

TurtleStitch Editor



Gehe zu www.turtlestitch.org und klicke **Run TurtleStitch!**



Run TurtleStitch!

The screenshot shows the TurtleStitch editor interface. On the left is a block palette with categories like 'Bewegung' (Movement) and 'Steuerung' (Control). The center is a script area with a sequence of blocks: 'Wenn angeklickt' (When clicked), 'zurücksetzen' (Reset), 'Kreuzstich 10 mal 10 zentriert' (Cross-stitch 10x10 centered), a 'wiederhole' (Repeat) block containing 'wiederhole 6 mal' (Repeat 6 times) with 'gehe 100 Schritte' (Go 100 steps) and 'drehe 60 Grad' (Turn 60 degrees), and 'gehe 2 Schritte' (Go 2 steps) with 'drehe 30 Grad' (Turn 30 degrees). The right side is a stage showing a complex geometric cross-stitch pattern. Below the stage are statistics: 'Stiche : 2364', 'Sprungstiche : 0', and 'Größe : 8.02 x 8.02 cm'. At the bottom are export options like 'Exportieren als SVG' and 'Exportieren als Melco/EXP'.

1. Dateimenu und
Einstellungen
Sprachen, speichern,...

2. Block-Kategorien

3. Block-Palette

Liste mit allen Blöcken in der
ausgewählten Kategorie

4. Skriptbereich

baue größere Programme aus Blöcken

5. Bühne

zeigt das aktuelle Design an

6. Informationen

Informationen über das Design, Fehler-
meldungen, Export-Optionen

TurtleStitch Blöcke

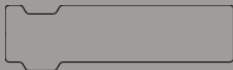


 Bewegung	 Steuerung
 Fühlen	 Operatoren
 Stift	 Variablen
 Stickerei	 Farben
 Andere	

Blöcke sind in verschiedene **Kategorien** einsortiert. Jede Kategorie hat eine eigene Blockfarbe.

Blöcke haben **3 verschiedenen Formen**, je nach Funktion

Befehle:
tun etwas



Funktionen:
berichten Werte



Prädikate:
berichten wahr/falsch



Blöcke können einen oder mehrere **Parameter (Eingaben)** haben. Diese Eingaben verändern, was ein Block macht.

Einige Eingabefelder benötigen einen bestimmten Eingabetyp. Du kannst sie an ihrer Form unterscheiden:



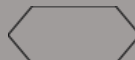
alle Eingabetypen



nur Listen



nur Zahlen



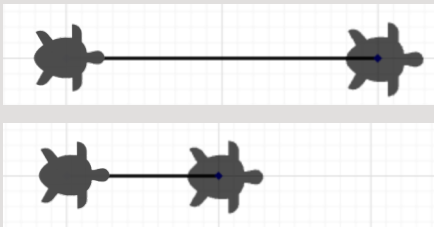
nur wahr/falsch

Die Turtle Bewegungen - Turtle Geometrie



Du kannst die Position der Turtle mit den Blöcken aus der blauen **Bewegung Kategorie** verändern.

Der **gehe** Block bewegt die Turtle **in die Richtung, in die sie gerade schaut**. Die **Strecke**, die sie bewegt wird, kann mit der **Eingabe** angepasst werden.



gehe Schritte

gehe Schritte

Die schwarze Linie repräsentiert den Faden der Stickerei, die blauen Karos die tatsächlichen Stiche.

Versuche, den gehe- und drehe-Block zu kombinieren, um interessante repetitive Formen zu erzeugen.

Du kannst die aktuelle Position der Turtle immer herausfinden, indem du einen **Haken im Kästchen vor den x- und y-Position-Blöcken in der Palette setzt**.

Dadurch werden auf der Bühne die aktuellen Koordinaten angezeigt.

x-Position

y-Position

x-Position

y-Position

Objekt x-Position	5
Objekt y-Position	10

Die Turtle Bewegungskordinaten



gehe zu zufällige Position ▾

gehe zu x: 0 y: 0

Du kannst die Turtle an **absolute Koordinaten** bewegen mit den **gehe zu**-Blöcken.

Du kannst die **x- und y-Koordinaten** unabhängig **setzen** und **ändern** mit diesen Blöcken.

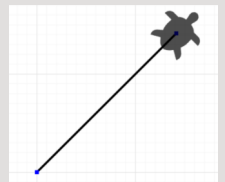
Die **setze**-Blöcke transportieren die Turtle an eine **absolute** Koordinate.

Die **ändere**-Blöcke verändern die Position der Turtle basierend auf den aktuellen Koordinaten.



zeige Richtung 45 ▾

ändere x um 100



zeige Richtung 45 ▾

gehe 100 Schritte

setze x auf 0

setze y auf 0

ändere x um 10

ändere y um 10

Diese Blöcke sind sehr nützlich, wenn du die Turtle an eine **bestimmte Position** bewegen möchtest. Du solltest diese Art der Bewegung der Turtle aber **vermeiden**, wenn du **wiederholende Formen in verschiedene Richtungen** erzeugen möchtest.

Es ist sehr schwer, die passenden Koordinaten für jede Richtung auszurechnen. Versuche in diesen Projekten, die drehe- und gehe-Blöcke mit relativer Orientierung zu verwenden.

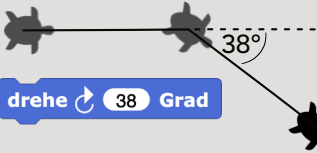
Richtungen Ändern



zeige Richtung 90 ▾

drehe ↻ 15 Grad

drehe ↻ 15 Grad



Du kannst die Richtung der Turtle mit dem **zeige Richtung**-Block auf einen **absoluten Wert** setzen.

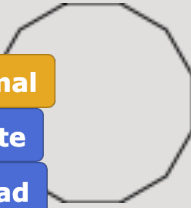
Du kannst die Richtung der Turtle auch **relativ zu ihrer aktuellen Richtung** mit dem **drehe**-Block verändern.

(90) rechts
(-90) links
(0) oben
(180) unten
zufällig

wiederhole 12 mal

gehe 20 Schritte

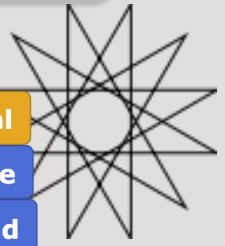
drehe ↻ 30 Grad



wiederhole 12 mal

gehe 100 Schritte

drehe ↻ 150 Grad



Versuche, den drehe- und gehe-Block mit einer Schleife zu verwenden. Teste verschiedene Winkel und Wiederholungen.

Du kannst die aktuelle Richtung der Turtle immer anzeigen lassen, indem du einen **Haken im Kästchen vor dem Richtung-Block** in der Palette setzt.

Dadurch wird **auf der Bühne** die aktuelle Richtung angezeigt.

Richtung

90



Richtung

Objekt Richtung 90

Kreisbögen Zeichnen

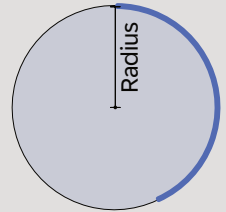


arc ↻ radius: 50 degrees: 155

Mit den **arc**-Blöcken kannst du Teile eines Kreises (Kreisbogen) zeichnen.

Der **radius** regelt die Größe des Kreises, die **degrees** welcher Anteil gezeichnet wird.

Du kannst zwischen Kreisbögen entscheiden, die sich mit oder gegen den Uhrzeigersinn drehen.



arc ↻ radius: 50 degrees: 60

arc ↻ radius: 50 degrees: 240

arc ↻ radius: 20 degrees: 240

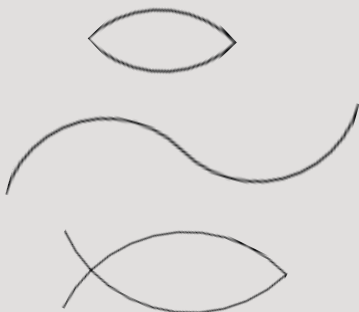
arc ↻ radius: 50 degrees: 240



Anderer Winkel.

Anderer Radius.

Andere Ausrichtung.



Du kannst die **arc**-Blöcke anstatt des `gehe _ Schritte` Blocks in Skripten, die du schon programmiert hast, probieren.

Versuche, die **arc**-Blöcke als Blütenblätter für eine Blume, eine Wellenlinie oder als Grundform für andere repetitive Formen zu verwenden.

Entscheidungen Treffen - Bedingte Anweisungen

Bedingte Anweisungen sind Blöcke, die die Befehle in der c-förmigen Eingabe basierend auf einer Bedingung ausführen.

Maustaste gedrückt?

3 × var1 > var2 / 4

Eine Bedingung kann ein einfacher sechseckiger Block (Prädikat) oder ein komplexerer Ausdruck aus mehreren verschachtelten Blöcken sein.

Alles ist möglich, solange das **Ergebnis wahr oder falsch** ist.

falls

Der **falls**-Block führt die Blöcke in der c-förmigen Eingabe nur aus, wenn die Bedingung wahr ist. Sonst wird dieser Teil des Programms weggelassen und danach weitergemacht.

Der **falls sonst** Block führt eine der beiden Möglichkeiten aus.

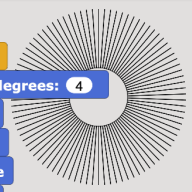
Ist die Bedingung wahr, werden die im falls-Teil eingeschlossenen Blöcke ausgeführt. Ist sie falsch, wird der sonst-Teil ausgeführt.

falls

sonst

wiederhole 90 mal

```
arc ↻ radius: 50 degrees: 4
drehe ↻ 90 Grad
gehe 100 Schritte
gehe -100 Schritte
drehe ↻ 90 Grad
```

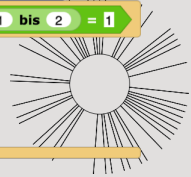


Alle 4 Grad wird
Linie gezeichnet

Nur wenn Zufalls-
zahl 1 ist wird
Linie gezeichnet.

wiederhole 90 mal

```
arc ↻ radius: 50 degrees: 4
falls Zufallszahl von 1 bis 2 = 1
  drehe ↻ 90 Grad
  gehe 100 Schritte
  gehe -100 Schritte
  drehe ↻ 90 Grad
```



Bedingungen Verschachteln - Logische Operatoren



Logische Operatoren können verwendet werden, um mehr als eine **Bedingung** zu **verknüpfen**.

Du findest drei davon in der Operatoren-Kategorie.

und

oder

nicht

nicht falsch wahr

nicht wahr falsch

Der **nicht**-Operator **dreht** den Wert in seinem Eingabefeld **um**.

Er wandelt falsch in wahr um und umgekehrt.

Der **und**-Operator nimmt zwei Prädikate als Eingabe und berichtet nur **wahr, wenn alle** davon **wahr sind**.

wahr und wahr wahr

wahr und falsch falsch

falsch und falsch falsch

wahr oder wahr wahr

wahr oder falsch wahr

falsch oder falsch falsch

Der **oder**-Operator nimmt zwei Prädikate als Eingabe und berichtet **wahr, wenn mindestens eines** davon **wahr ist**.

Schleifen



Schleifen können verwendet werden, um Teile des Programms zu **wiederholen**.

Jeder Block innerhalb des c-förmigen Eingabefelds wird so oft wiederholt, wie auf der Schleife angegeben ist.

fortlaufend



Die **fortlaufend-Schleife** wiederholt die eingeschlossenen Blöcke **endlos** (oder bis auf den roten Stopp-Knopf oder das aktive Skript geklickt wird).

The **wiederhole x mal** Block **wiederholt x mal**.

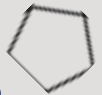
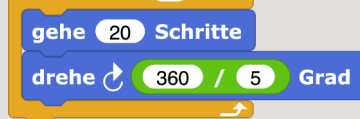
Er ist besonders nützlich für alles, was mit dem **Total Turtle Trip** zu tun hat:

Um eine geschlossene Form zu erzeugen, sollte das Produkt von Drehwinkel und Anzahl an Wiederholungen 360° sein.

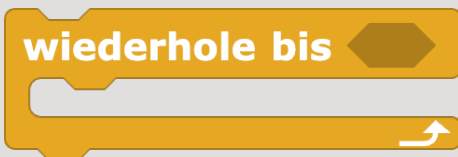
wiederhole 10 mal



wiederhole 5 mal



wiederhole bis



Der **wiederhole bis** Block nimmt eine Stopp-Bedingung als erste Eingabe und wiederholt die eingeschlossenen Blöcke bis diese Bedingung wahr wird.

Die für-Schleife



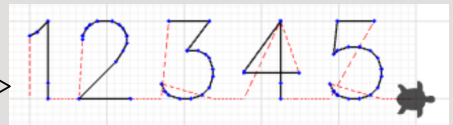
Die **für-Schleife** enthält eine **interne Variable i**, die auf den Startwert gesetzt wird, wenn die Schleife ausgeführt wird.

i wird bei jeder Wiederholung **automatisch um 1 erhöht**, bis der Endwert erreicht wird.

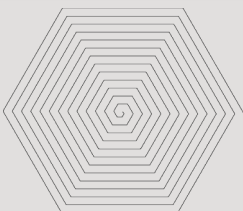
Startwert Endwert

```
für i = 1 bis 10
```

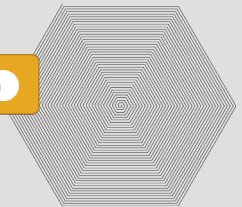
```
für i = 1 bis 5  
  schreibe Text i in Grösse 50  
  gehe 20 Schritte
```



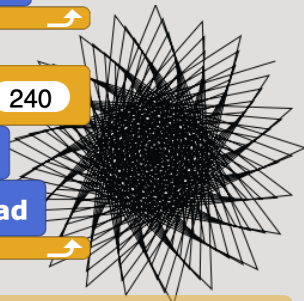
Die **für-Schleife** kann für Spiralen und andere interessante Formen verwendet werden:



```
für i = 1 bis 240  
  gehe i Schritte  
  drehe 60 Grad
```



```
für i = 1 bis 240  
  gehe i Schritte  
  drehe 137 Grad
```



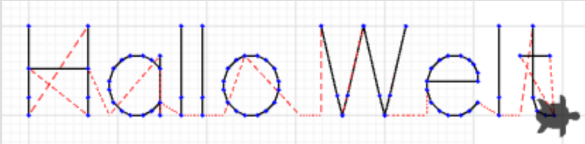
```
ändere i um 2
```

Mit dem **ändere um**-Block kann i um mehr als 1 bei jeder Wiederholung verändert werden, um weniger dichte Muster zu erzeugen.

Texte Sticken



schreibe Text `HalloWelt` in Grösse 21



Der `schreibe Text _ in Grösse _` Block lässt dich Texte sticken.
Beim Erstellen der Buchstaben werden automatisch Sprungstiche an den benötigten Stellen eingefügt.

Länge von Text `HalloWelt` mit Grösse 21 124

Mit dem `Länge von Text _` Block kannst du herausfinden, wie lang dein Text in Pixeln auf der Bühne sein wird. 50 Pixel auf der TurtleStitch-Bühne entsprechen 1 cm. Die Buchstaben haben keine einheitliche Länge.

Buchstabenlänge		
	A	B
26	A	16
1	B	14
2	C	15
3	D	14
4	E	13
5	F	13
6	G	15
7	H	14
8	I	0
9	J	10
10	K	14
11	L	12
12	M	16
13	N	14
14	O	16
15	P	14
16	Q	16
17	R	14
18	S	14
19	T	14
20	U	14
21	V	16
22	W	20
23	X	14
24	Y	16
25	Z	14

Zickzack mit Dichte 3 Breite 8 zentriert ✓

Du kannst die Schrift auch mit verschiedenen Stichtypen benutzen.



Texte Sticken



Wenn du für jeden Buchstaben noch weitere Befehle ausführen möchtest, kannst du den Text, den du schreiben möchtest, nach Buchstaben **trennen** und dann jeden Buchstaben mit dem **für jedes Element** Block verarbeiten.

1	H
2	a
3	l
4	l
5	o
6	
7	W
8	e
9	l
10	t

Länge: 10 ▼

trenne HalloWelt nach Buchstabe ▼

für jedes Element von

Du kannst z.B. die Buchstaben in Schlangenlinien oder im Kreis schreiben, indem du nach jedem Buchstaben drehst und gehst.

schreibe Text Element in Grösse 25

drehe ↻ Zufallszahl von -20 bis 20 Grad

Sprungstich ✓

gehe 10 Schritte

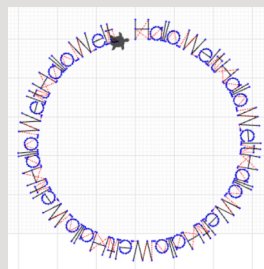
Sprungstich ✗



schreibe Text Element in Grösse 25

drehe ↻ 5 Grad

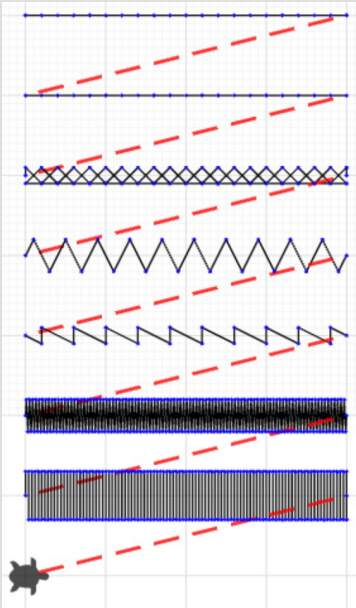
gehe 2 Schritte



Stichtypen



Stickerei



Laufstich in 10 er Schritten

Dreifach-Laufstich in 10 er Schritten

Kreuzstich 10 mal 10 zentriert ✓

Zickzack mit Dichte 20 Breite 20 zentriert ✓

Z-Stich mit Dichte 20 Breite 10 zentriert ✓

Satinstich mit Breite 20 zentriert ✓

Tatami-Stich mit Breite 100 Intervall 40 zentriert ✓

Der Stichtyp beeinflusst die Anzahl an Stichen in einem Design.

Kreuzstich Dichte 5
(ein Stich alle 5 Pixel, das Stichmuster ist 10 Pixel hoch)

Kreuzstich Dichte 10
(ein Stich alle 5 Pixel, das Stichmuster ist 20 Pixel hoch)

Designs können in verschiedenen **Stichtypen** erstellt werden. Die Blöcke dafür befinden sich in der **Stickerei**-Kategorie.

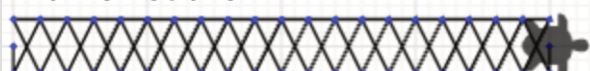
Du kannst zwischen einfachen Laufstichen und komplexeren Mustern, bei denen die Dimensionen einzeln angepasst werden können, entscheiden.

2 cm 62 Stiche



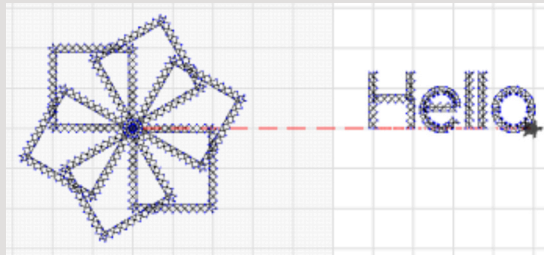
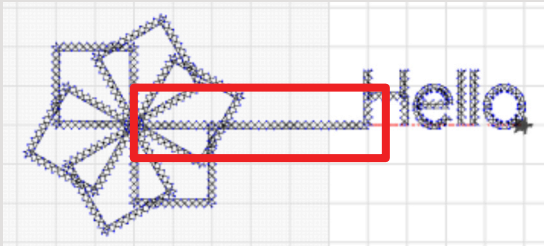
Kreuzstich 5 mal 10 zentriert ✓

2 cm 32 Stiche



Kreuzstich 10 mal 20 zentriert ✓

Stichtypen - Sprungstich



Wenn du mit der Stickmaschine arbeitest, sind alle Teile eines Designs immer durch den Faden verbunden.

Das Aktivieren eines **Sprungstichs** ermöglicht dir, diesen Teil des Fadens herauszutrennen, wenn die Stickerei fertig ist.

Sprungstich

gehe 300 Schritte

Sprungstich

Sprungstiche werden auf der Bühne als rote gestrichelte Linien angezeigt:

Sprungstich

gehe 100 Schritte



Wird der Sprungstich deaktiviert, wird der Stichtyp auf den vorher aktiven zurückgesetzt:

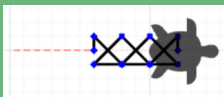
Kreuzstich 10 mal 10 zentriert

Sprungstich

gehe 30 Schritte

Sprungstich

gehe 30 Schritte



Mit Daten Arbeiten - Listen



Du kannst **Listen** benutzen, um mehrere Werte in einer Datenstruktur zu speichern. Diese Werte können alles von Zahlen über Text über eine andere Liste sein (eine Tabelle ist eine z.B. eine Liste aus Listen).

1	1
2	3
3	5

Länge: 3 ▼

Liste 1 3 5 ◀▶

Du kannst mit dem **Element _ von _** Block direkt auf einzelne Elemente der Liste im zweiten Eingabefeld zugreifen.

Element 1 ▼ von ☰

Du kannst entweder eine der Möglichkeiten **aus dem Dropdown-Menü auswählen** oder direkt einen **Index in die erste Eingabe schreiben**.

Länge ▼ von ☰

- Länge
- Rang
- Dimensionen
- Auflistung
- Spalten
- Umkehrung
- Textzeilen
- csv
- json

Mit dem **von Block für Listen** kannst du Infos über eine Liste herausfinden (die Länge-Eingabe berichtet z.B. die Anzahl der Elemente der Liste).

Außerdem kannst du Operationen auf die Liste anwenden, z.B. mit der Umkehrung-Eingabe.

Länge ▼ von Liste 1 3 5 ◀▶

3

1	5
2	3
3	1

Länge: 3 ▼

Umkehrung ▼ von Liste 1 3 5 ◀▶

Mit Daten Arbeiten - Listen



setze Form ▾ auf Liste ▶

Um eine Liste in einer Variablen zu speichern, damit du während des Programms Elemente hinzufügen kannst, musst du **die Variable als** eine (leere) **Liste initialisieren.**

In deinem Programm kannst du die Liste mit den **füge hinzu** oder **entferne** Blöcken erweitern oder verkürzen.

Form	
1	90
2	0

Länge: 2 ▾

füge etwas zu Liste hinzu

entferne 1 aus Liste

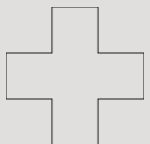
füge Richtung zu Form hinzu

für jedes Element von Liste

mache etwas mit Element

Der **für jedes Element** Block ist eine **Schleife, die nur mit Listen funktioniert.** Sie führt die Blöcke in der c-förmigen Eingabe für alle Elemente der Liste einmal aus.

Dieses Beispiel erzeugt ein Design aus einer Liste von Richtungen. Diese Liste könnte auch automatisch mit einem Programm, das die Turtle mit den Pfeiltasten kontrolliert, erstellt werden.



für jedes Element von Liste 90 0 90 180 90 180 270 180 270 0 270 0

zeige Richtung Element

gehe 100 Schritte

Einfache Mathematische Operatoren



Mathematische Operatoren befinden sich in der **Operatoren Kategorie**:

Sie können **verschachtelt** Operatoren werden, um kompliziertere Rechnungen durchzuführen.

Sie können als **Eingabe für andere Blöcke** verwendet werden.



20 + 8 - 4 / 6 × 2

8

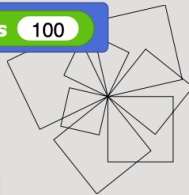
Verschachtelte Operatoren werden **von innen nach außen** evaluiert. Jeder Block steht für eine Klammer. In diesem Fall

$$(((20 + 8) - 4) / 6) \times 2 = 8$$

wiederhole 10 mal

Quadrat Zufallszahl von 50 bis 100

drehe 360 / 7 Grad



Er kann verwendet werden, um **Variation** in generative Kunst-Projekte zu bringen.

Der **Zufallszahl**

Operator berichtet eine zufällige Zahl zwischen den Zahlen im ersten und zweiten Eingabefeld.

Zufallszahl von 1 bis 10

9

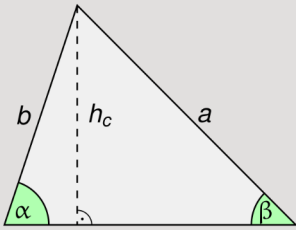
Mehr Mathematische Operatoren



Wurzel ▼ von 10

Im Dropdown Menü des **von**-Operators kannst du eine Operation auswählen. Er berichtet das Ergebnis der Operation und der Zahl im zweiten Eingabefeld.

Damit kannst du z.B. die Winkel und Seitenlänge von Dreiecken mit der Sinusregel berechnen.



$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma}$$

$$\sin \beta = \frac{h_c}{a} \quad \sin \alpha = \frac{h_c}{b}$$

Der **modulo** Operator berichtet den **Rest** der Division der ersten geteilt durch die zweite Zahl.

Damit kannst du z.B. etwas nur jedes zweite, dritte, ... Mal machen.

falls Zahl modulo 2 = 1

mach dies

sonst

mach das

Dieses Skript führt "mach dies" für jede gerade und "mach das" für jede ungerade Zahl aus.

Betrag
neg
sign
Aufgerundet
Abgerundet
Wurzel
sin
cos
tan
asin
acos
atan
ln
log
lg
e[^]
10[^]
2[^]
id

5 modulo 3

2

Variablen erzeugen



Variablen können verwendet werden, um in einem Programm **Werte zu speichern und abzurufen**. Sie können Zahlen, Text, Listen oder sogar Blöcke speichern.

Winkel

50

1 5
2 10

Position

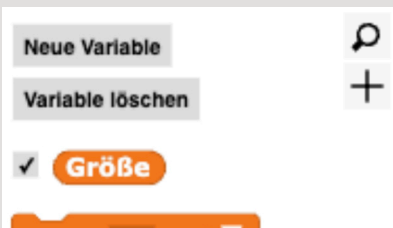
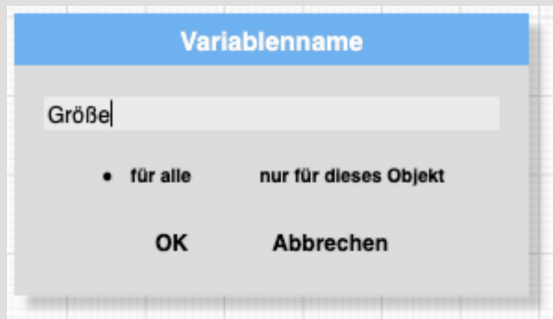
Länge: 2



Du kannst eine Variable erstellen, indem du auf den **Neue Variable** Knopf in der Variablen Kategorie klickst.

Du kannst der Variable einen Namen geben und dann OK drücken, um sie zu erstellen.

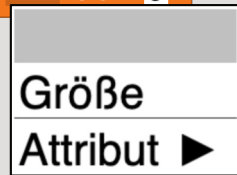
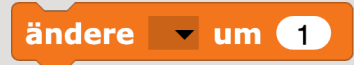
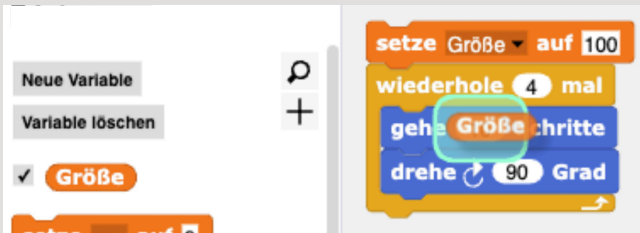
Versuche, einen Namen zu wählen, der gut beschreibt, was die Variable speichern soll.



Deine neue Variable erscheint am **oberen Ende der Variablen Kategorie**.

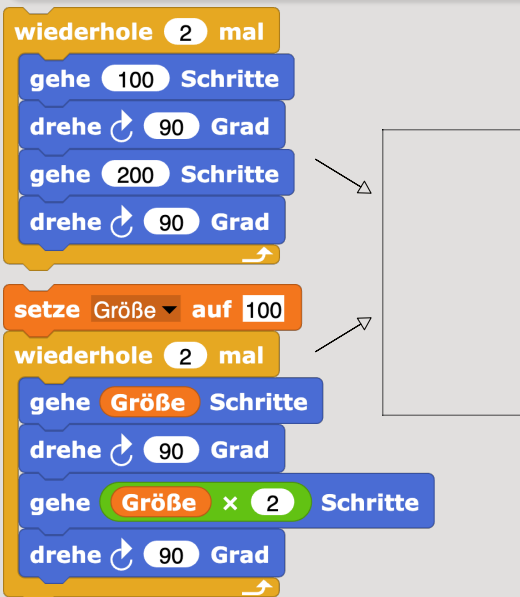
Wenn das Häkchen gesetzt ist, wird ihr Wert auf der Bühne angezeigt.

Variablen benutzen



Du kannst die Variablen in einem Programm verwenden, indem du ihre **orangenen Reporter** aus der Palette in ein Skript ziehst.

Du kannst die Werte einer Variablen **setzen** oder **ändern** mit den gleichnamigen Blöcken aus der Variablen-Kategorie. Die gewünschte Variable kannst du im **Dropdown-Menü** der Blöcke auswählen.



Es ist häufig hilfreich eine Form mit einer Variablen zu beschreiben. Das nennt man **Parametrisierung**.

In diesem Fall erzeugen beide Skripte dasselbe Rechteck.

Wenn du die Größe oder Proportion der Form ändern möchtest, ist das in der unteren Version deutlich leichter, weil alles relativ zur **Variable Größe** programmiert ist.

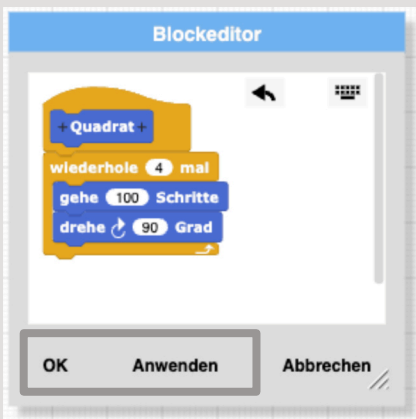
Eigene Blöcke Erstellen



Klicke **+** am oberen Ende jeder Palette, um das **Neuer Block**-Fenster zu öffnen.



1. Wähle eine **Kategorie** für den neuen Block.
2. Gib dem Block einen **Namen**.
3. Wähle den **Blocktyp** aus.
4. Klicke **OK**, um den Blockeditor zu öffnen.



Du kannst den Blockeditor **wie den Skriptbereich programmieren**.

Verbinde, was passieren soll, wenn der Block aufgerufen wird (Definition), mit dem **gelben Startblock (Prototyp)**.

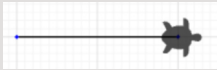
Klicke **Anwenden**, wenn du den Block speichern möchtest, und **OK**, wenn du speichern und den Editor schließen möchtest.

Neue Blöcke erscheinen **am Ende ihrer Kategorie**.

Eigene Blöcke Erstellen



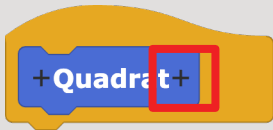
gehe 20 Schritte



gehe 100 Schritte

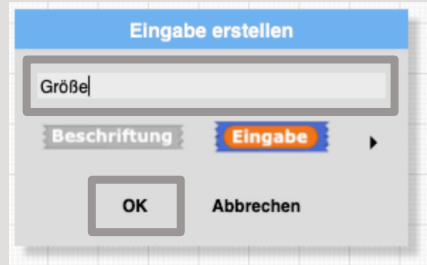
Some blocks have **parameters (inputs)** and behave differently based on their input.

You can add a parameter e.g. for the size to a custom block, too.



Click on the **+** next to **any word** in the block header.

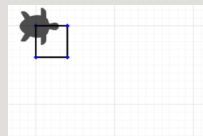
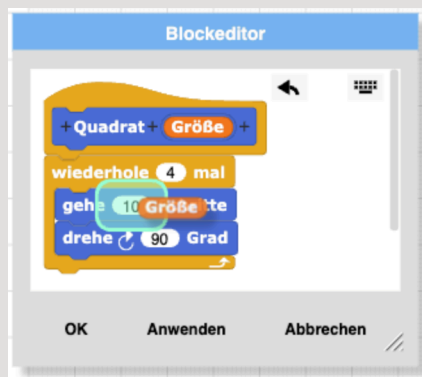
Select a **name** and press **OK**.



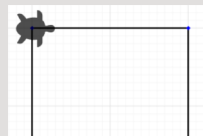
Der Wert der Eingabe wird im **orangenen Oval** gespeichert.

Ziehe das Oval aus dem Blockprototyp an die gewünschte **Position** in der Blockdefinition.

Drücke **OK/Anwenden**.



Quadrat 20



Quadrat 100

Stickdateien Exportieren



Überprüfe alle wichtigen Einstellungen bevor du eine Stickdatei exportierst! Nicht behobene Warnungen können die Stickmaschine oder den Stoff beschädigen.

Stiche : 150 Sprungstiche : 32 Größe : 14.06 x 5.62 cm

60 are too long! (will get clamped) DENSITY WARNING!

+ - Anzeige zurücksetzen Röntgenbild Turbomodus

Passend zoomen

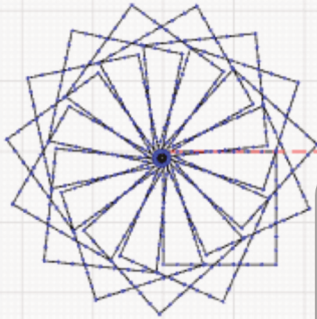
1. Design

3. Größe

2. Warnungen

1. Stelle sicher, dass etwas zum Exportieren **auf der Bühne zu sehen** ist. Eine Datei von einer leeren Bühne kann die Maschine zum Abstürzen bringen.
2. **Entferne alle Warnungen**, bevor du Dateien exportierst. Werf auch einen Blick auf die Karte "Warnungen vermeiden"
3. Überprüfe, dass dein Design die **maximale Größe des Stickrahmens nicht überschreitet**. Zu große Designs werden auf der Maschine nicht angezeigt.

Stickdateien Exportieren



TurtleStitch

1. Versuche, eine sinnvolle **Zahl an Stichen** in deinem Design zu haben. 2000 Stiche benötigen je nach Maschine in etwa 5 min.

Die Zahl an Stichen kann durch den Stichtyp oder die Distanz zwischen den Stichen verändert werden.

1.

Stiche : 701

Sprungstiche : 23

Größe : 9.98 x 4.38 cm

+ - Anzeige zurücksetzen

Röntgenbild

Turbomodus

Passend zoomen

2.

2. Wähle das passende **Dateiformat** für deine Maschine aus.

Exportieren als SVG

Exportieren als Melco/EXP

Exportieren als Tajima/DST

Ignoriere Farben beim Export

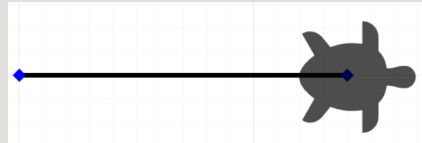
Falls du dir unsicher bist, welches Format du benötigst, wirf einen Blick in das Handbuch der Maschine.

Wenn du auf einen der Export-Knöpfe klickst, wird dein Design in den Downloads-Ordner deines Browsers heruntergeladen. Benutze einen USB Stick um es an die Maschine zu übertragen.

Warnungen Vermeiden



gehe 70 Schritte



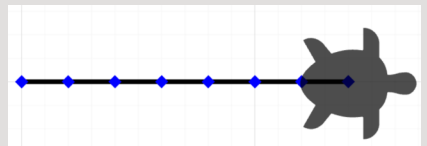
Ohne einen ausgewählten Stichtyp fügt die Turtle nur einen Stich am Anfang und Ende jeder Bewegung ein, in diesem Fall bei 0 und 70 Schritten.

Dieses Skript führt deshalb zu einer Warnung. Der **Abstand zwischen den Stichen ist zu groß**, das kann dazu führen, dass der Stoff sich zusammenzieht:

X are too long! (will get clamped)

Laufstich in 10 er Schritten

gehe 70 Schritte



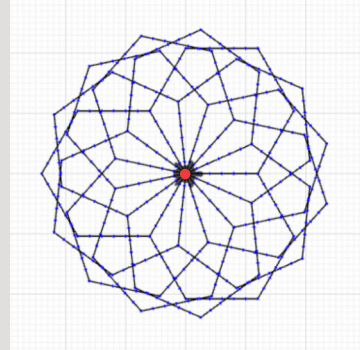
Mit den Blöcken aus der **Stickerei**-Kategorie, die den **Stichtyp** verändern, kann dieses Problem behoben werden.

Für mehr Informationen über die Stichtypen, schaue auf der Karte "Stichtypen" nach.

Warnungen Vermeiden



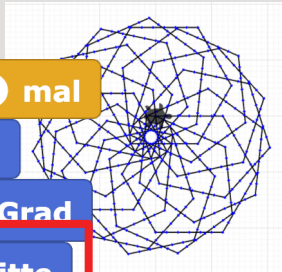
```
wiederhole 15 mal
  Sechseck 60
  drehe ↻ 24 Grad
```



Dieses Skript führt zu einer **DENSITY WARNING**
- die Nadel sticht zu oft in dieselbe Stelle.

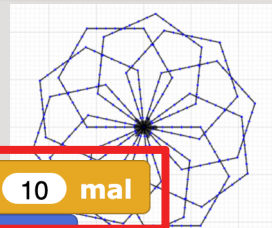
Durch nicht behobene DENSITY WARNINGS kann die Nadel abbrechen.

```
wiederhole 15 mal
  Sechseck 60
  drehe ↻ 24 Grad
  gehe 8 Schritte
```



8 Schritte zu gehen, bevor das nächste Sechseck gemalt wird, führt zu einem kleinen "stichfreien" Kreis in der Mitte des Designs.

```
wiederhole 10 mal
  Sechseck 60
  drehe ↻ 36 Grad
```



Die Anzahl der Sechsecke zu verringern, die an der selben Stelle starten (< 12) behebt die Dichtewarning.