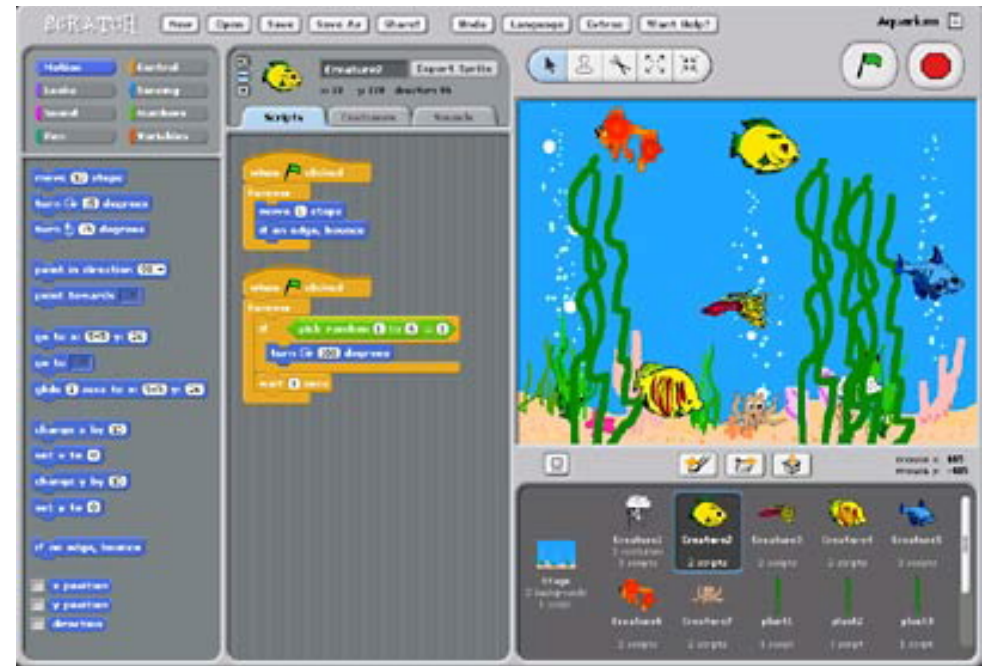


Warum

SCRATCH

?



[Home](#)[Blog](#)[Bibliothek](#)[Vorträge](#)[Beruf](#)

You are here: [Beats Weblog](#) » Gedanken zu Programmierumgebungen für "Medien und Informatik"

Gedanken zu Programmierumgebungen für "Medien und Informatik"

📅 21 May 2016 - 15:16 | 🗨️ Version 4 | 👤 [TobiasH](#)

In den deutschsprachigen Kantonen und Pädagogischen Hochschulen laufen derzeit die Vorbereitungen zur Einführung des **Lehrplans 21** ([Bibliothek:w02172](#)) auf Hochtouren, insbesondere auch für das neue Modul "Medien und Informatik". Eine nicht zentrale, aber trotzdem nicht ganz unwichtige Frage ist die nach Programmierumgebungen.

Als Diskussionsbasis in verschiedenen Kontexten habe ich mich mal hingesetzt und meine diesbezüglichen Überlegungen etwas strukturiert. Es handelt sich somit um eine Konkretisierung der allgemeinen Frage **Welche Programmiersprache eignet sich für die Schule?** ([Bibliothek:f156](#)) auf die aktuelle Situation in der Schweiz und dem bevorstehenden Lehrplan 21.

(direkt kommentiert werden kann dieser Text drüber bei [Google Docs](#))

- ↓ Gedanken zu Programmierumgebungen für "Medien und Informatik"
 - ↓ 1. Worum geht es?
 - ↓ 2. Wie sieht die Ausgangslage aus?
 - ↓ 3. Warum braucht es Empfehlungen?
 - ↓ 4. Eine oder mehrere Umgebungen?
 - ↓ 5. Was sind die Anforderungen an eine Programmierumgebung für das Modul Medien und Informatik?
 - ↓ 5.1 Didaktisierte oder professionell genutzte Sprache?
 - ↓ 5.2 Textbasierte Sprache oder Blocksprache?
 - ↓ 6. Empfehlung: Scratch & scratchkompatible Sprachen
 - ↓ 6.1 Was ist Scratch?
 - ↓ 6.2 Warum Scratch?

Voraussetzungen

- **Benötigt wird eine Programmierumgebung für die 5./6. Klasse Primarschule und die Sekundarstufe I**
- **Es geht um Allgemein- und nicht Berufsbildung**
- **In der 5./6. Klasse keine Fachlehrpersonen**
- **Derzeit wenig bis kein Informatikvorwissen bei den Lehrpersonen**
- **Beschränkte Ressourcen zur Weiterbildung von Lehrpersonen, die "Medien und Informatik" unterrichten werden**
- **Derzeit wenig kohärentes, zum Modullehrplan "Medien und Informatik" passendes Unterrichtsmaterial**
- **Heterogene Infrastrukturvoraussetzungen in den Schulen.**

Warum braucht es Empfehlungen?

- **Mangelndes Vorwissen der Lehrpersonen:** Weil die meisten Lehrpersonen bisher über wenig Informatik-Kompetenzen verfügen, wären sie mit der Auswahl selbst überfordert.
- **Wenig Material vorhanden:** Weil bisher wenig stufenspezifisches Unterrichtsmaterial existiert, fördern konkrete Umgebungsempfehlungen die Nutzbarkeit des zu entwickelnden Materials

Eine oder mehrere Umgebungen?

- Parallel mehrere Umgebungen?
- Andere Umgebung in der Primarschule als auf der Sek-Stufe?

Anforderungen an eine Sprache für M+I

- **Motivierend**
- **Leichter Einstieg**
- **Vielseitig verwendbar**
 - **Sprachumfang/Ausrichtung**
 - **Technische Verfügbarkeit**
 - **Lizenzrechtliche Verfügbarkeit**
- **Längerfristig verfügbar**
- **Mächtig**
- **Enaktiv**

Didaktisiert oder professionell genutzt?

The image displays a screenshot of an IDE (likely Eclipse) with several panels open, illustrating a development environment for a web application.

- Left Panel (State Machine):** Titled "Kara, der programmierbare Matriküler", it shows a state machine diagram with states "start", "track", and "Stop". Below the diagram is a control panel with buttons and a "track" dropdown menu.
- Project Explorer (Files):** Shows the project structure, including folders like "AsyncRequests", "Web Pages", "src", and "web".
- Source Editor:** Displays the Java code for `AjaxCometServlet.java`. The code includes annotations like `@WebServlet` and implements methods like `destroy()`, `doGet()`, and `doPost()`.
- Members Navigator:** Lists the methods of the `AjaxCometServlet` class, such as `destroy()`, `doGet()`, and `doPost()`.
- Hierarchy:** Shows the class hierarchy, including `HttpServlet`, `GenericServlet`, and `Servlet`.
- Properties:** Displays the properties of the `AjaxCometServlet` class, including its name, extension, and file size.
- Javadoc:** Shows the Javadoc for the `destroy()` method, explaining that it is called by the servlet container to indicate that the servlet is being taken out of service.

Grafisch/Blockbasiert oder textuell?

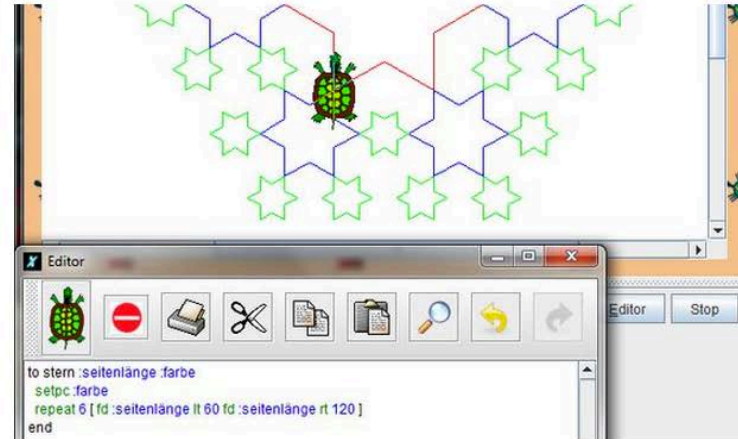
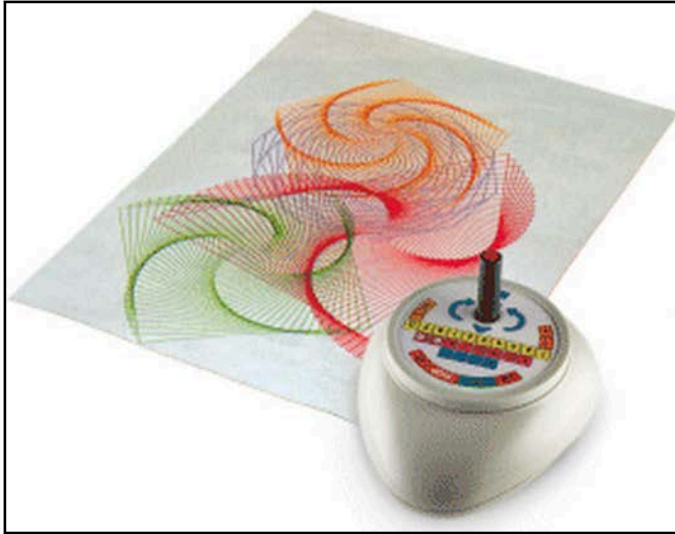


```
Python script
1 def azureml_main(expr_as_frame):
2     import pandas as pd
3     expr = expr_as_frame.iat[0,0]
4     result = pd.DataFrame({'Expr': [expr], \
5                             'Result': [eval(expr)]})
6     return result,
```

doebe.li/a1273 Blockbasierte Sprachen verhindern Syntaxfehler

doebe.li/a1274 Blockbasierte Sprachen erleichtern den Einstieg ins Programmieren

Edu-Programmiersprachen gibt es wie Sand am Meer...



Die Geschichte von Scratch



Jean Piaget:
Konstruktivismus



Seymour Papert:
Konstruktionismus
& Logo



`forward 50`



Mitchel Resnick:
NetLogo &
Scratch



Von Logo zu Scratch



`forward 50`



Mitchel Resnick (2012)

“Learning lessons from Papert’s experiences of Logo, we’ve designed Scratch to move beyond Logo along three dimensions, making programming more tinkerable, more meaningful, and more social.”

Grundprinzipien von Scratch

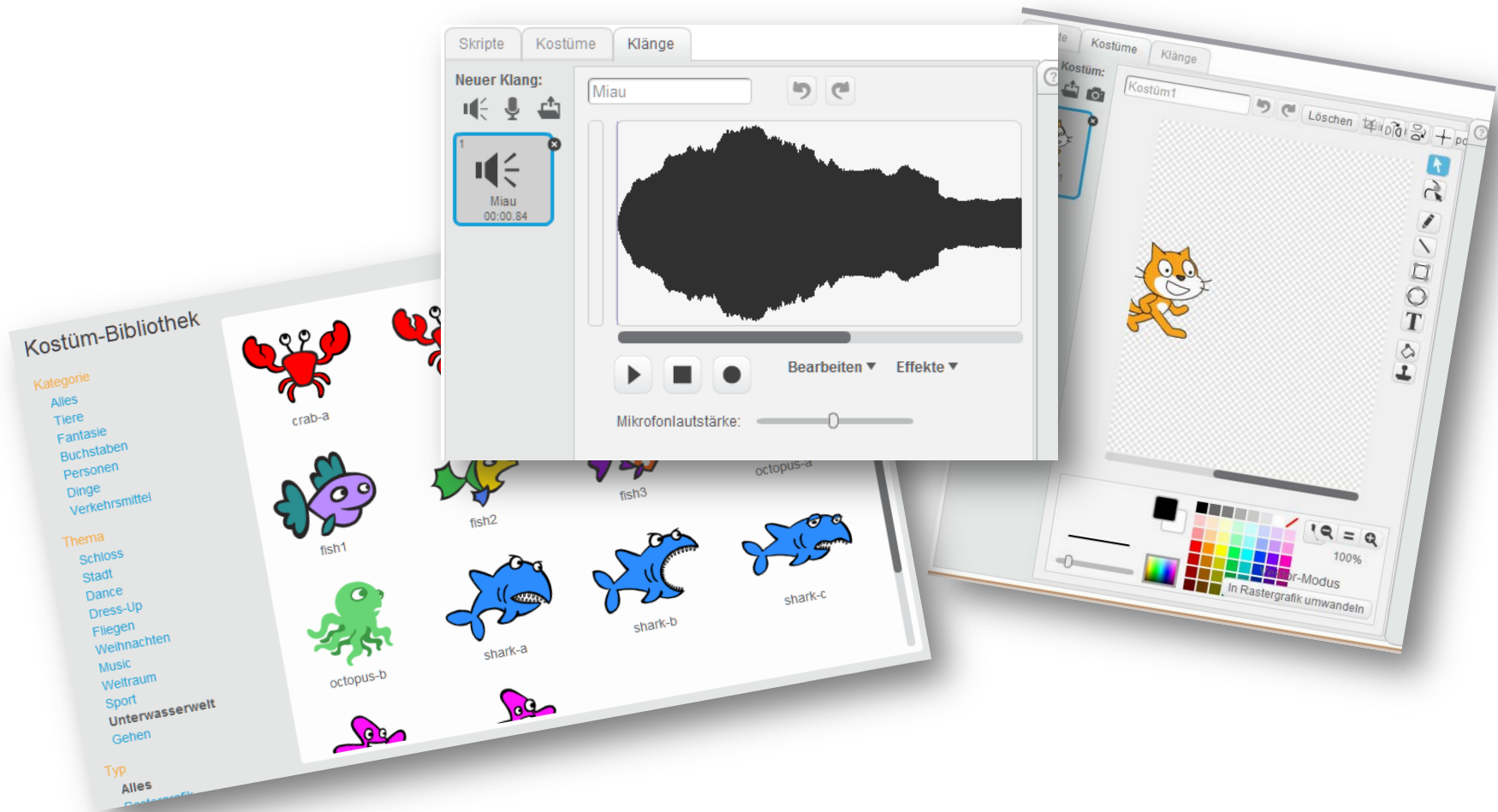


«more meaningful»

«more tinkerable»

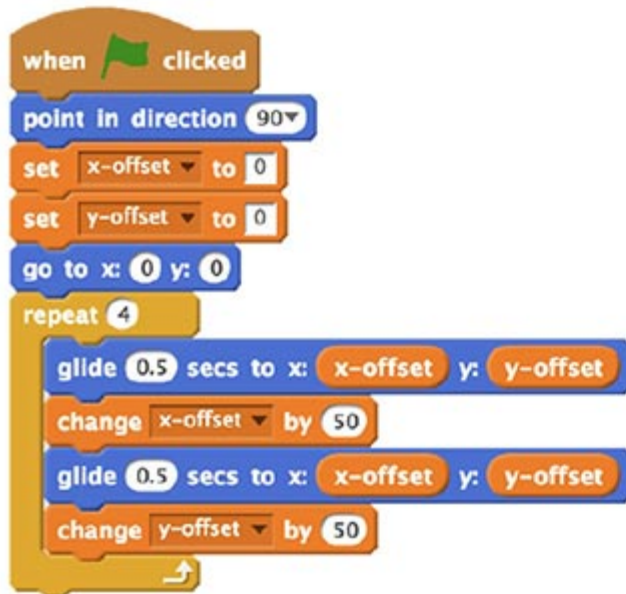
«more social»

imagine : More meaningful



program

More tinkerable: block vs. text

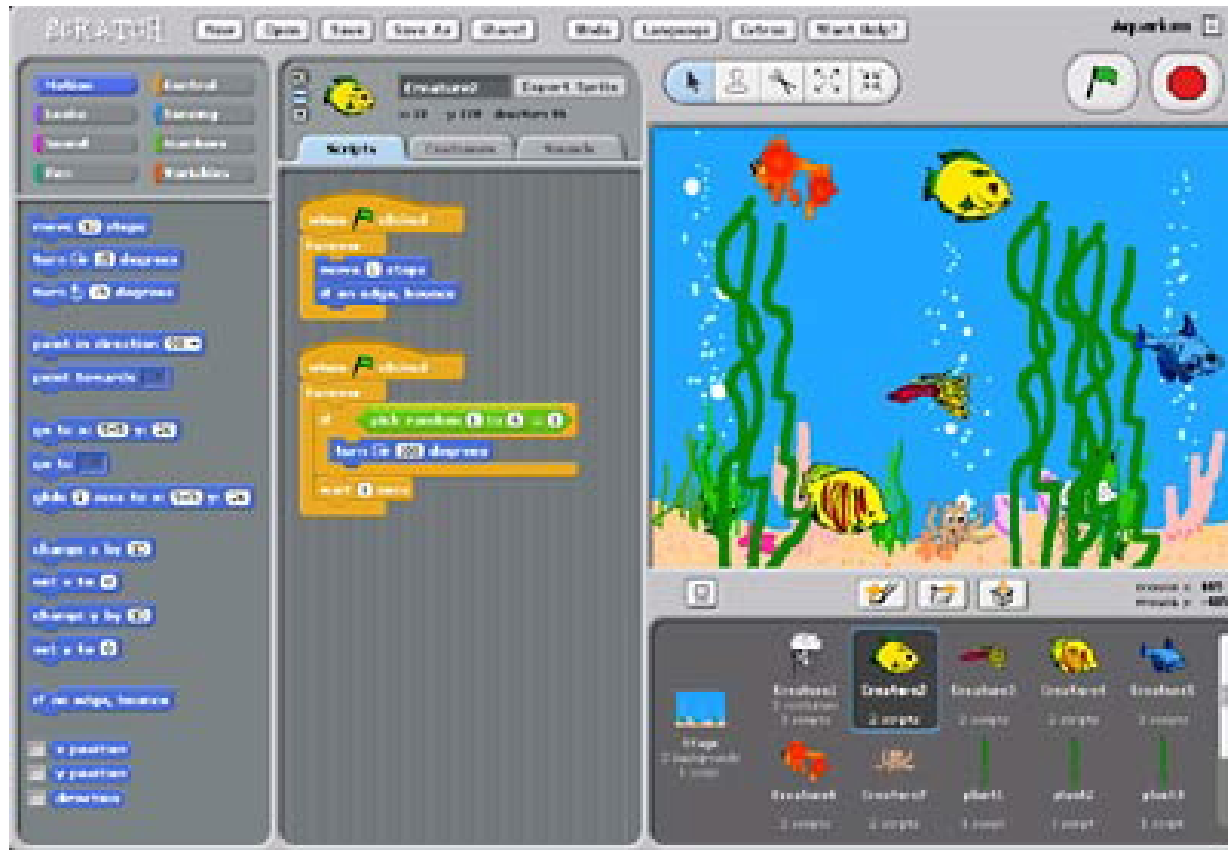


```
$(document).ready(function(){
  var $sprite = $('#.sprite');
  var xOffset = parseInt($('#.sprite').css('left').
    replace(/px/, ''), 10);
  var yOffset = parseInt($('#.sprite').css('top').
    replace(/px/, ''), 10);
  var i;

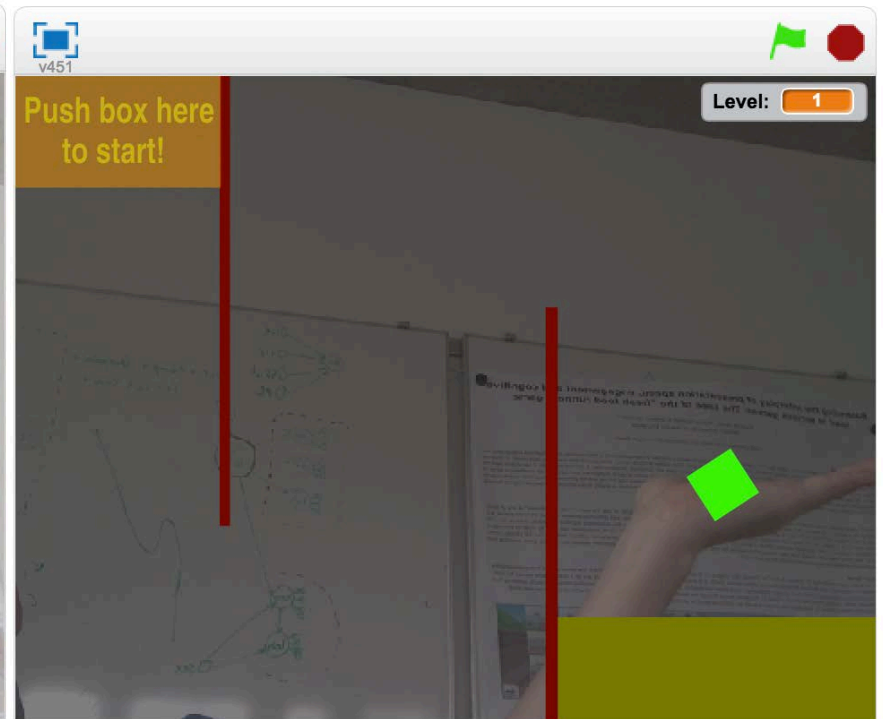
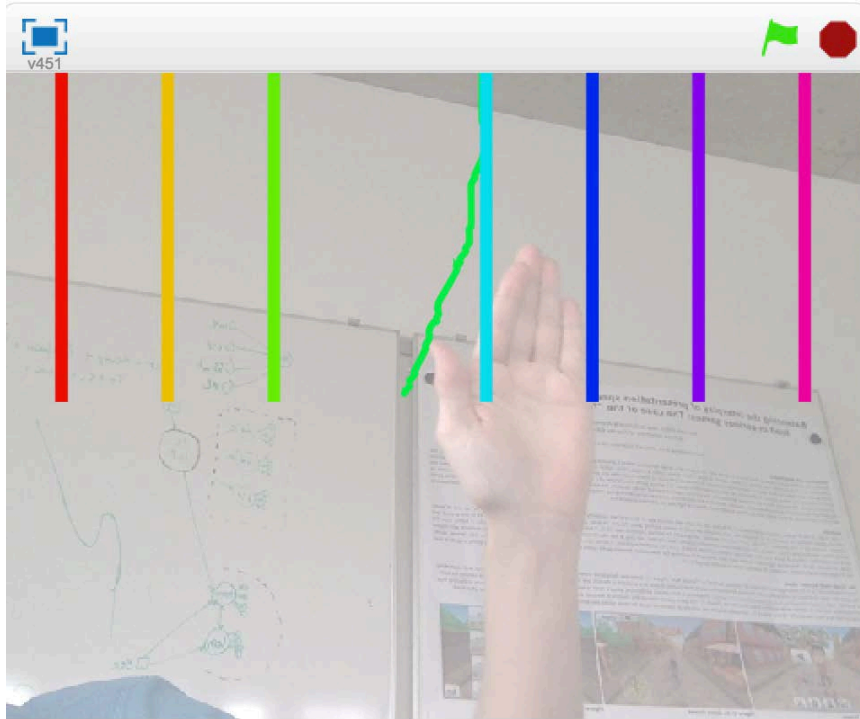
  for (i = 0; i < 4; i++) {
    xOffset += 50;
    yOffset -= 50;
    $sprite.animate({
      left: xOffset
    }).animate({
      top: yOffset
    });
  }
});
```

program

: More tinkerable: live coding



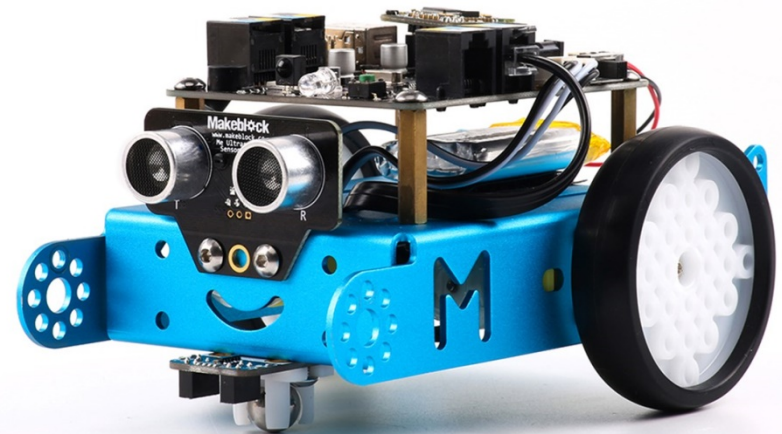
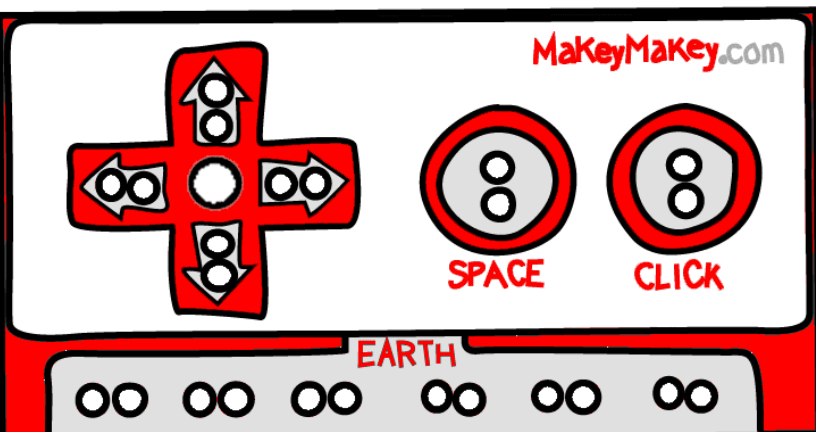
Herangehensweise: Bottom-up statt top-down



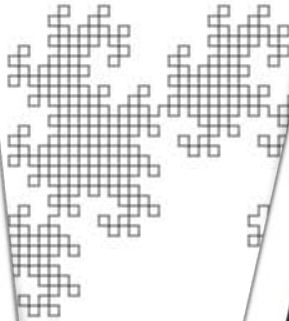
<https://scratch.mit.edu/studios/201435/>

program

More tinkerable: Sensoren und Aktoren

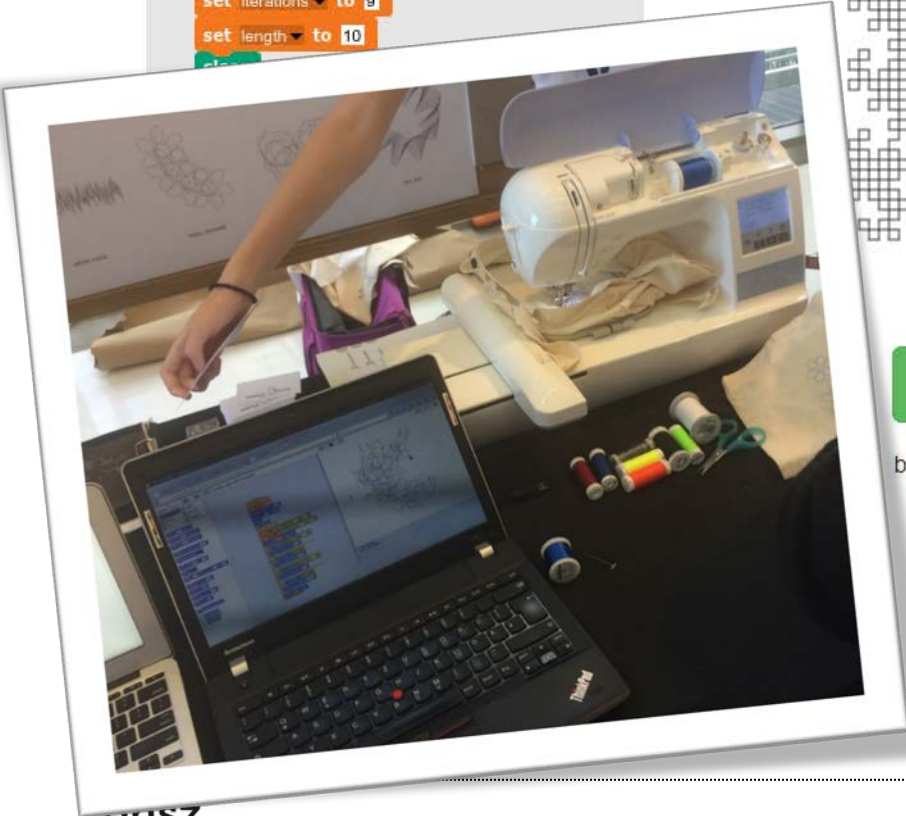


code! - draw! - stitch!



Run now!

by [STITCHCO](#)





: More sociable: ScratchR

Erschaffe Geschichten, Spiele, und Animationen
Teile sie mit anderen weltweit

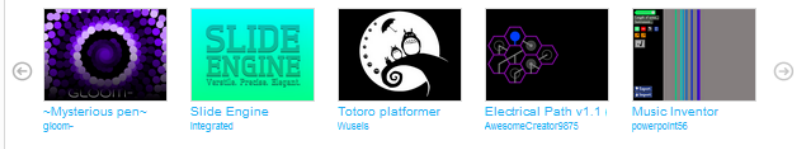


Eine kreative Lerngemeinschaft mit **17.758.764** geteilten Projekten

UBER SCRATCH | FÜR LEHRKRÄFTE | FÜR ELTERN



Vorgestellte Projekte

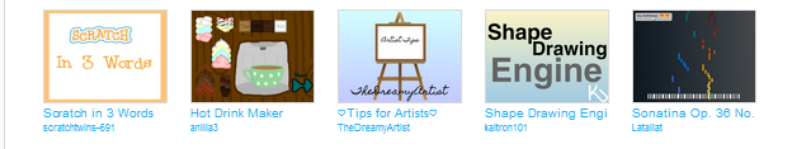


Vorgestellte Studios



Projekte vorgestellt von Kurator ForeverArtist

Mehr erfahren



Community statistics at a glance

- 17'731'978** projects shared,
 - 14'435'279** users registered,
 - 93'664'866** comments posted,
 - 2'970'475** studios created
- ...and growing!

Website traffic last month

- 144'777'540** pageviews
- 27'845'820** visits
- 13'540'550** unique visitors

scratch.mit.edu: Das YouTube des Programmierens



: More sociable: ScratchR

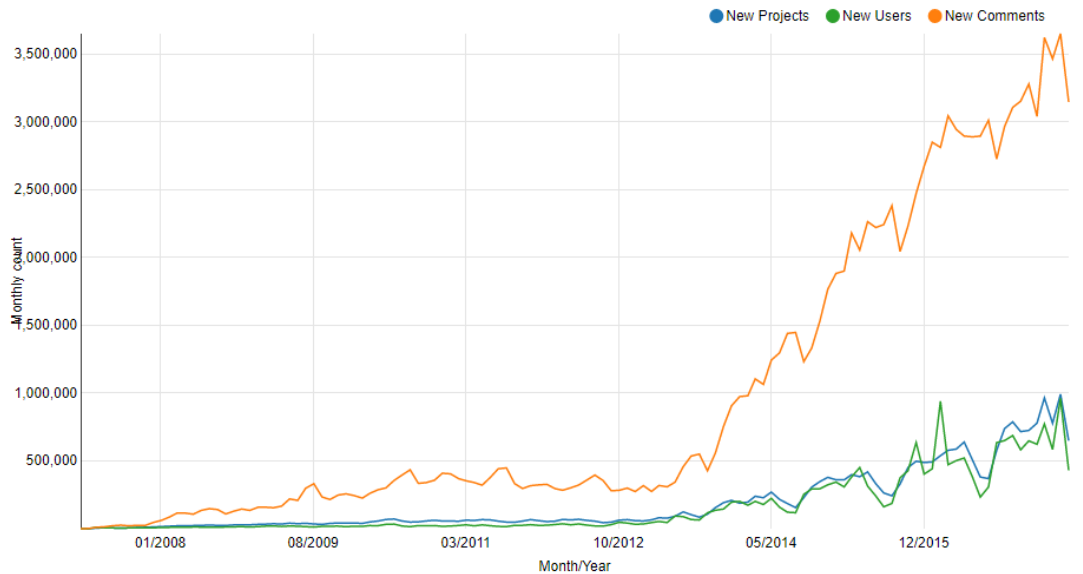
Community statistics at a glance

- 🐱 24.028.090 projects shared,
 - 🐱 19.961.818 users registered,
 - 🐱 123.327.576 comments posted,
 - 🐱 3.633.643 studios created
- ...and growing!

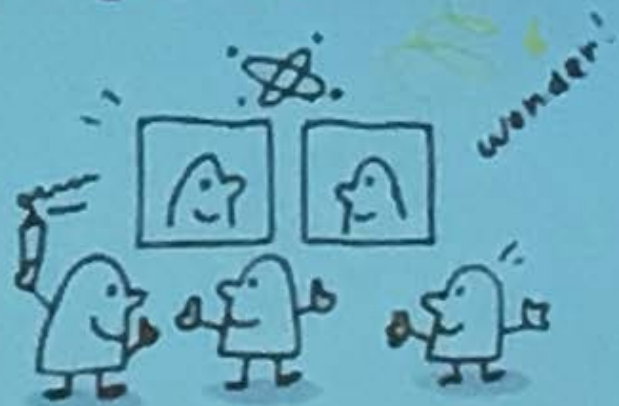
Website traffic last month

- 🐱 100.126.650 pageviews
- 🐱 17.559.420 visits
- 🐱 9.076.330 unique visitors

Monthly Activity Trends



projects



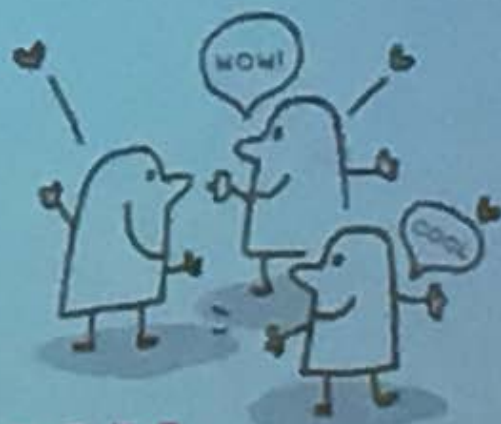
passion



play



peers



Passion

it is all about falling in Love with the material!

EDUCATION HAS VERY LITTLE TO DO WITH EXPLANATION



Seymour grew up in South Africa. He wanted education to be available for ALL CHILDREN



Good morning



IF THINGS DON'T WORK THE WAY YOU LIKE, IT IS AN OPPORTUNITY TO LEARN...

change things, TRY AGAIN!

Play

ENGAGE KIDS TO EXPERIMENT & TINKER

Purpose: Give a voice to everyone



COCEHDI-IPN:

Qualitan Experience

Early Childhood Education

with Technology

Ecogstun for the educational technology

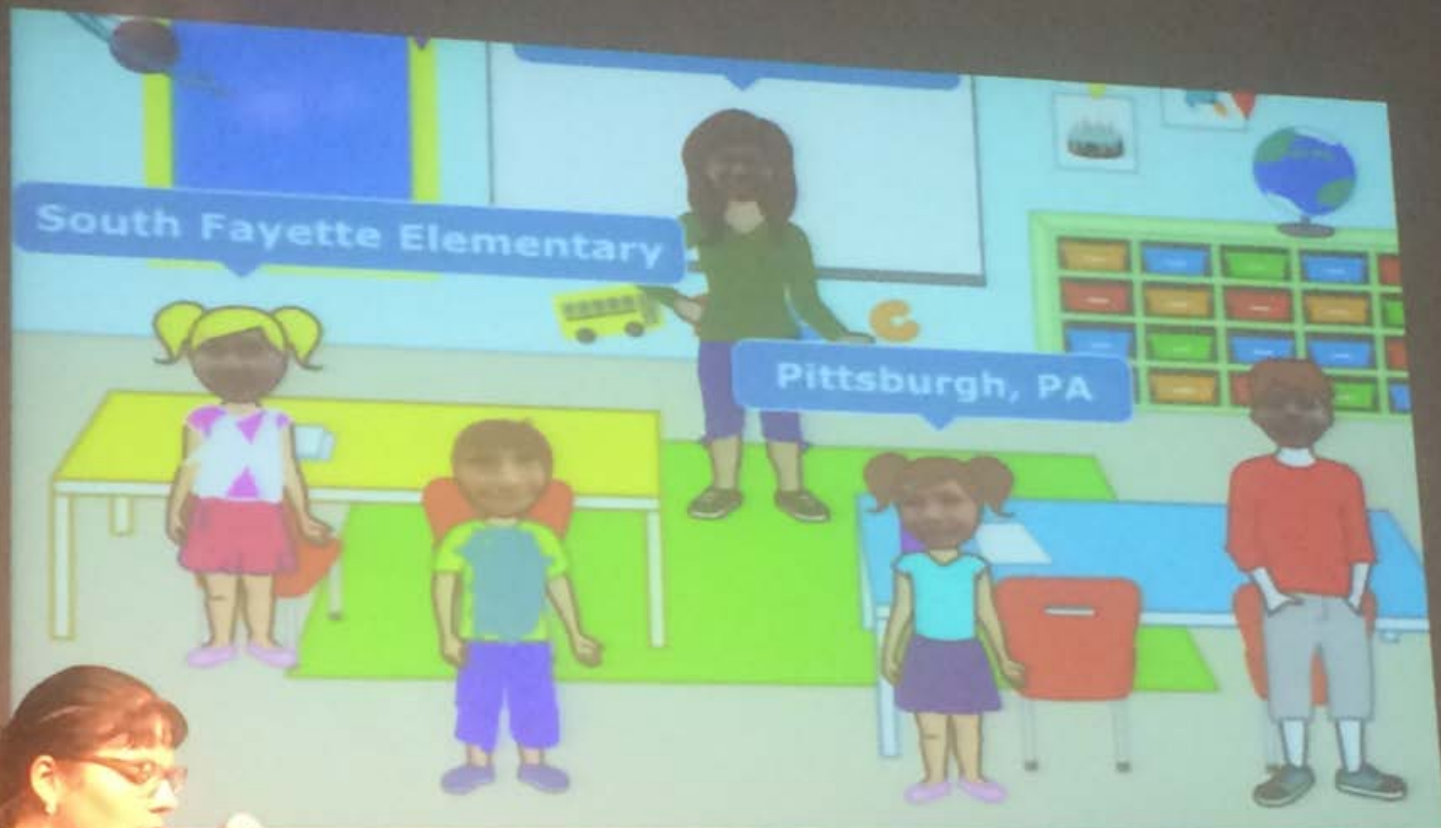
development, and training ?

in science, technology and

innova

Cocendi-IPN: A Mexican
experience in early
childhood education with
technology





K-2 CODE KIDS

SOUTHFAYETTE.ORG

WRITING IN CODE
IS LIKE WRITING
IN WORDS—IT IS
HOW WE
COMMUNICATE
WITH
COMPUTERS.

2nd Grader





«computational participation»

[doebe.li/b5889](https://doi.org/10.1017/9781017058889)