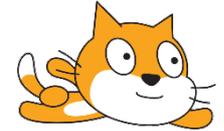


Scratch 3.0 Vertiefungskurs

Kurstag 2





Kurstag 1 (Mittwoch, 24.2.21, 13.00 - 16.00 Uhr)

- Programmierkonzepte wie Schleifen, Bedingungen, Variablen, Unterprogramme usw. verstehen und anwenden können.
- Strukturiertes Vorgehen bei der Umsetzung einer Projektidee mittels Dekomposition, Mustererkennung und Abstraktion.

Kurstag 2 (Mittwoch, 24.3.21, 13.00 – 16.00 Uhr)

- Vorgehensweise bei der Fehlersuche.
- Scratch Erweiterungen mit LEGO Mindstorms, LEGO WeDo, MakeyMakey, micro:bit.
- Didaktische Hinweise (Kursaufbau, Best Practices, Materialien, Beurteilung usw.).
- Themen nach Wunsch.

Optional aber empfohlen: Für die Wochen zwischen den Kurstagen kann an einem individuellen Projekt gearbeitet werden.

Vorgehensweise bei der Fehlersuche



Ein aufgeräumter Code verhindert Fehler und erleichtert die Fehlersuche



- Saubere, kompakte Darstellung des Codes. Anwenden der Programmierkonzepte: sich wiederholende Programmteile in Unterprogrammen auslagern oder in Schleifen zusammenfassen.
- Dokumentieren der Programmteile über die Kommentarfunktion beim Block.
- Sinnvolle Namensgebung bei Variablen und Unterprogrammen (damit die Bezeichnung bereits die Funktion beschreibt).
- Wenn etwas funktioniert, abspeichern und mit einer Kopie und Versionsnummer weiterarbeiten. So kann man auf frühere funktionierende Versionen zurückgreifen.
- Programmteile aufräumen: Code, der unter einer Figur gespeichert ist, soll nur mit der Figur zu tun haben. Allgemeiner Code besser unter der «Bühne» abspeichern.

«Sie, es gaht nöd!» – Was tun?



1) Rückfragen:

- Was genau funktioniert nicht?
- Hat es früher mal funktioniert?
- Was hast du hinzugefügt oder geändert seit es das letzte Mal funktioniert hat?
- Was möchtest du genau machen?

2) Checkliste der häufigsten Fehler durchgehen:

- <http://ilearnit.ch/download/ScratchProjektideen.pdf> (Seite 9-10)



- Kopie erstellen
- Aus der Sichtweise des Computers denken. Code nachvollziehen.
- Variablenwerte anzeigen.
- Doppelklick auf Programmcode Teile, um diese direkt zu testen.
- Programmcode isolieren durch wegziehen des Rests des Codes

Debuggen gezielt üben: Debug It!
Übungen von Scratch Creative
Computing Curriculum
herunterladen

[http://scratched.gse.harvard.edu/
guide/curriculum.html](http://scratched.gse.harvard.edu/guide/curriculum.html)

Debug It!

DEBUG IT!

HELP! CAN YOU DEBUG THESE FIVE SCRATCH PROGRAMS?

In this activity, you will investigate what is going wrong and find a solution for each of the five Debug It challenges.

START HERE

- 1. Go to the Unit 1 Debug It! studies: <http://scratched.gse.harvard.edu/studies/1-1-181>
- 2. Test and debug each of the five debugging challenges in the studio.
- 3. Write down your solution or submit the buggy program with your solution.



- 1. Make a list of possible bugs in the program.
- 2. Keep track of any work! This can be a useful reminder of what you have already tried and point you toward what to try next.
- 3. Share and compare your problem finding and problem solving approaches with a neighbor until you find something that works for you!

1. DEBUG IT! 1. <http://scratched.gse.harvard.edu/studies/1-1-181>

When the green flag is clicked, both Goto and Scratch Cat should start dancing. But only Scratch Cat starts dancing! How do we fix the program?

2. DEBUG IT! 2. <http://scratched.gse.harvard.edu/studies/1-1-181>

In this project, when the green flag is clicked, the Scratch Cat should start on the left side of the stage, say something about being on the left side of the stage, and say something about being on the right side. It needs the function the green flag is clicked, but not again. How do we fix the program?

3. DEBUG IT! 3. <http://scratched.gse.harvard.edu/studies/1-1-181>

The Scratch Cat should do a flip when the space key is pressed. But when the space key is pressed, nothing happens! How do we fix the program?

4. DEBUG IT! 4. <http://scratched.gse.harvard.edu/studies/1-1-181>

In this project, when the green flag is clicked, the Scratch Cat should say "Hello, everyone!" in a speech bubble and a sound. But the speech bubble happens before the sound - and the Scratch Cat only smiles on "Hello, everyone!" How do we fix the program?

5. DEBUG IT! 5. <http://scratched.gse.harvard.edu/studies/1-1-181>

Share your testing and debugging practices with a partner. Make notes of the similarities and differences in your debugging.

- Add code commentary by right-clicking on blocks in your script. This can help others understand different parts of your program.
- Help a neighbor!

Objective

In this activity, students investigate a problem and find a solution, while developing debugging strategies.

Standards Alignment

CSTA Standards: 1A-AP-14, 1B-AP-15, 2-AP-17

K12CS Concepts: Computing Systems, Algorithms and Programming
K12CS Practice 6: Testing and Refining Computational Artifacts

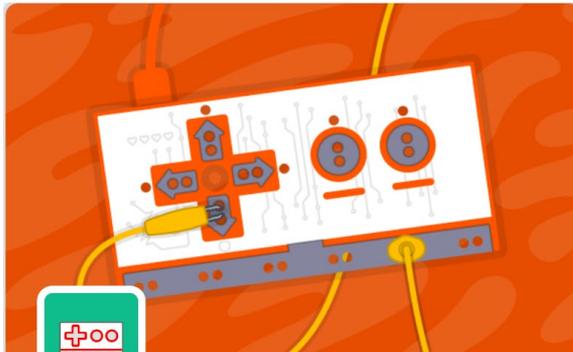
[Preview](#) | [Edit](#) | [Download](#)

Verbindung mit der physischen Welt





Makey Makey



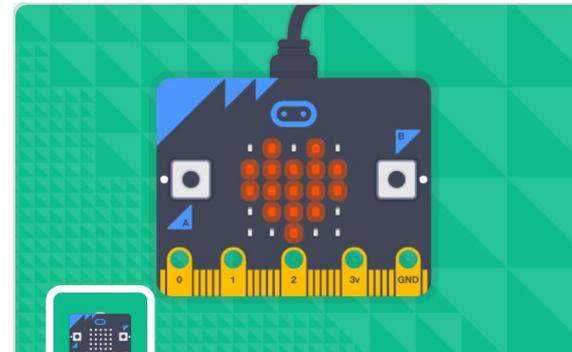
Makey Makey

Mach alles zu einer Taste.

Zusammenarbeit mit
JoyLabz

<http://ilearnit.ch/download/MakeyMakeyProjektideen.pdf>

micro:bit



micro:bit

Verbinde deine Projekte mit der
realen Welt.

Erfordert



Zusammenarbeit mit
micro:bit

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



LEGO WeDo



LEGO Education WeDo 2.0

Baue mit Motoren und Sensoren.

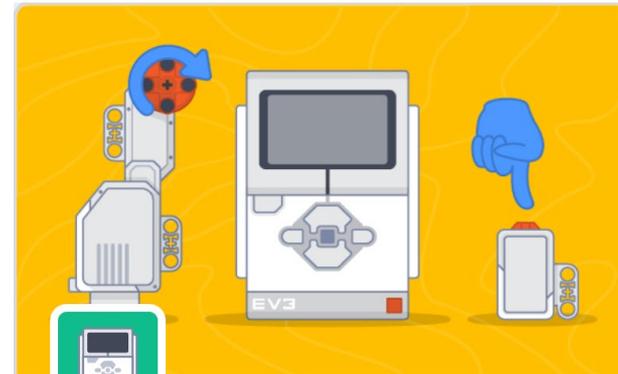
Erfordert



Zusammenarbeit mit

LEGO

LEGO Mindstorms EV3



LEGO MINDSTORMS EV3

Baue interaktive Roboter und mehr.

Erfordert

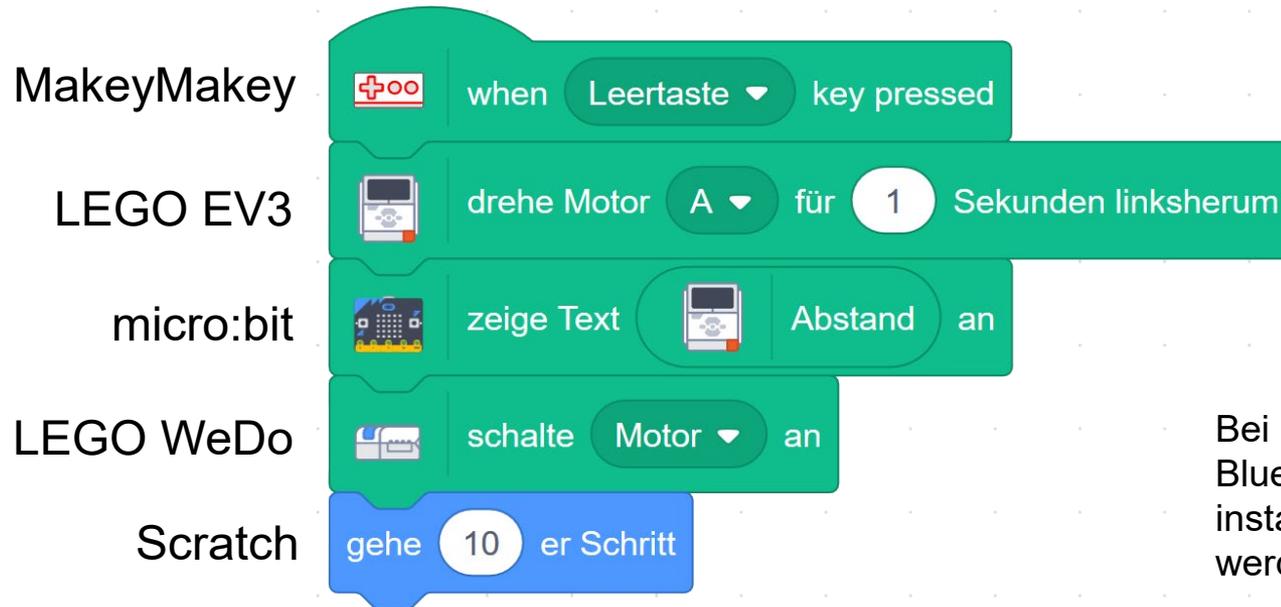


Zusammenarbeit mit

LEGO

→ Bis jetzt noch keine Scratch Extension vorhanden für die neuen Modelle LEGO Spike Prime und LEGO Robot Inventor! Kommt aber bestimmt noch.

Verbindung mit der physischen Welt

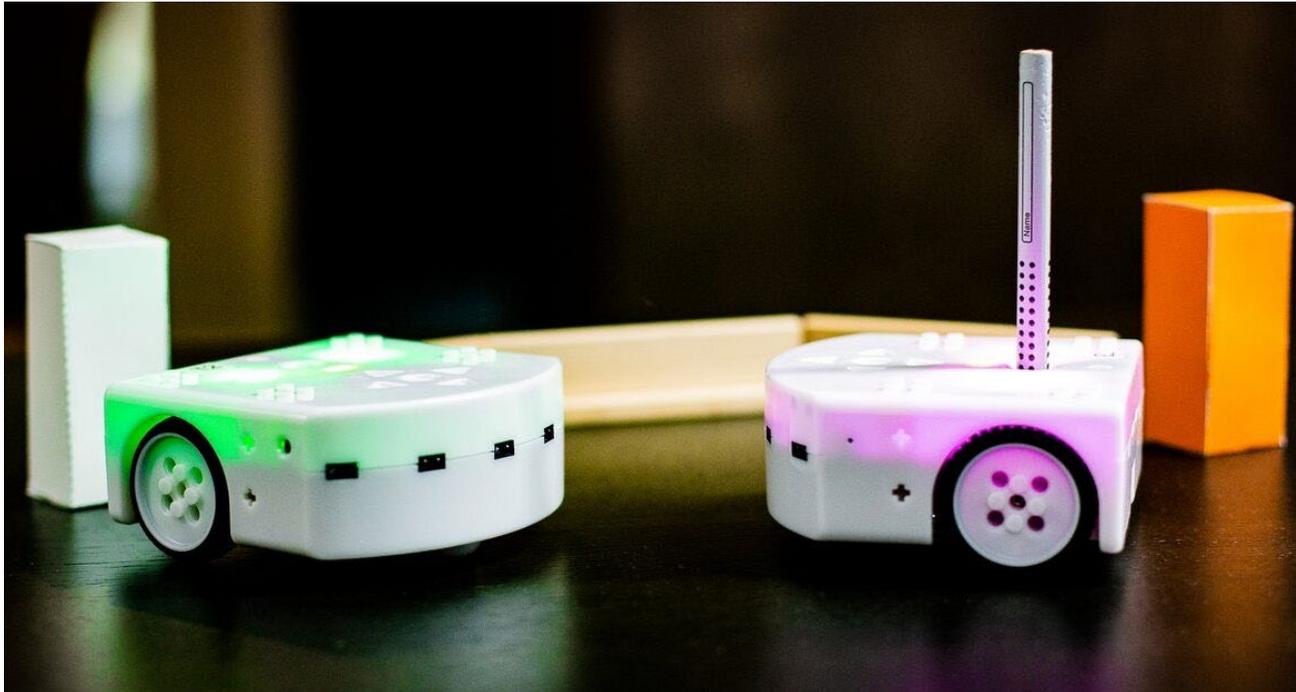


Bei kabelloser Verbindung über Bluetooth muss Scratch Link installiert sein und gestartet werden



Alle Extensions funktionieren gleichzeitig!

Thymio Roboter mit Scratch programmieren

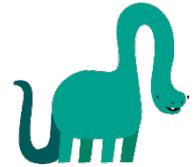


Umfangreicher und frei zugänglicher Thymio MOOC in Deutsch. Von der EPFL erstellt in Zusammenarbeit mit Pädagogischen Hochschulen

<https://courseware.epfl.ch/courses/course-v1:EPFL+THYMIO-DE+2020/about>

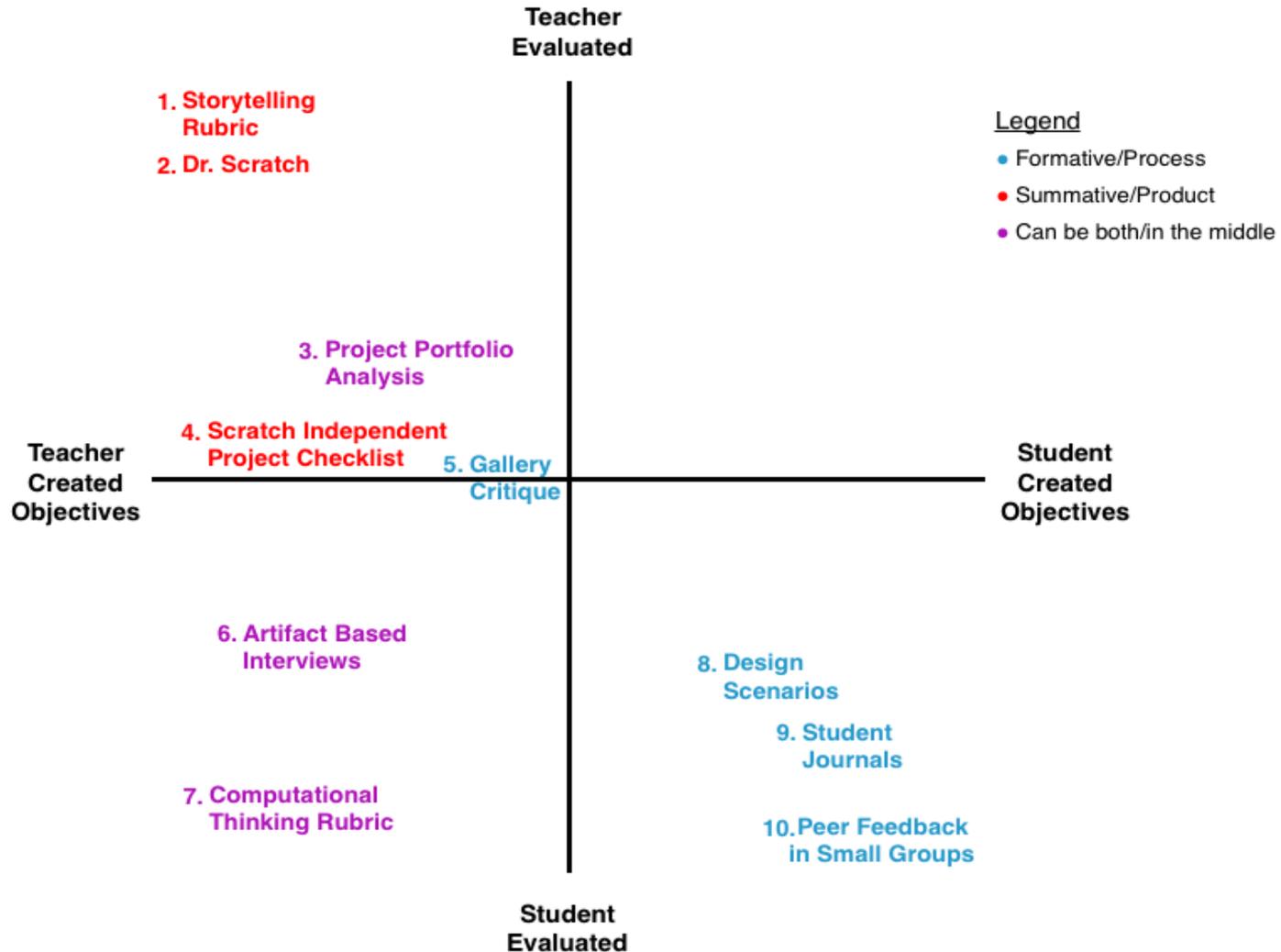
Didaktische Hinweise





VisualLightBox.com

Beurteilung von Programmierkompetenzen





Für Selbstbeurteilung oder gegenseitiges Feedback (Peer-feedback)

Checklisten legen Standards für die Qualität der Arbeit fest. Sie können vom Lehrer oder in Zusammenarbeit mit den Schülern entwickelt werden.

Während der Projektarbeit, können die Schülerinnen und Schüler gegenseitiges Feedback geben oder durch Selbstbeurteilung den eigenen Fortschritt mit den geplanten Lernzielen vergleichen.

Durch diesen Prozess sollten die Schülerinnen und Schüler die Stärken ihrer Arbeit erkennen, sowie die Bereiche, die noch verbessert werden sollen. Von Vorteil sind Verbesserungsvorschläge als Resultat des gegenseitigen Feedbacks.

Wenn man die gegenseitigen Feedbacks öfters wiederholt, gewöhnen sich die Schülerinnen und Schüler an den Prozess und das Feedback wird immer konstruktiver. Das Verständnis für die Kriterien auf der Checkliste nimmt ebenfalls zu.

Beispiel einer Checkliste



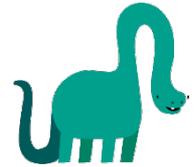
Kategorie	Niveau 1 (1 Punkt)	Niveau 2 (2 Punkte)	Niveau 3 (3 Punkte)
Entwurf auf Papier	Entwurf stimmt nicht mit dem entwickelten Endprodukt überein	Entwurf beinhaltet nur einige der umgesetzten Ideen, skizziert aber nicht ganz das Endprodukt	Entwurf beschreibt mehr oder minder vollständig das entwickelte Endprodukt
Entwicklungs-Tagebuch	praktisch keine Dokumentation des Entwicklungsprozesses, nur einige grobe Skizzen des Endprodukts	grobe Dokumentation, einigen konkreten Problemstellungen beschrieben, keine Entscheidungen oder Überlegungen festgehalten	weitgehend vollständige und umfangreiche Dokumentation, Design-Entscheidungen und Alternativen wurden festgehalten
Endprodukt / Programmieren	kein klarer Sinn des Produkts erkennbar	Sinn des Produkts erkennbar aber nicht offensichtlich oder nicht selbsterklärend	Sinn des Produkts ist klar und logisch
	keine Interaktionsmöglichkeiten für den Betrachter	wenige Interaktionsmöglichkeiten für den Betrachter	klare Interaktionsmöglichkeiten für den Betrachter mit Anleitung / Hilfestellung
	Figuren wurden nicht sinnvoll benannt, Skripte sind durcheinander und zeigen keine Ordnung	einige Figuren wurden sinnvoll benannt und Skripte im Anzeigebereich in der Regel aufgeräumt und organisiert	alle Figuren und Skripte wurden sinnvoll benannt, organisiert und teilweise kommentiert
	mehrere Befehlsblöcke wurden unsinnig zusammengesetzt, werden nie erreicht oder sind unnötig	alle Befehlsblöcke wurden sinnvoll verwendet und werden auch tatsächlich erreicht	es wurden bewusst Skripte mit Schleifen oder eigenen Blöcken optimiert, Figuren wiederverwendet usw.
	Produkt hat viele Bugs und Fehler, die nicht bedacht wurden	Produkt hat noch einige Bugs und Fehler, einige Fällen wurden nicht beachtet	Produkt hat praktisch keine Bugs und Fehler, alles wurde gut getestet
	keine eigenen Grafiken oder Töne verwendet	einzelne eigene Grafiken oder Töne verwendet	verwendet vielfach eigene Grafiken oder Töne

Beispiel einer Checkliste



Kategorie	Niveau 1 (1 Punkt)	Niveau 2 (2 Punkte)	Niveau 3 (3 Punkte)
Entwicklungsprozess	keine Anpassungen am Entwurf, kein Versuch von alternativen Lösungswegen, keine Überarbeitungen erkennbar	Entwurfsideen wurden bei Bedarf überarbeitet, alternative Lösungen ausprobiert, Überarbeitungsprozess erkennbar	Entwurf wurde fortlaufend angepasst, alternative Lösungen abgewogen, getestet und ggf. dokumentiert
	Unterrichtszeit wurde schlecht genutzt, Termine nicht eingehalten	Unterrichtszeit wurde meist gut genutzt, Termine wurden in der Regel eingehalten	Unterrichtszeit wurde immer gut genutzt, Termine wurden eingehalten, ggf. selbstständige Weiterarbeit
Überfachliche Kompetenzen Selbstständigkeit Eigenständigkeit Selbstreflexion	zeigt kaum persönliches Engagement, hat eine negative Einstellung zum Arbeiten.	zeigt in der Regel persönliches Engagement, ist meist konzentriert an der Arbeit	stets hohes persönliches Engagement (Motivation, Selbständigkeit, allgemeine positive Einstellung) und ist konzentriert bei der Arbeit.
Überfachliche Kompetenzen Kooperationsfähigkeit	keine Zusammenarbeit im Team. Hat eine negative Einstellung zu der Zusammenarbeit im Team. Kooperiert nicht.	wenig oder einseitige, Zusammenarbeit im Team, kooperiert meistens aktiv und respektvoll im Team.	sehr gute, sich ergänzende Zusammenarbeit im Team. Kooperiert aktiv und respektvoll im Team.

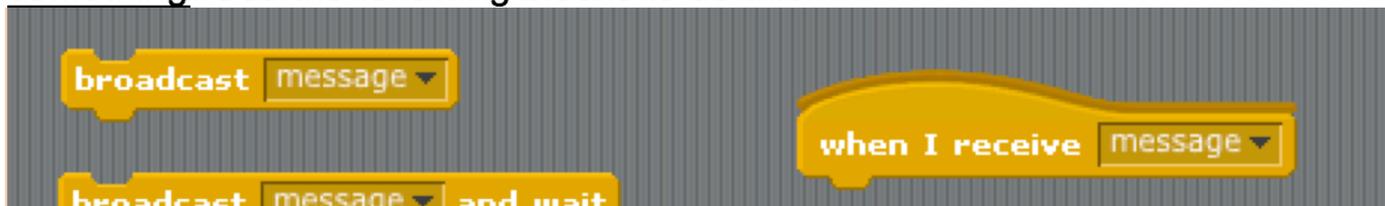
Beispiel einer Checkliste



1. **Sprites:** Your project must include at least 2 sprites. At least one of them must be your original drawing.
2. **Costumes:** The sprite you draw by hand must have at least 3 costumes that appear throughout the project.
3. **Stage:** You must use a stage. It may be drawn by hand or imported.
4. **Motion:** At least one of the sprites must move at some point in the project, using the (x,y) coordinate system to direct its motion.
5. **Loops:** Your project must use at least one loop, created using one of the blocks shown below.



6. **Broadcasting & Receiving Messages:** The stage or one of the sprites must broadcast at least one message, which must be received by another object, causing it to do something. Use the following blocks to do this.



Beispiel einer Checkliste



7. → **Variables**: You must use a variable to create a switch that turns on or off when the variable reaches a certain number, causing something to happen. ¶



8. → **Query (“Ask”)**: The program must ask the user to answer a question. Something should happen depending on their answer. ¶



9. → **Story/Experience**: Your project should use all these pieces to tell a story or provide a meaningful experience for the user. It should be creative and fun. ¶



WHY?

HOW?

WORKING ON

CONTACT US

ORGANIZATIONS



SIGN UP

SIGN IN ▾

Analyze your Scratch projects

Welcome to the Dr. Scratch website, an analytical tool that evaluates your Scratch projects in a variety of computational areas. This analyzer is a helpful tool to evaluate your own projects, or those of your Scratch students.

LEARN MORE

There are two options to analyze your Scratch project now!

1. Introduce the **url** of your Scratch project, you don't have to download it:

ANALYZE BY URL

2. If you have your **project** downloaded in the computer you can analyze it here:

Choose Project

ANALYZE MY PROJECT

Automatisierte Analyse von Scratch Programmen



HELP

DR. SCRATCH(BETA VERSION)



Score: **11/21** [Tweet](#)

The level of your project is...
DEVELOPING!

You're doing a great job. Keep it up!!!

[Come back to your Scratch project.](#)

Best practice

1 sprite attributes.

0 sprite naming.

Project certificate

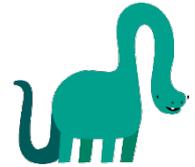
<https://scratch.mit.edu/projects/157900135/>

Download

Level up

Level





The Dr. Scratch team
has the honour of presenting this

CERTIFICATE

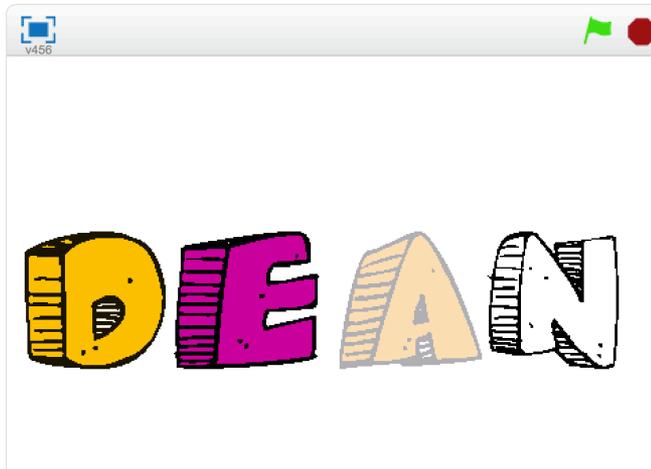
to the project <https://scratch.mit.edu/projects/157900135/>
because it has obtained a score of
11/21

This project has been analyzed with Dr. Scratch (www.drscratch.org).

Dr. Scratch aims to provide a means of learning and feedback on the quality of the projects in Scratch.



- Die Lernenden bekommen ein Scratch Projekt präsentiert und müssen in 4 Bereichen etwas tun:
- (1) erklären, was das ausgewählte Projekt tut,
- (2) beschreiben, wie es erweitert werden könnte,
- (3) einen Fehler beheben und
- (4) Hinzufügen einer neuen Funktion.



Anleitung

In the Name project, Dean (the project creator) has designed an animated project that features his name. How could we extend this project? Dean wants the N to appear after the A, not at the same time. What is the bug? How do we fix the bug? Dean wants the N to do something interesting (like the other letters), but only when the N is clicked. How do we add this feature?

Anmerkungen und Danksagungen

Part of a computational thinking assessment strategy.
<http://scratched.gse.harvard.edu/ct/>

Artefakt-basierte Interviews am Anfang, in der Mitte und am Ende des Projekts



INTERVIEWING STUDENTS ABOUT SCRATCH PROGRAMMING EXPERIENCE

One approach to assessing learners' development is through conversations about their projects and processes. This protocol engages learners in conversations about their projects at the beginning, middle, and end of their Scratch experience.

AT THE BEGINNING OF THE SCRATCH UNIT

Defining Scratch
Ask the learner to define Scratch, and explain its functionality.

1. If a friend wasn't here today and asked you what Scratch is, and what you can do with it, what would you tell them?

Project Feedback
Share two Scratch projects with the learner and ask them to provide feedback to the project creator for one of the projects.

2. Is there anything you would want to ask the creator before giving them feedback?
3. How do you think the project could be improved? What suggestions would you give the creator to make the project more interactive?
4. Any ideas for how you would do this?

Debug It
Present the learner with a Debug It! challenge from Scratch Curriculum Guide, available online at <http://scratched.gse.harvard.edu/guide>

5. What's going on?
6. How would you fix it?
7. Want to give it try and do what you told me?
8. Did it work like you expected it to work?
9. Can you tell me what you think is going on?
10. (If the learner is not able to debug the project) Before we move on, where would you go for help if you wanted to fix this?

Project Process
Now that the class has been introduced to the development process for a current Scratch project.

11. Please show me a project you created. Why did you choose this project to share?
12. (If the project is a work-in-progress) What do you think you'll need to do to make it complete?
13. Where did you get that idea? What inspired you?
14. Did you plan this project before you started programming? If yes, what did you do to plan for it?
15. Can you share what you've done so far? (Look for any borrowed or imported assets (images, sounds) in the project) Where did you get those? What made you choose those?

DURING THE MIDDLE OF THE SCRATCH UNIT

Defining Scratch
Ask the learner to define Scratch, and explain its functionality.

1. If a friend wasn't here today and asked you what Scratch is, and what you can do with it, what would you tell them?

Project Feedback
Share two Scratch projects with the learner and ask them to provide feedback to the project creator for one of the projects.

2. Is there anything you would want to ask the creator before giving them feedback?
3. How do you think the project could be improved? What suggestions would you give the creator to make the project more interactive?
4. Any idea how you would do this?

Debug It
Present the learner with a Debug It! challenge from Scratch Curriculum Guide, available online at <http://scratched.gse.harvard.edu/guide>

5. What's going on?
6. How would you fix it?
7. Want to give it try and do what you told me?
8. Did it work like you expected it to work?
9. Can you tell me what you think is going on after your changes?
10. (If the student is not able to debug the project) Before we move on, where would you go for help if you wanted to fix this?

Project Process
Ask the student about their planning and development process for a recently completed Scratch project.

11. Can you tell me how you got the idea for that? What made you think of that?
12. Did you plan your project before you started programming? If yes, what did you do to plan for it?
13. Can you describe what you did in Scratch to make your project? (Ask about specific project elements – e.g., how did you get the character to disappear and reappear later in the story?)
14. Were there things that were particularly challenging? How did you figure out how to do that? Where did you go for help?
15. (Look for any borrowed or imported assets (images, sounds) in the project) Where did you get those? What made you choose those?
16. What are you most proud of about your project? What did you enjoy most about the process? What would you change? Why?
17. Did you share your project with anyone? If yes, who did you share it with? How? If no, do you plan to share it with anyone? Why or why not?

AT THE END OF THE SCRATCH UNIT

Defining Scratch
Ask the learner to define Scratch, and explain its functionality.

1. If a friend wasn't here today and asked you what Scratch is, and what you can do with it, what would you tell them?

Project Feedback
Share two Scratch projects with the learner and ask them to provide feedback to the project creator for one of the projects.

2. Is there anything you would want to ask the creator before giving them feedback?
3. How do you think the project could be improved? What suggestions would you give the creator to make the project more interactive?
4. Any idea how you would do this?

Debug It
Present the learner with a Debug It! challenge from Scratch Curriculum Guide, available online at <http://scratched.gse.harvard.edu/guide>

5. What's going on?
6. How would you fix it?
7. Want to give it try and do what you told me?
8. Did it work like you expected it to work?
9. Can you tell me what you think is going on after your changes?
10. (If the student is not able to debug the project) Before we move on, where would you go for help if you wanted to fix this?

Project Process
Ask the student about their planning and development process for a recently completed Scratch project.

11. Can you tell me how you got the idea for that? What made you think of that?
12. Did you plan your project before you started programming? If yes, what did you do to plan for it?
13. Can you describe what you did in Scratch to make your project? (Ask about specific project elements – e.g., how did you get the character to disappear and reappear later in the story?)
14. Were there things that were particularly challenging? How did you figure out how to do that? Where did you go for help?
15. (Look for any borrowed or imported assets (images, sounds) in the project) Where did you get those? What made you choose those?
16. What are you most proud of about your project? What did you enjoy most about the process? What would you change? Why?
17. Did you share your project with anyone? If yes, who did you share it with? How? If no, do you plan to share it with anyone? Why or why not?

Wiederkehrende Fragekategorien:

- Definiere Scratch
- Projekt Feedback
- Debuggen
- Projekt Prozess



Beispiel aus dem Leitfaden:

Sobald die SuS mit dem eigenen Projekt begonnen haben:

- Ich habe gesehen, dass du in deinem Projekt angedacht hast die Idee X umzusetzen. Hast du bereits eine konkrete Vorstellung, wie du das machen könntest?
- Du hast Scratch gerade erst kennengelernt, aber welche Arten von Projekten würdest du mit Scratch in Zukunft gern umsetzen wollen?
- Warum würdest du diese Art von Projekt machen wollen? Wie würdest du vorgehen, um so ein Projekt zu erstellen? Wo würdest du nach Hilfe suchen, wenn du in Scratch nicht weiter kommst?
- Bitte zeige mir dein konkretes Projekt an dem du arbeitest. Worum geht es in deinem Projekt? Warum hast du das Projekt ausgewählt? Wie soll dein Projekt am Ende aussehen, welche Funktionen soll es dann haben?
- Woher hast du deine Ideen dafür genommen? Was hat dich inspiriert dazu?
- Kannst du mir zeigen, was du bereits erreicht hast? (Fragen zu spezifischen Projektteilen stellen - zum Beispiel: Wie hast du es geschafft, dass die Bälle so umher fliegen?)

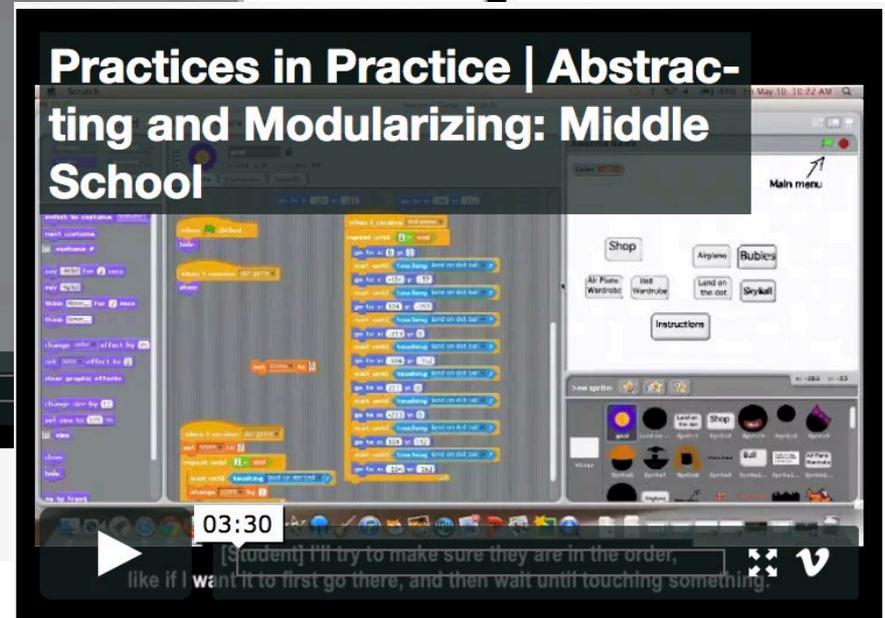
Artefakt-basierte Interviews – Videobeispiele



MIDDLE SCHOOL

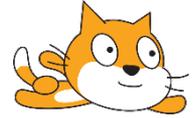


MIDDLE SCHOOL



MIDDLE SCHOOL

Entwicklertagebuch



Eliana E and Destiny O (

Name
04/10/2013
Date

Scratch Computational Thinking Journal

Abstracting and modularizing means
exploring connections
between the whole
and the parts.

Use the space below to write and draw about your practice of abstracting and modularizing while using Scratch.

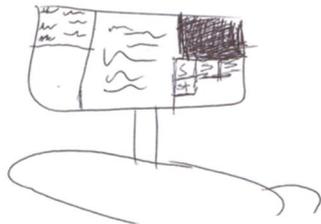
How did you decide what sprites were needed, and what each should do?

I decided what sprite I ~~needed~~ looked at my board and my story also my backgrounds



How did you organize the scripts of your sprites into meaningful, easily understandable stacks?

I looked at my background to see which sprite I need first and then I looked at my story.



KOCCO U

Name
04/10/2013
Date

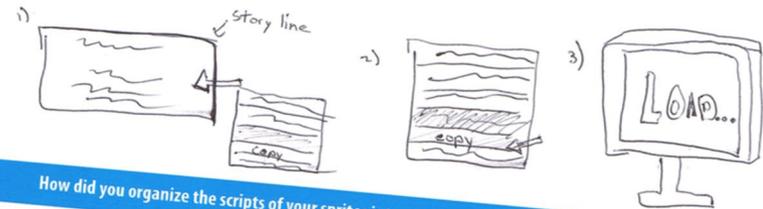
Scratch Computational Thinking Journal

Abstracting and modularizing means
exploring connections
between the whole
and the parts.

Use the space below to write and draw about your practice of abstracting and modularizing while using Scratch.

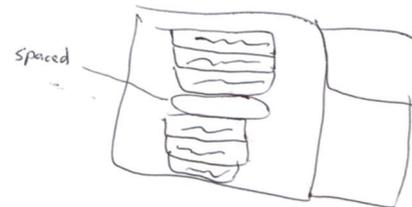
How did you decide what sprites were needed, and what each should do?

I decided by looking at my story lines. After that, it was like copy and paste.



How did you organize the scripts of your sprites into meaningful, easily understandable stacks?

I organized it neat by spacing out the lines.



Prüfung klassisch: Programme lesen und interpretieren



Tom hat ein Spiel mit Scratch gebaut, bei dem der Spieler einem Regentropfen ausweichen muss, der vom Himmel fällt. Nachdem der Regentropfen platschend am Boden aufgeschlagen ist, erscheint er immer wieder zufällig an einer anderen Stelle am Himmel und fällt erneut herunter. Welches der folgenden Programme hat Tom für den Regentropfen gebaut?

```
Wenn Flagge angeklickt
wiederhole fortlaufend
  wiederhole 10 mal
    gehe 10 er-Schritt
    spiele Klang Platsch
  warte 1 Sek.
```

```
Wenn Flagge angeklickt
wiederhole fortlaufend
  wiederhole bis y-Position < -180
    ändere y um -10
  spiele Klang Platsch
  warte 0.2 Sek.
  setze y auf 180
  setze x auf Zufallszahl von -200 bis 200
```

```
Wenn Flagge angeklickt
wiederhole fortlaufend
  ändere y um -180
  warte 1 Sek.
  spiele Klang Platsch
  setze x auf 0
```

```
Wenn Flagge angeklickt
wiederhole bis wird Rand berührt?
  ändere y um -10
warte 1 Sek.
spiele Klang Platsch
setze x auf Zufallszahl von -200 bis 200
```

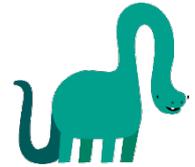
Beispiel eines Beurteilungsrasters



Kategorie	Niveau 1 (1 Punkt)	Niveau 2 (2 Punkte)	Niveau 3 (3 Punkte)
Planen	Die SchülerInnen erstellen einen Plan, der einige Details des Projekts enthält und/oder einige Ideen, wie dies in Scratch gemacht werden kann.	Die SchülerInnen erstellen einen Plan, der das Projekt beschreibt, und erwähnen einige Ideen, wie dies in Scratch gemacht werden kann.	Die SchülerInnen erstellen einen Plan, welcher das zu erstellende Projekt detailliert beschreibt. Der Schüler hat sich überlegt, wie er dies in Scratch erreichen kann, einschließlich spezifischer Blöcke und Logik, die verwendet werden können.
Entwickeln	Die SchülerInnen zeigen teilweise Kenntnisse von Scratch, indem sie versuchen, das Projekt wie in den Designspezifikationen beschrieben abzuschließen. Viele Funktionen funktionieren nicht oder fehlen. Die SchülerInnen erwähnen alle Änderungen am ursprünglichen Plan.	Die SchülerInnen demonstrieren ihr Wissen über Scratch, indem sie das Projekt erfolgreich abschließen und die meisten der gesteckten Ziele erreichen. Die SchülerInnen vergleichen das Endprodukt mit ihrem Plan und beschreiben eventuelle Änderungen am Plan.	Die SchülerInnen demonstrieren ausgefeilte Kenntnisse von Scratch. Sie schliessen das Projekt erfolgreich ab und erfüllen die gesteckten Ziele. Das Projekt ist kreativ & interessant für den Nutzer und alle Teile passen zu einem sinnvollen Ganzen. Das Programmierwerkzeug wird auf anspruchsvolle Weise eingesetzt. Der/die SchülerIn vergleicht das Endprodukt mit dem ursprünglichen Plan und begründet Änderungen am Plan.
Beschreiben und Beurteilen	Die SchülerInnen beschreiben nur das Endprodukt und/oder teilweise den Prozess der Erstellung.	Die SchülerInnen beschreiben den Entwicklungsprozess ihres Scratch-Projekts, erwähnt Herausforderungen und/oder Erfolge und nennt Ideen für weitere Verbesserungen.	Die SchülerInnen beschreiben den Entwicklungsprozess ihres Scratch-Projekts sorgfältig und vollständig, sie beschreiben Herausforderungen und Erfolge und diskutiert Ideen zur weiteren Verbesserung des Endprodukts.
Einstellung	Die SchülerInnen zeigen nur gelegentlich ein zufriedenstellendes Niveau in Bezug auf persönliches Engagement oder ihre Einstellung zum Arbeiten und/oder der Zusammenarbeit mit anderen.	Die SchülerInnen zeigen häufig ein zufriedenstellendes Niveau sowohl im persönlichen Engagement als auch konzentriert bei der Arbeit, aktiv in der Kooperation und mit Respekt gegenüber anderen.	Die SchülerInnen haben stets ein zufriedenstellendes Niveau bezüglich persönliches Engagement (Motivation, Selbständigkeit, allgemeine positive Einstellung) und sind konzentriert bei der Arbeit, arbeiten gut Zusammen und haben, Respekt vor anderen.

Weitere Ressourcen





AKTIVITÄTEN FÜR DEN UNTERRICHT

+ NEUE AKTIVITÄT ERSTELLEN

AUSWAHL SCHLIESSEN 1/1 | Q

FACH

ALLE

SCHULSTUFE

ALLE

SUCHEN

SCRATCH

SPRACHE

DEUTSCH



DE

CC

Chevalley Clara

Aktivität Scratch

Die Schüler werden lernen wie man das



DE

RE

Revey Estelle

Junior Scratch-Aktivität

In dieser Aktivität für den 7vklus 1 werden



DE

GS

Guntern Sophie

Aktivität Scratch, Schulweg



DE

JA

Justine Ambord

Aktivität Scratch

Bei dieser Aktivität werden die

<https://www.roteco.ch/de>



<https://mia.phsz.ch/Informatikdidaktik/ScratchBewertung>

<https://mia.phsz.ch/Informatikdidaktik/ScratchMaterial>

<https://www.roteco.ch/de>

<https://courseware.epfl.ch/courses/course-v1:EPFL+THYMIO-DE+2020/about>

<https://scratch.mit.edu/ideas> *Scratch Tutorials und Coding Cards*

<http://scratched.gse.harvard.edu/ct/defining.html>

<http://scratched.gse.harvard.edu/guide/curriculum.html>

<https://mia.phsz.ch/Informatikdidaktik/BuecherProgrammierenMitScratch>

Aktuellere Bücher sind auf der Kursseite aufgelistet:

<https://mia.phsz.ch/Primarstufe/ScratchFuerFortgeschritteneFebruarMaerz2021>