



Tag 2

In Strukturen Denken und die Anfänge des Programmierens

Pseudocode schreiben in Scratch

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

In den meisten Programmiersprachen gibt es *Datentypen*, und jede Programmiersprache, die ausgeführt werden soll, muss auch so in Maschinencode umgewandelt werden, dass diese Datentypen mindestens nach dem Interpretieren oder Kompilieren erkennbar sind.

In der Folgenden Aufgabe geht es darum, sich mit diesen Datentypen vertraut zu machen und herauszufinden, wie genau solche Datentypen funktionieren und wie Variablen mit Datentypen befüllt werden können.

Dazu nehmen wir uns das Online-Tool *Scratch* zur Hilfe. Hier können wir einfachen Pseudocode ausführen und sehen, wie Variablen gesetzt werden können.

Scratch ist ein Drag-and-Drop Editor, das heißt, hier werden wir hauptsächlich mit der Maus arbeiten.

Repository

Erstellt euch ein Repository mit Bezeichnung 02-Tag2 als eigenes **privates** Repo, um die Fragen zu beantworten und **ladet uns ein**: <https://github.com/yakha-Alexander> & <https://github.com/mannoordhingra>

Materialien

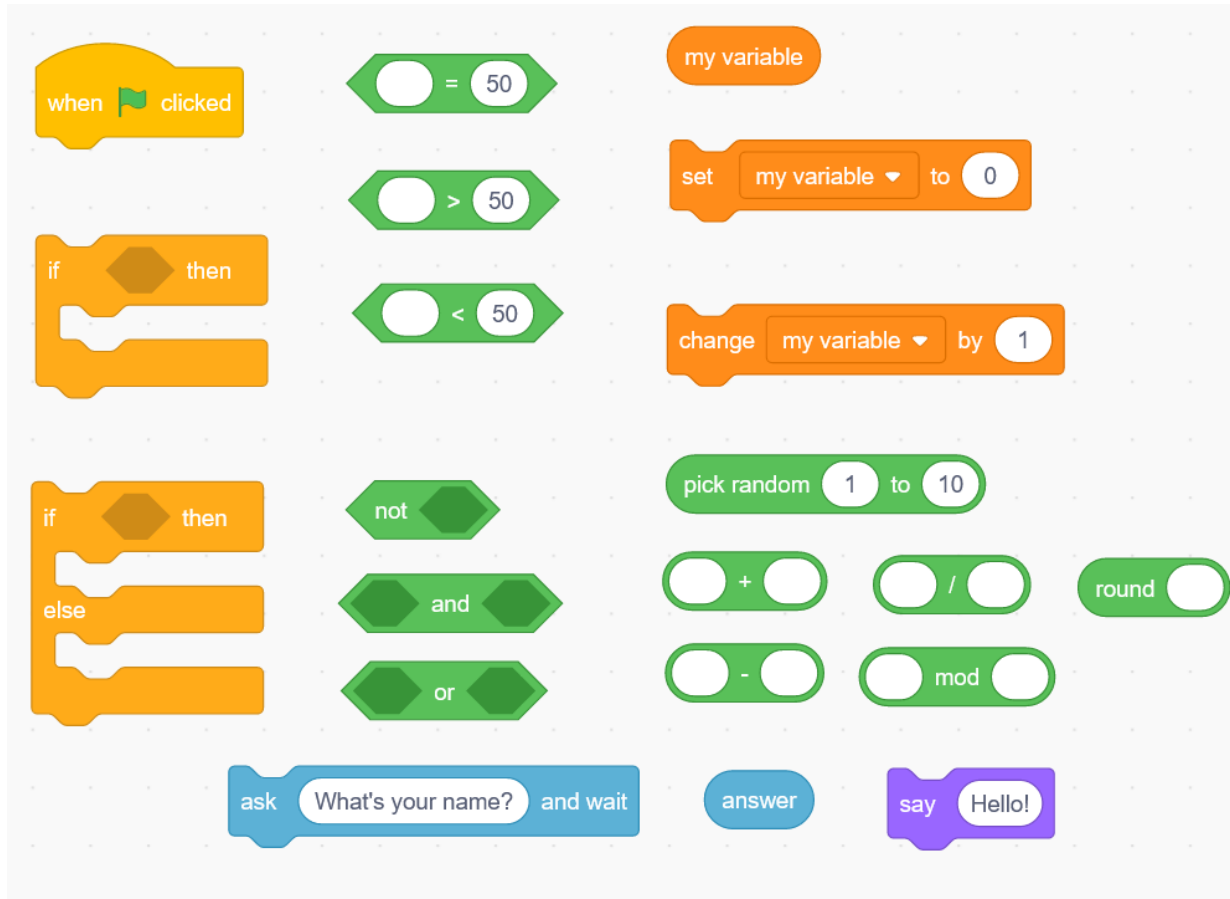
Scratch Online Editor

Es besteht kein Grund sich den Editor herunterzuladen, wir werden ihn lediglich zur Vertiefung des Strukturierten Denkens nutzen, und keine Spiele, oder ähnliches damit erstellen. Wer dies trotzdem einmal in seiner Freizeit ausprobieren möchte, kann dies gerne tun.

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

Der Scratch Online Editor

Abbildung 1



Der Scratch Online Editor

Erklärung Abbildung 1

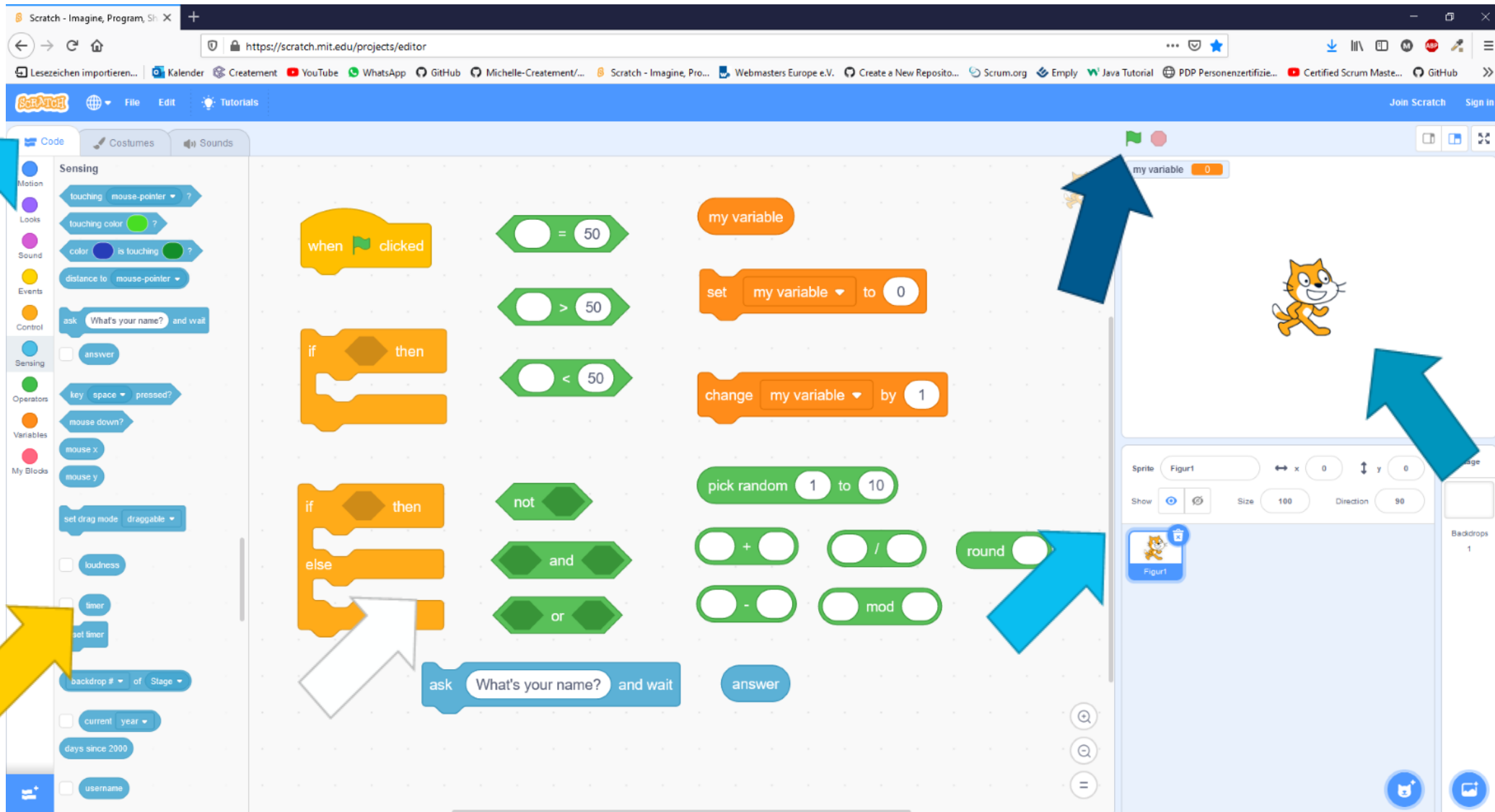
Heute benötigen wir nur diese Befehle zum Ausführen und Ausprobieren von eigenem Code.

Ein wichtiger Befehl ist „when [grüne Fahne] clicked“. Dieser leitet den *Programstart* ein. Was jeder dieser Befehle macht, könnt ihr herausfinden, indem ihr das Scratch Projekt (endet auf .sb3) „Befehlsübersicht“ öffnet und die Befehle einzeln anklickt.

Alle folgenden Aufgaben sollten im Browser im Scratch Editor gemacht werden und im Git-Projekt abgespeichert werden.

Unten wird beschrieben, wie man sich im Scratch Editor zurechtfindet und wie man Programme ausführt.

Aufbau des Scratch Editors (Abbildung 2)



Legende zu Abbildung 2

Die **hellblauen** Pfeile zeigen auf den einzigen Bereich von Scratch, den wir nutzen werden, den Code Bereich der Figur 1.

Der **gelbe** Pfeil zeigt uns verschiedene Befehle an und wir können links auswählen welche Gruppe wir sehen wollen, um schneller zu navigieren.

Der **weiße** Pfeil zeigt auf das Interface, wo wir Blöcke sammeln können.

Der **dunkelblaue** Pfeil zeigt auf die grüne Fahne, die das Programm startet.

Der **türkise** Pfeil zeigt auf die „Canvas“, bzw. die grafische Oberfläche unseres Programms. Diese brauchen wir hier lediglich, um Dinge einzutippen als Eingabe und zu sehen, welche Werte unsere Variablen haben.

Wir werden uns in den nächsten zwei Tagen mit diesem Editor viele Pseudocode Beispiele ansehen, von daher ist es sinnvoll bei Problemen in der Handhabung nachzufragen, wie etwas funktioniert.

Aufgabe 1 (1/2)

Höher/Niedriger 2

Macht diese Aufgabe in einem neuen Scratch Projekt und speichert diese regelmäßig im Git-Repo ab.

Achtung: Scratch lässt unangemeldete Nutzer nur unter dem Dateinamen „Scratch Projekt“ abspeichern, ihr könnt diese Datei aber umbenennen.

1. Schreibe ein Programm, welches den Nutzer darum bittet, sich eine Zahl ausdenken. Der Nutzer soll diese Zahl dann eingeben, diese wird abgespeichert. Prüfe auch, ob die Zahl zwischen 0 und 100 liegt.
2. Lasse nun den Computer raten, dieser soll zufällig raten.
 - a) Tipp: D.h. der Computer generiert eine zufällige Zahl zwischen 0 und 100, ist die Zahl höher, soll der Computer nur noch zwischen 0 und der zuvor generierten Zahl raten, umgekehrt (bei niedriger) von der zuvor generierten Zahl bis 100, natürlich wird in der zweiten Runde die Zahl aus der ersten Runde benötigt, und so wird geraten, bis die Zahl gleich ist.
 - b) Offensichtlich sollte der Computer raten können, bis dieser die Zahl erraten hat.
3. Der Nutzer sollte dem Computer Feedback geben, ob dieser höher oder niedriger raten muss.
4. Zähle auch hier, wie lange es dauert, bis der Computer die Zahl richtig geraten hat.
5. Sollte der Computer laut Nutzer gewonnen haben, prüft dieser, ob es wirklich die eingegebene Zahl war.

Aufgabe 1 (2/2)

Höher/Niedriger 2

Beispiel Durchlauf:

Nutzer: 55

Computer: 43

Nutzer: 1 (Eingabe für höher)

Computer: 86

Nutzer: -1 (Eingabe für niedriger)

Computer: 55

Nutzer: 0

Computer: Stimmt! Ich habe gewonnen.

Aufgabe 2

Fragen zu Algorithmen

1. Ist die Art des Computers zu raten schlau?
2. Wie könnte man diese Suche verbessern?
3. Findest Du einen Begriff, der genau beschreibt, was der Computer besser machen könnte?
Dafür kannst Du im Internet recherchieren.

Viel Erfolg!

Dein yakha Team