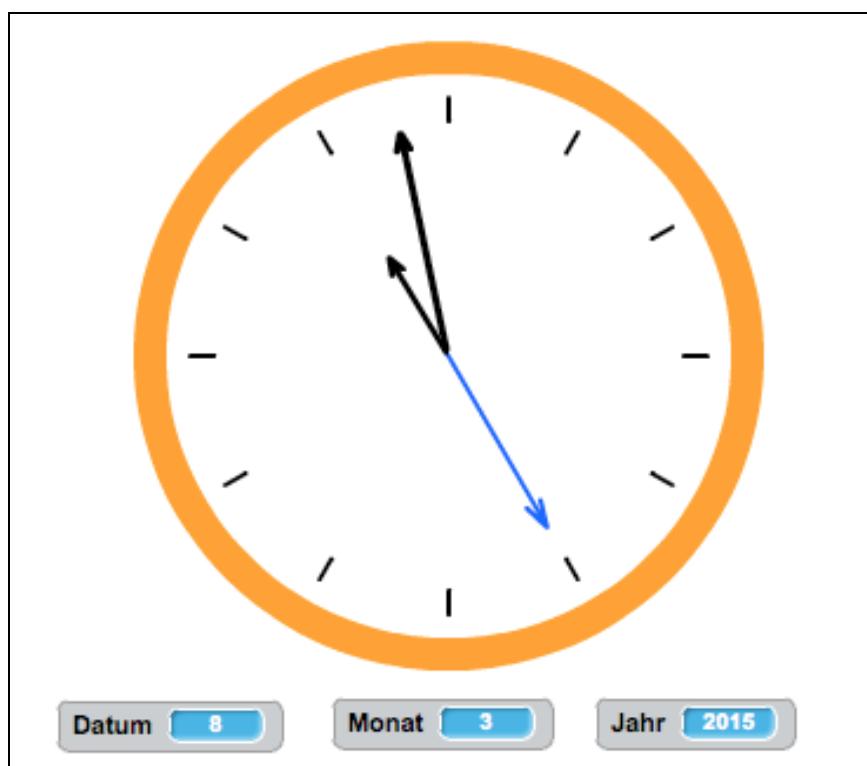
**Beschreibung:**

- a) Definiere 2 Variablen „**Ecken der Figur**“ [3...60]. und „**Anzahl Figuren**“ [0...60], welche du als Regler darstellst.
- b) Wenn die Leertaste (Space) gedrückt wird, soll ein Mandala gezeichnet werden, welches aus so vielen Figuren besteht, wie in der „Anzahl Figuren“ eingestellt wurde. Jede Figur hat so viele Ecken, wie in „Ecken der Figur“ eingestellt wurde.
Lasse auch die Seitenlänge der Figuren automatisch so berechnen, dass die Figuren noch auf den Bildschirm passen.
- c) Verfeinere die Darstellung der Mandalas nach deinem Geschmack (Farbe, Stiftdicke, Farbstärke, übermalen, ...)
- d) Speichere dein Programm als **Aufgabe I-04** ab.

SCRATCH

Aufgabe I-05 Analoge Uhr mit Zeitvariablen (für Fortgeschrittene)

- a) Öffne dein Projekt aus **Aufgabe D-02** und speichere es unter **Aufgabe I-05** ab.
- b) Animiere nun den bestehenden Sekundenzeiger indem du die vorhandene Zeitvariable benutzt. 
- c) Füge sowohl Minuten- als auch Stundenzeiger hinzu.
- d) Zeige auch das Datum an.
- e) Speichere das Projekt.





I.2.1 Wahr oder falsch

Wie ihr im Kapitel E (Entscheidungen treffen) schon gesehen habt, dienen Blöcke mit spitzen Enden dazu eine **Bedingung** zu formulieren.

Um genauer zu sein: Ein Block mit spitzen Enden liefert entweder den Wert „**wahr**“ (Bedingung erfüllt) oder „**falsch**“ (Bedingung nicht erfüllt). Solche Blöcke kann man dann z.B. in „falls <...> ...sonst...“ ; „warte bis <...>“ ; „wiederhole bis <...>“ - Blöcken benutzen.

Die Gruppe „**Operatoren**“ ermöglicht es uns nun, Werte (Zahlen oder Texte) miteinander zu vergleichen um diese zur Steuerung unserer Skripte einsetzen zu können.

Hierzu gehören die Vergleiche:

Beispiele:



Aufgabe I-06 Fishchomp - III

Ziel: Höchste Punktzahl (Highscore) merken und anzeigen

Beschreibung:

- Öffne dein Spiel *FishChomp – II* (**Aufgabe I-03**) und speichere es als **Aufgabe I-06** ab.
- Ändere die Berechnung des Countdowns so, dass du jetzt eine „wiederhole bis“ Schleife benutzen kannst (Dies ergibt eine flexiblere Lösung, da du nun die Spieldauer des Spiels ändern kannst, indem du nur an einer einzigen Stelle den Anfangswert für „Countdown“ änderst).
- Erstelle dann eine Variable „*Highscore*“, die bei Spielende überprüft, ob *Score* den aktuellen *Highscore* übertrifft. Ist das der Fall, wird der aktuelle *Score* als *Highscore* übernommen und ein Klang (deiner Wahl) wird abgespielt.

SCRATCH

Aufgabe I-07

Ratespiel

Ziel:

Eine vom Computer ausgedachte geheime Zahl erraten

Vorgeschichte:

Du musst unbedingt telefonieren, aber der Akku deines Telefons ist leer. Cassy hat ein Telefon dabei, aber sie möchte eigentlich nicht, dass du mit ihrem Telefon anrufst, deshalb stellt sie dir folgende Aufgabe: „*Wenn du meinen vierstelligen geheimen Pin Code in weniger als 15 Versuchen errätst, kannst du telefonieren, solange du willst. Ich sage dir bei jedem Versuch nur, ob mein Pin Code größer oder kleiner ist, als die Zahl, die du rätst.*“



Beschreibung:

Erstelle ein Scratch Programm das Cassy und ihr Ratespiel simuliert:

- Beim Start deines Programms wird in einer für den Benutzer unsichtbaren Variable namens „**Pin**“ eine Zufallszahl zwischen 1000 und 9999 gespeichert. Diese Zahl muss der Spieler dann erraten.
- Die Vorgeschichte kannst du kurz darstellen. (z.B. Cassy telefoniert anfangs, und sagt nach dem Telefongespräch: „Aha, du willst also mit meinem Telefon anrufen“, „Dazu musst du zuerst meinen Pin Code in weniger als 15 Versuchen erraten!“)
- Benutze den Block **Frage Wie ist mein Pin Code? und warte** um eine neue Zahl einzugeben. Diesen Block findest du in der Blockpalette „**Fühlen**“. Der Wert,



der als Antwort vom Spieler eingetippt wurde, steht dann automatisch in der vordefinierten Variablen **Antwort**.

- Eine weitere (sichtbare) Variable namens „**Versuche**“ zeigt an, wie viele Rateversuche du schon unternommen hast.
- Falls du die Pin-Zahl in weniger als 15 Versuchen geraten hast, **ärgert sie sich** und gibt dir das Telefon ☺
- Falls du die Zahl nach 14 Versuchen immer noch nicht erraten hast, **freut sie sich** und tanzt einige Sekunden lang...
- Bilder für Cassy findest du in der Bilderbibliothek von Scratch. Für die Tanzeinlage kannst du Klänge aus der Bibliothek importieren oder eigene Klänge benutzen. Beim Tanzen kann es auch ganz lustig sein einige Effekte aus der Blockpalette „**Aussehen**“ einzusetzen :



Viel Spaß beim Programmieren und beim Raten!

I.2.2 Für Fortgeschrittene: Zusammengesetzte Bedingungen

Oft benötigt man mehrere Bedingungen, um etwas zu überprüfen. Diese Bedingungen können dann zusammengesetzt (miteinander verknüpft) werden

In der Mathematik z.B. muss man manchmal überprüfen, ob ein Wert in einem bestimmten Bereich liegt: $0 < x < 1000$

Eine solche Bedingung kann man in Scratch mit folgenden Blöcken erstellen:



<**und**> bedeutet, dass beide Bedingungen gleichzeitig erfüllt sein müssen

<**oder**> bedeutet, dass mindestens eine der beiden Bedingungen erfüllt sein muss

<**nicht**> bedeutet, dass die Bedingung nicht erfüllt sein darf

Um zu überprüfen, ob $0 < x < 1000$ muss man in Scratch schreiben:



Um zu überprüfen, ob $x \leq 0$ oder $x \geq 1000$ ist, kann man in Scratch schreiben:



Dasselbe kann man aber auch so schreiben:





J. Prozeduren

Aus bereits bestehenden Befehlen kann man neue Befehlsböcke erzeugen, diese nennt man **Prozeduren**. Prozeduren kann man, genau wie Befehle, in Programmen aufrufen.

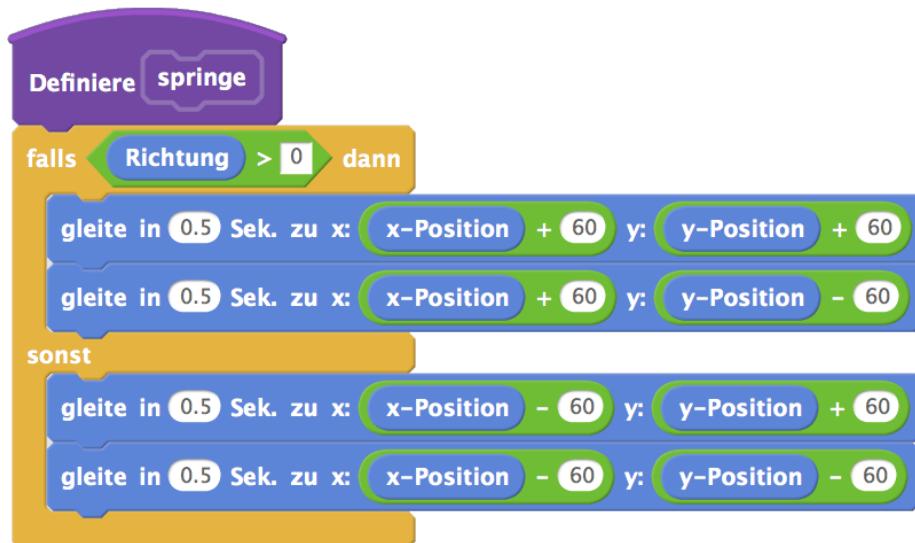
J.1. Prozeduren definieren

In der Blockpalette können neue Prozeduren durch einen Klick auf den Knopf erzeugt werden. In dem Dialog kannst du anschließend den Namen der Prozedur eingeben.



Nach Bestätigung durch den Knopf **OK** erscheint der Hut **Definiere** im Programmreich. Unter ihm kannst du jetzt die Befehle andocken, welche beim Aufruf der Prozedur ausgeführt werden sollen.

In folgendem Beispiel wird die Prozedur **springe** definiert welche dem Sprite erlaubt 60 Pixel hoch und 120 Pixel weit zu springen.





J.2. Prozeduren aufrufen

Nachdem du eine neue Prozedur definiert hast, erscheint diese als Block in der Blockpalette und du kannst sie wie einen Befehlsblock in ein Programm einfügen.



Merke:

In einem Sprite kannst du nur die Prozeduren benutzen welche du in diesem Sprite definiert hast und nicht die Prozeduren anderer Sprites.

Aufgabe J-01 Die Katze springt

Ziel: Die Katze springt über Hindernisse.



Beschreibung:

- Definiere die Prozedur **springe** wie oben angegeben.
- Falls die Leertaste gedrückt wird, springt die Katze. Benutze die Prozedur **springe!**
- Durch Klicken auf die grüne Fahne geht die Katze auf dem Strand hin und her. Wenn sie einen Stein berührt, springt sie über ihn. Benutze die Prozedur **springe!**
- Speichere als **Aufgabe J-01** ab.
- Stell dir vor du hättest die Prozedur **springe** nicht benutzen können. Was kannst du daraus schließen? Welche Vorteile hat das Benutzen von Prozeduren?



J.3. Argumente und Parameter

Viele Befehle, welche du bereits kennst, haben **Argumente**.

Beispiele:



Der **gehe** Befehl hat ein Argument. Dieses gibt die Größe eines Schrittes an (im Beispiel: 10).

Das Eingabefeld ist rund weil es eine Zahl enthalten muss.



Der **sage** Befehl hat ein Argument. Dieses gibt den Text an welcher das Sprite sagt (im Beispiel: „Hallo!“).

Das Eingabefeld ist rechteckig weil es einen Text enthalten muss.



Der **warte bis** Befehl hat ein Argument. Dieses gibt an ob das Programm warten muss oder fortgesetzt werden kann.

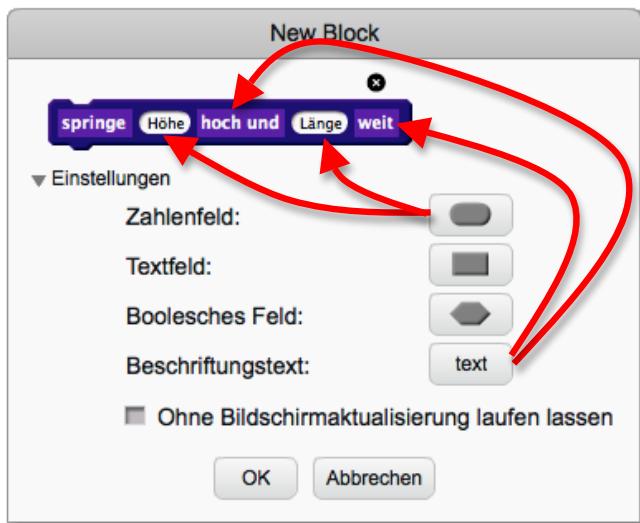
Das Eingabefeld hat spitze Enden weil es eine Bedingung enthalten muss deren Wert **wahr** oder **falsch** sein kann.



Dieser **sage** Befehl hat zwei Argumente. Den ausgesprochenen Text (rechteckiges Feld) und die Dauer während der der Text angezeigt wird (rundes Feld).

Auch Prozeduren können Argumente haben. Bei der Definition einer Prozedur können wir die Platzhalter für die Argumente festlegen, diese Platzhalter nennt man in der Programmierung **Parameter**.

Dazu müssen wir im Dialog den Teil Einstellungen aufklappen.

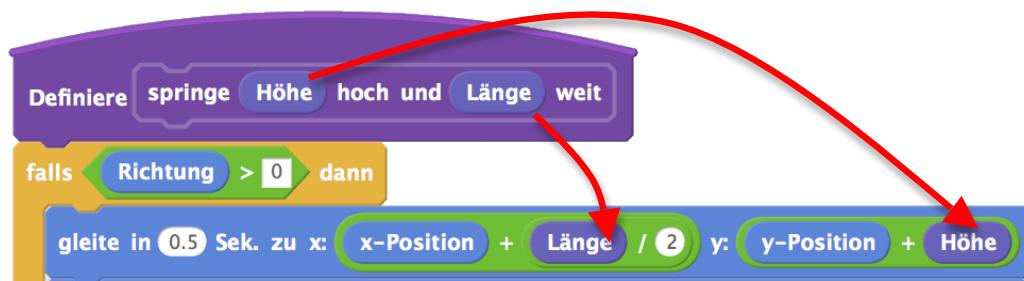


SCRATCH

In dem vorherigen Beispiel haben wir der Prozedur **springe** zwei Parameter hinzugefügt. Einen Parameter zur Eingabe der Höhe und einen zur Eingabe der Länge. Damit können wir beim Aufruf der Prozedur angeben wie hoch und wie weit das Sprite springen soll.

Weil beide Parameter Zahlen enthalten müssen, haben wir auf die runden Zahlenfelder geklickt. Anschließend haben wir den Parametern einen Namen gegeben. Die beiden Beschriftungstexte erlauben es, die Prozedur verständlicher zu gestalten.

Nach Bestätigung durch den Knopf **OK** erscheint der Hut **Definiere** mit den definierten Parametern und Beschriftungen.



Die Parameter können nun mit der Maus dort wo sie im angedockten Programmcode gebraucht werden, geschoben werden.

Merke:

Du kannst die Definition einer Prozedur auch nachträglich ändern: mit der rechten Maustaste auf den Hut **Definiere** klicken und im Kontextmenü **Bearbeiten** auswählen.

Aufgabe J-02 Die Katze springt auch höher

Ziel: Die Katze springt auch über höhere Hindernisse.



Beschreibung:

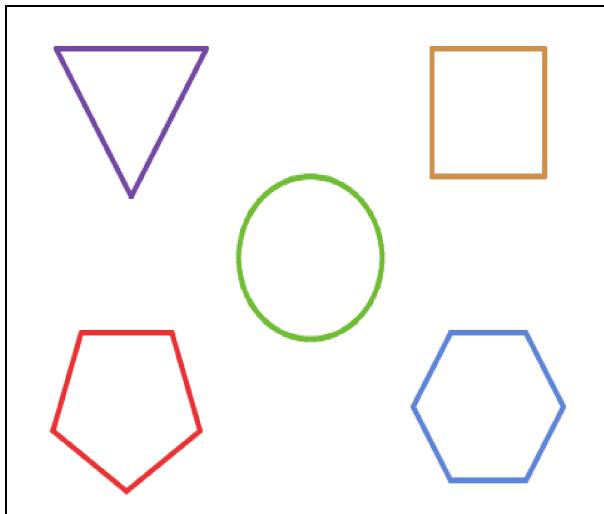
- Öffne das Projekt **Aufgabe J-01** und speichere es unter dem Namen **Aufgabe J-02** ab.
- Definiere die Prozedur **springe** wie oben angegeben mit den Parametern **Höhe** und **Länge** und vervollständige den Programmcode der Prozedur.

SCRATCH

- c) Setze eine Palme auf den Strand und ändere das Skript der Katze sodass sie sowohl über den Stein (kleiner Sprung) wie über die Palme (großer Sprung) springen kann!
- d) Speichere das Projekt ab.

Aufgabe J-03

Ziel: Regelmäßige Polygone zeichnen.



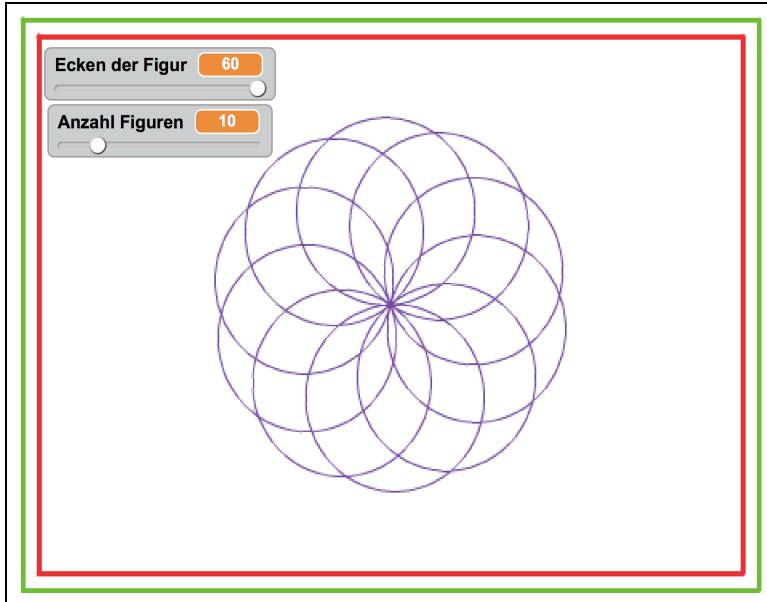
Beschreibung:

- a) Erstelle eine Prozedur **polygon** welche ein regelmäßiges Polygon zeichnet. Die Anzahl der Ecken sowie die Länge einer Seite werden als Argumente angegeben.
- b) Erstelle ein Programm welches, wie oben angegeben, verschiedene Polygone in verschiedenen Farben auf die Bühne zeichnet.
- c) Speichere als **Aufgabe J-03** ab.

SCRATCH

Aufgabe J-04

Ziel: Mandala Generator mit Prozeduren.



Beschreibung:

- Öffne das Projekt **Aufgabe J-03** und speichere es unter dem Namen **Aufgabe J-04** ab.
- Erstelle die Prozedur **mandala** welche ein Mandala bestehend aus einer bestimmten Anzahl von Polygonen zeichnet. Benutze dafür die Prozedur **polygon**. Überlege welche Parameter du für die Prozedur **mandala** benötigst.
- Benutze die Prozedur **mandala** um einen Mandala Generator zu programmieren (siehe **Aufgabe I-04** aus dem Kapitel I).
- Erstelle die Prozedur **rechteck** welche ein Rechteck zeichnet. Die Länge und die Breite des Rechtecks werden als Argumente angegeben.
- Benutze die Prozedur **rechteck** um einen doppelten Rahmen um ein bestehendes Mandala zu zeichnen (durch Drücken der Taste „r“).
- Ändere die Definition der Prozedur indem du die Option „Ohne Bildschirmaktualisierung laufen lassen“ auswählst. Starte den Mandala Generator noch einmal. Wasstellst du fest?

Ohne Bildschirmaktualisierung laufen lassen

- Speichere das Projekt ab.

ScRATCH

IMPRESSUM

Copyright:

Ministère de l'Education nationale, de l'Enfance et de la Jeunesse
29, rue Aldringen
L-2926 Luxembourg

Année édition: 2015

N° commande: ED/EST/340

ISBN: 978-99959-1-038-9