



Vergleich wichtiger Bausteine: Scratch – MIT-App-Inventor 2

Scratch

Variable

Nach dem Klick auf „Neue Variable“ gibt man den Namen der Variable ein und entscheidet, ob sie nur für die aktuelle Figur oder für alle Figuren gültig sein soll.



Erstellen einer neuen Variable liefert:



Initialisierung und Übergabe eines neuen Wertes ist hier nicht klar getrennt. Beides geschieht durch „setze ... auf ...“

Die Methode „ändere ... um ...“ erhöht oder vermindert den Wert der Variable.

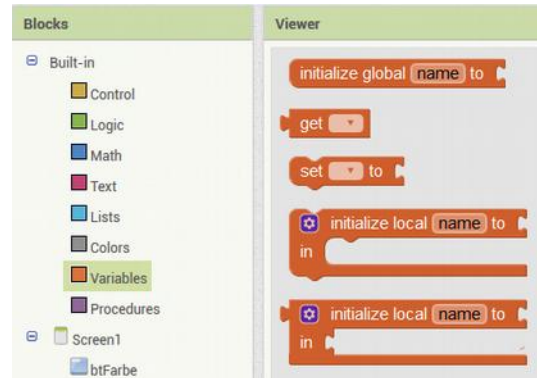
Grundsätzlich sind nur Variablen eines Zahlentyps nutzbar.

„Zeigt“ man die Variable, wird auf der Bühne ein Kästchen mit dem Variablennamen und dem aktuellen Wert gezeigt.



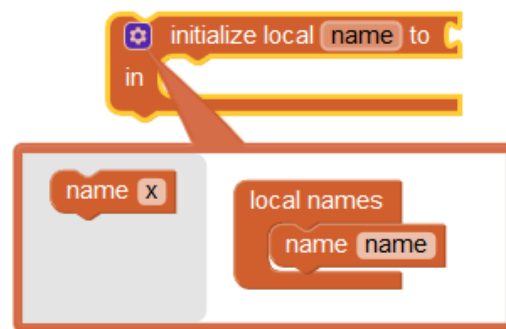
MIT-App-Inventor 2

Hier sind die nötigen Blöcke schon vorhanden.

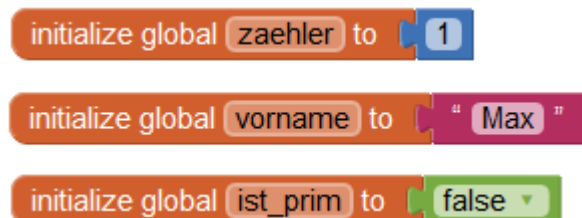


Es wird zwischen globalen und lokalen Variablen entschieden.

Bei globalen Variablen braucht man für jede neue Variable einen eigenen Block, bei lokalen kann man mehrere gleichzeitig nutzen:



Variable können von den verschiedensten Typen sein. Hier drei Initialisierungen:



Für jede Variable gibt es eine get- und eine set-Methode.



Scratch

MIT-App-Inventor 2

Verzweigungen

Es gibt einen einfachen „falls ... dann“ - Block und einen „falls ... dann ... sonst“ - Block.



Hat man mehr als zwei Möglichkeiten, müssen Blöcke ineinander gesetzt werden.



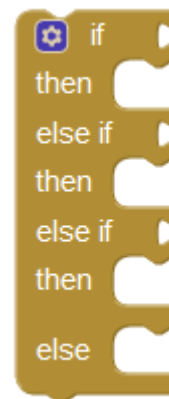
Es gibt einen Block, der beliebig erweitert werden kann:



Zur Erweiterung klickt man auf das Zeichen für Einstellungen und zieht die gewünschten Blockteile nach rechts:



Auch hier könnte man Blöcke ineinander setzen, kann es aber auch so lösen:





Scratch

MIT-App-Inventor 2

Bedingungen

Bei den Operatoren findet man Vergleichs-

operatoren und logische Verknüpfungen.

Beispiele:



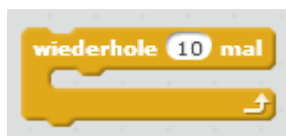
Die logischen Verknüpfungen findet man unter Logic, die Vergleichsoperatoren bei Math.

Beispiele:

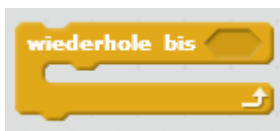


Schleifen

Zählschleife:

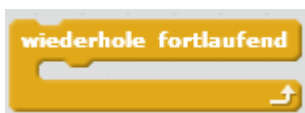


Bedingte Schleife:

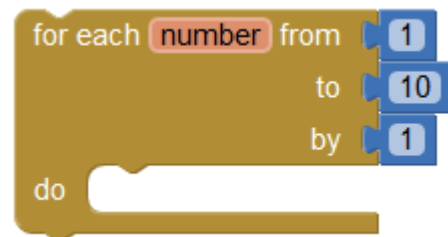


Die Anweisungen im Schleifenkörper werden so lange wiederholt, bis die Bedingung erfüllt ist.

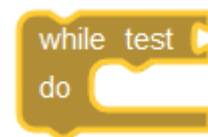
Endlosschleife:



Zählschleife:



Bedingte Schleife:



Solange die Bedingung erfüllt ist, werden die Anweisungen im Schleifenkörper wiederholt.

Hier gibt es keine Endlosschleife. Möchte man etwas z.B. jede Sekunde wiederholen, nutzt man eine Zeitgeber-Komponente (Palette Sensors: Clock).

Ereignis:



Scratch

MIT-App-Inventor 2

Listen

Definition der Liste:



Eingabe über „+“ oder im Code mit



oder



Element löschen:



Element ersetzen:



Auf ein bestimmtes Element der Liste kann mit



auf die Länge der Liste mit

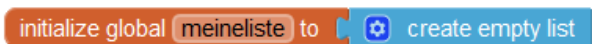


zugegriffen werden.

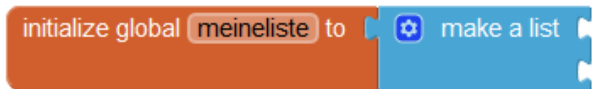


Alles was in diesem Ereignis-Block eingefügt ist, wird z.B. jede Sekunde (Eigenschaft „TimerInterval“: 1000 ms) ausgeführt.

Erzeugen einer leeren Liste:



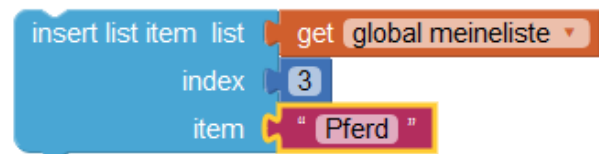
Erzeugen einer Liste mit Elementen:



Anhängen eines neuen Elements:



Einfügen eines Elements an einer bestimmten Position:



Element löschen:



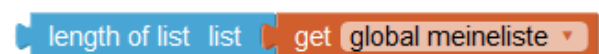
Element ersetzen:



Zugriff auf ein bestimmtes Element:



Zugriff auf die Länge der Liste:



Es gibt noch viele weitere Möglichkeiten, mit Listen zu arbeiten. Siehe Blocks – Lists.



Scratch

MIT-App-Inventor 2

Angezeigt werden Listenelemente in einer ListView (User Interface):

```
set lvMeineListe . Elements to get global meineliste
```

Objekte bewegen

Objekte (Figuren) werden auf die Bühne gesetzt und können mit Blöcken aus der Palette Bewegung in ihrer Position verändert werden.

```
Wenn ich angeklickt werde
  gehe zu x: 100 y: 200
```

Um Objekte bewegen zu können, werden sie auf eine Leinwand (Canvas) gesetzt.

Diese findet man in der Palette Drawing and Animation. Dort gibt es noch zwei weitere Komponenten: einen Ball und ein ImageSprite.

Der Ball ist, was der Name schon sagt. Das ImageSprite ist eine Bildkomponente, die bewegt werden kann.

Alle Möglichkeiten findet man bei den Blocks direkt bei der Canvas bzw. dem ImageSprite.


Beispiel:

```
when imKatze . Touched
  x y
do call imKatze . MoveTo
  x 100
  y 200
```

Mit dem Finger malen:

```
when Canvas1 . Dragged
  startX startY prevX prevY currentX curre
do call Canvas1 . DrawLine
  x1 get prevX
  y1 get prevY
  x2 get currentX
  y2 get currentY
```



Scratch	MIT-App-Inventor 2
	<p>Zwischen einzelnen „Bällen“ oder ImageSprites gibt es auch Kollisionsabfragen.</p> <p>Zwei Beispiele zur Canvasnutzung:</p> <p>imp8_baelle</p> <p>Malen</p>
<p>Beenden</p> <p>Das Programm kann durch Klick auf das rote Stopp-Schild beendet werden.</p> <p>Außerdem beendet folgender Block alle Skripte:</p> 	<p>Um die ganze App zu schließen, nutzt man:</p>  <p>Hat man mehrere Screens und möchte nur einen davon beenden:</p>  <p>Hierbei kann man auch einen Rückgabewert anhängen:</p>  <p>(Blöcke unter Control, nicht beim Screen!)</p>

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>.
Scratch ist lizenziert unter *CC BY-SA 2.0* (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).

Der MIT App Inventor (<http://appinventor.mit.edu>) wurde ursprünglich von einem Entwicklerteam um Mark Friedman und Hal Abelson bei Google entwickelt und 2012 an das MIT übergeben.
Der MIT App Inventor wird unter der Creative Commons Attribution-ShareAlike 3.0 Unported License veröffentlicht: <https://creativecommons.org/licenses/by-sa/3.0>