



# 4

## **Verstehen und Entwickeln von Code**

Planen und Entwickeln einer Reihe von Anweisungen an ein Computersystem, um ein bestimmtes Problem oder eine spezifische Aufgabe zu lösen.



Illustration: Daria Rüttimann

Kompetenzbereich

## Inhalts- und Medienentwicklung

Kompetenz

## Verstehen und Entwickeln von Code



Version 1.2  
Lizenz: Namensnennung - Weitergabe unter gleichen  
Bedingungen 4.0 International (CC BY-SA 4.0)



Hier geht es zur  
zentralen Downloadseite  
der Materialien:  
[bit.ly/dja-material](https://bit.ly/dja-material)

# Thematische Einführung

## digitale jugend arbeit

In Deutschland sprechen sich politische Institutionen und Bildungsinitiativen seit Jahren für eine stärkere Förderung informationstechnologischer Kompetenzen junger Menschen aus. Diese Fähigkeiten werden unter anderem als „Digitale Mündigkeit“ bezeichnet und sollen durch ein eigenes Unterrichtsfach vermittelt oder als Kompetenz in verschiedene Fächer integriert werden. Hierbei sollen junge Menschen Kompetenzen im Bereich des Programmierens erlernen, mit denen sie die komplexe digitale Welt verstehen, kritisch bewerten und kreativ mitgestalten können. Obwohl Kinder und Jugendliche immer früher mit digitaler Technologie konfrontiert werden, besteht ein großer Bedarf nach der Förderung dieser Kompetenzen nicht nur bei jungen Menschen. Aber worum geht es beim Programmieren überhaupt?

Programmieren ist eine Tätigkeit, bei der Programme geschrieben werden. Programme sind eine Ansammlung von Algorithmen; Algorithmen wiederum sind Anweisungen, wie bestimmte Aufgaben schrittweise gelöst werden. Algorithmen existierten in der Mathematik auch schon vor Computern – selbst die ersten Computerprogramme wurden vor diesen entwickelt. Zunächst wurden diese Algorithmen in „natürlicher Sprache“ formuliert, d. h. in den Sprachen, die wir als Menschen sprechen. Diese Sprachen sind jedoch vieldeutig und können keine logisch eindeutigen Anweisungen ausdrücken. Die Anweisungen in den Programmen müssen

daher für einen Computer übersetzt werden. Hier kommen die Programmiersprachen ins Spiel. Diese können die entwickelten Anweisungen unmissverständlich an einen Computer kommunizieren, sodass er versteht, was er tun soll.

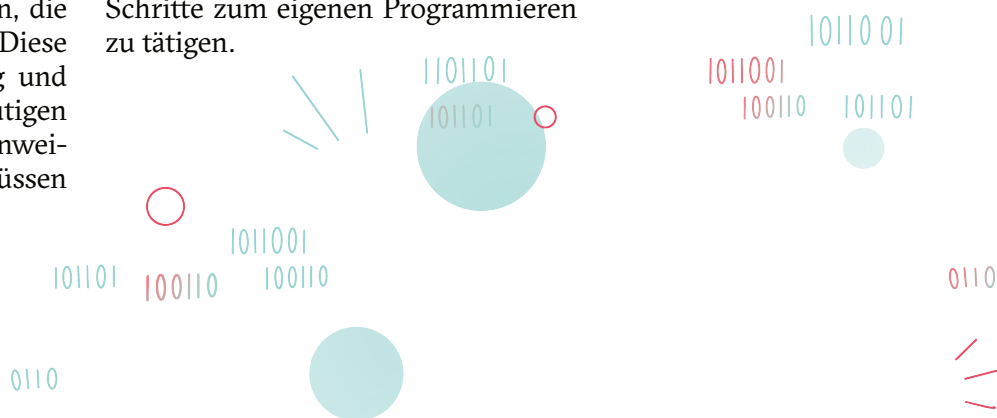
Programmieren bedeutet also nicht nur abgehackte Sätze in eine Tastatur einzugeben. Vielmehr sollen kreative Lösungsansätze für Probleme entwickelt werden, die es dann logisch zu strukturieren und schließlich für ein Gegenüber verständlich zu formulieren gilt. Aus dieser Perspektive betrachtet, ist Programmieren also der Art und Weise, wie wir unseren Problemen im Alltag begegnen, sehr ähnlich. Programmieren ermöglicht uns allerdings, die Hilfe von Computern in Anspruch zu nehmen und diese sogar selber zu erstellen.

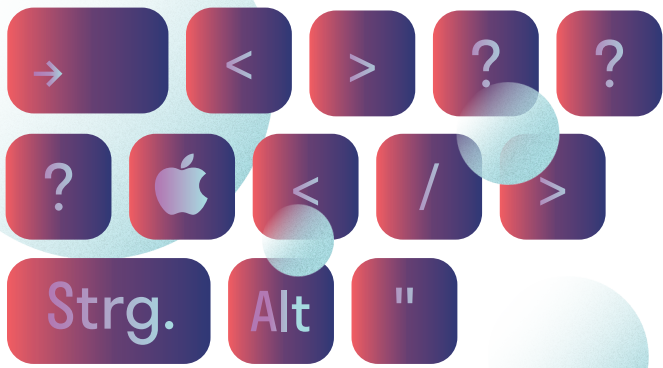
Dieses Modul nähert sich dem Verstehen und Entwickeln von Code an, indem es über Mythen aufklärt, welche das Lernen von Programmiersprachen, das Image von Programmierer:innen und die Anwendungspraxis des Programmierens betreffen. Auf diese Weise wird ein Einblick in die Logik, Kultur und Geschichte des Programmierens gewährt. Anschließend gibt es die Möglichkeit, spielerisch und einsteigerfreundlich die ersten Schritte zum eigenen Programmieren zu tätigen.

### Inhalt

### Seite

<b>Aufgabe 1</b>	s.49
Arbeitsmaterial 1	s.51
Arbeitsmaterial 2	s.51
Arbeitsmaterial 3	s.52
Arbeitsmaterial 4	s.52
Arbeitsmaterial 5	s.53
Arbeitsmaterial 6	s.53
Arbeitsmaterial 7	s.54
Arbeitsmaterial 8	s.54
Arbeitsmaterial 9	s.55
Arbeitsmaterial 10	s.55
Arbeitsmaterial 11	s.56
Arbeitsmaterial 12	s.56
<b>Aufgabe 2</b>	s.57





# digitale jugend arbeit

Kompetenzbereich  
Inhalts- und  
Medienentwicklung

Kompetenz  
Verstehen und  
Entwickeln von Code

Stufe  
Einstieg

Methode  
Gallery-Walk

Ausstattung  
Bildungsmaterialien +  
Ausgedruckte Arbeits-  
materialien

Dauer  
90 Minuten

## Programmiermythen defragmentiert

@Trainer:innen · Moderationsbriefing · 3.4

Diese Übung soll das Aufgabenfeld des Code-Schreibens und die dahinterstehenden Programmierer:innen entmystifizieren und gleichzeitig ein exploratives Eintauchen in die Thematik ermöglichen.

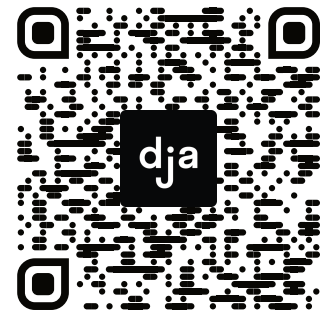
### Ablauf

Zunächst haben die Teilnehmer:innen Zeit, sich mit unserer Auswahl von zwölf Mythen über das Programmieren auseinanderzusetzen. Dabei sind die einzelnen Mythen im Raum verteilt und die Teilnehmer:innen können in ihrem eigenen Tempo und in individueller Reihenfolge zwischen den Stationen wechseln. Zuerst sollen sie sich die Behauptung durchlesen und kurz ihre eigenen Gedanken dazu sammeln und reflektieren. Danach besuchen sie die beigefügten Quellen, welche die jeweiligen Vorurteile in Frage stellen und den Teilnehmer:innen weitere Denkipulse geben.

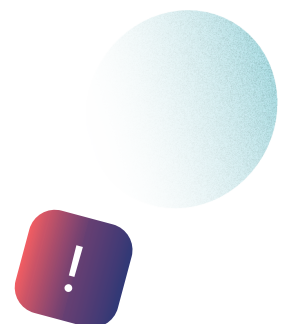
Zum Schluss findet im Plenum eine gemeinsame Reflexion statt. Dabei sollten die aus den Quellen gewonnene Eindrücke gesammelt und darauf aufbauend alle Mythen kurz besprochen werden. Schwerpunkte auf einzelnen Themen dieser Abschlussdiskussion können je nach Bedarf und Fokus der Gruppe gewählt werden.

### Hinweis zur Moderation

- Der Gallerywalk dient dazu, den Teilnehmer:innen ihrem jeweiligen Wissensstand entsprechende Denkipulse zur Verfügung zu stellen. Sie müssen sich nicht mit allen zwölf Mythen auseinandersetzen oder alle verlinkten Artikel vollständig lesen. Die Teilnehmer:innen sollten daher darauf hingewiesen werden, dass sie sich auf diejenigen Mythen konzentrieren können, welche sie besonders interessieren.



Hier geht es zur zentralen  
Downloadseite der Materialien:  
»[bit.ly/dja-material](https://bit.ly/dja-material)«





Kompetenzbereich  
Inhalts- und  
Medienentwicklung

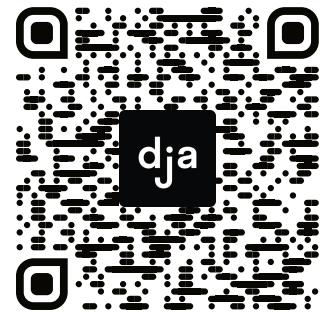
Kompetenz  
Verstehen und  
Entwickeln von Code

Stufe  
Einstieg

Methode  
Gallery-Walk

Ausstattung  
Bildungsmaterialien +  
Ausgedruckte Arbeits-  
materialien

Dauer  
90 Minuten



Hier geht es zur zentralen  
Downloadseite der Materialien:  
>>[bit.ly/dja-material](https://bit.ly/dja-material)<<

## Übersicht über die Programmier-Mythen

### Lernen von Programmiersprachen

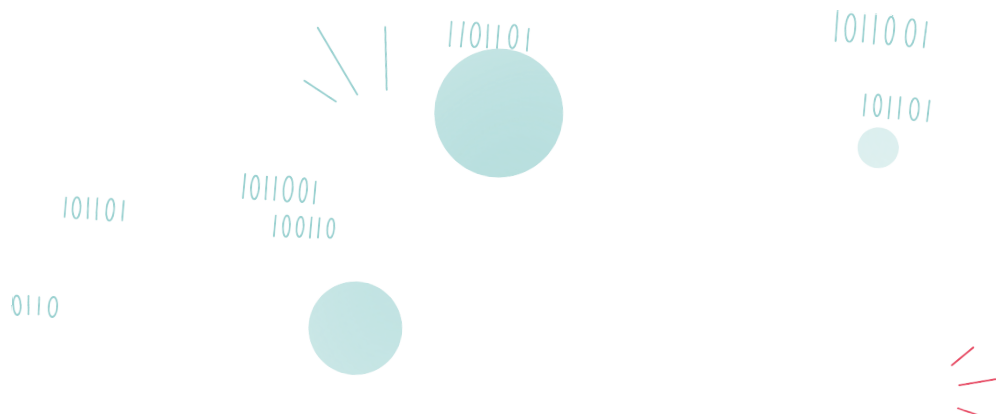
- „Programmieren zu lernen ist eine anstrengende und zeitaufwändige Tätigkeit.“
- „Programmieren ist eng mit Mathematik verknüpft und erfordert einen hohen Kenntnisstand dieser.“
- „Es gibt eine große Vielfalt von Programmiersprachen und davon ist jede so anders, dass ich immer wieder von vorne anfangen muss.“
- „Das Programmieren wird in naher Zukunft durch künstliche Intelligenzen betrieben. Das Erlernen jener Fähigkeiten ist daher überflüssig.“

### Image von Programmierer:innen

- „Programmieren ist eine Einzelbeschäftigung ohne großen menschlichen Kontakt.“
- „Programmieren ist eine Domäne von weißen Männern.“
- „Alle Programmierer:innen sind ausgebildete Erwachsene.“
- „Alle Hacker:innen sind Programmierer:innen, die Codes für illegale und kriminelle Aktivitäten ausarbeiten.“

### Anwendung

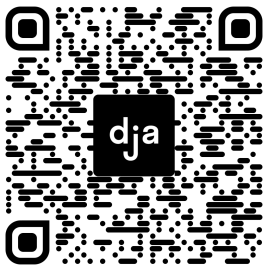
- „Programmieren erfordert eine teure Grundausstattung, wodurch der Einstieg recht kostspielig ist.“
- „Programmieren findet nur Anwendung in digitalen Räumen. Wer Basteln mag, wird lieber Ingenieur:in.“
- „Programme werden nur in schwer nachvollziehbaren Quellcodes geschrieben.“
- „Programmieren entstand erst mit der Erfindung des Computers.“





# Mythos N<sup>o</sup> 1: „Programmieren zu lernen ist eine anstrengende und zeitaufwändige Tätigkeit.“

Lernen von Programmiersprachen



„Programmieren ist wie Küssen: Man kann darüber reden, man kann es beschreiben, aber man weiß erst was es bedeutet, wenn man es getan hat. Vielen wird es so gehen: Ich würde gerne Programmieren lernen, aber ich glaube, ich kann es nicht! Für diese Menschen hat Cecily Carver folgende Tipps.“

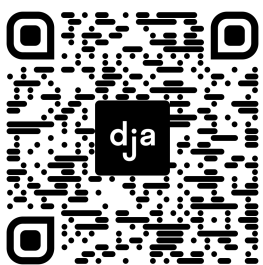
»**Programmieren ist kein Hexenwerk: Lass dich nicht verunsichern**« *t3n* (2015)

[t3n.de/news/programmieren-lernen-588904/](https://t3n.de/news/programmieren-lernen-588904/)



# Mythos N<sup>o</sup> 2: „Programmieren ist eng mit Mathematik verknüpft und erfordert einen hohen Kenntnisstand dieser.“

Lernen von Programmiersprachen



„Die meisten Probleme, die Programmierer zu lösen haben, sind effiziente Datenverwaltungen. Hierfür müssen eher Datenstrukturen bekannt sein, als höhere Mathematik.“

»**Wie viel Mathematik brauche ich zum Programmieren?**«

*Progen.org*

[progen.org/doku.php?id=start:math](https://progen.org/doku.php?id=start:math)



# Mythos N<sup>o</sup> 3: „Es gibt eine große Vielfalt von Programmiersprachen und davon ist jede so anders, dass ich immer wieder von vorne anfangen muss.“

## Lernen von Programmiersprachen



»Die richtigen Programmiersprachen für Anfänger.«

*It-talents* Landwehr, Jesko (2018)

[it-talents.de/blog/it-talents/so-gelingt-der-einstieg-in-die-programmierung](https://it-talents.de/blog/it-talents/so-gelingt-der-einstieg-in-die-programmierung)



»Überblick über verschiedene Programmiersprachen« *BMU Verlag*

[bmu-verlag.de/uberblick-uber-verschiedene-programmiersprachen/](https://bmu-verlag.de/uberblick-uber-verschiedene-programmiersprachen/)



# Mythos N<sup>o</sup> 4: „Programmieren wird in naher Zukunft durch künstliche Intelligenzen betrieben. Das Erlernen jener Fähigkeiten ist daher überflüssig.“

## Lernen von Programmiersprachen



»Wird künstliche Intelligenz Programmierer ersetzen?«

*SALESmanago Blog* (2019)

[blog.salesmanago.de/marketing-automation/wird-kuenstliche-intelligenz-programmierer-ersetzen/](https://blog.salesmanago.de/marketing-automation/wird-kuenstliche-intelligenz-programmierer-ersetzen/)



»Künstliche Intelligenz Vs. Mensch – Sind Wir Ersetzbar?«

*MoreThanDigital*

Wengeler, Mathias (2019)

[morethandigital.info/kuenstliche-intelligenz-vs-mensch-sind-wir-ersetzbar/](https://morethandigital.info/kuenstliche-intelligenz-vs-mensch-sind-wir-ersetzbar/)



# Mythos N° 5: „Programmieren ist eine Einzelbeschäftigung ohne großen menschlichen Kontakt.“

Image von Programmierer:innen



»*Programmieren ist sozialer geworden*« *golem.de*  
[golem.de/news/programmieren-programming-motherfucker-do-you-speak-it-1405-106106-2.html](https://golem.de/news/programmieren-programming-motherfucker-do-you-speak-it-1405-106106-2.html)



»**Warum ein zweiter Blick auf den Nutzen von Pair Programming lohnt – Gemeinsam weniger einsam programmiert**«  
*entwickler.de* Trotzki, Thomas & Thomas Schissler (2018)  
[entwickler.de/online/windowsdeveloper/pair-programming-579843832.html](https://entwickler.de/online/windowsdeveloper/pair-programming-579843832.html)



# Mythos N° 6: „Programmieren ist eine Domäne von weißen Männern.“

Image von Programmierer:innen



„Eine wichtige Informatik-Erfindung nennen, die von einer Frau stammt? Da kommt selten eine Antwort. Und doch belegen die Gewinnerinnen zahlreicher Auszeichnungen, wie etwa des berühmten Turing-Awards: Zentrale Erfindungen in der Informatik stammen von Frauen. Wir stellen sechs von ihnen vor.“

»**Heldinnen der Informatik**« *SFR 3* Buchmann, Peter (2014)  
[srf.ch/radio-srf-3/digital/heldinnen-der-informatik](https://srf.ch/radio-srf-3/digital/heldinnen-der-informatik)  
[9 Black Women in Data Science to Know](#)





# Mythos N° 7: „Alle Programmierer:innen sind ausgebildete Erwachsene.“

## Image von Programmierer:innen



Über Jugend hackt *Jugend hackt*  
[jugendhackt.org/ueber/](https://jugendhackt.org/ueber/)



»Programmieren lernen: Die besten Produkte für Kinder und Jugendliche«  
*Edition F Knodel, Philipp (2016)*  
[editionf.com/Programmieren-lernen-Tolle-Produkte-fuer-Kinder-und-Jugendliche/](https://editionf.com/Programmieren-lernen-Tolle-Produkte-fuer-Kinder-und-Jugendliche/)



# Mythos N° 8: „Alle Hacker:innen sind Programmierer:innen, die Codes für illegale und kriminelle Aktivitäten ausarbeiten.“

## Image von Programmierer:innen



»Was ist ein Hacker?« *Vpnoverview Janssen, David (2020)*  
[vpnoverview.com/de/sicheres-internet/cyberkriminalitaet/was-ist-ein-hacker/](https://vpnoverview.com/de/sicheres-internet/cyberkriminalitaet/was-ist-ein-hacker/)

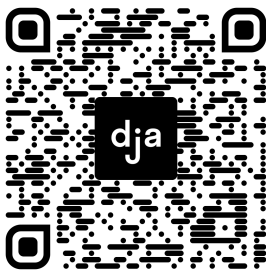


*Hackerethik* Chaos Coputer Club  
[ccc.de/de/hackerethik](https://ccc.de/de/hackerethik)



# Mythos N° 9: „Programmieren erfordert eine teure Grundausstattung, wodurch der Einstieg recht kostspielig ist.“

## Anwendung



„Der *Raspberry Pi* ist ein Minicomputer auf einer nur Scheckkarten-großen Platine, der sich vielfältig erweitern lässt. Er wurde ursprünglich zu Lern- und Demonstrationszwecken entwickelt, eignet sich aber auch für die Realisierung verschiedener privater oder professioneller Anwendungen.“

»**Definition Was ist Raspberry Pi?**« *BigData-Insider*

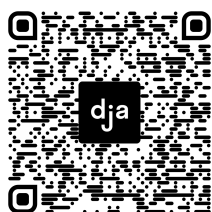
Luber, Stefan & Nico Litzel (2017)

[bigdata-insider.de/was-ist-raspberry-pi-a-670954/](http://bigdata-insider.de/was-ist-raspberry-pi-a-670954/)



# Mythos N° 10: „Programmieren findet nur Anwendung in digitalen Räumen. Wer Basteln mag, wird lieber Ingenieur:in.“

## Anwendung



»Mit Robotern das Programmieren lernen«

*Bildung.digital*  
[bildung.digital/artikel/  
mit-robotern-das-  
programmieren-lernen](http://bildung.digital/artikel/mit-robotern-das-programmieren-lernen)



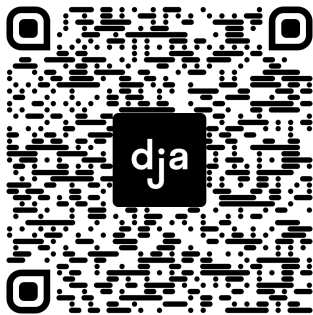
»Programmieren und Basteln mit dem Einplatinencomputer«

Rheinwerk Verlag GmbH (2021)  
[rheinwerk-verlag.de/zeit-fuer-  
weiterbildung/programmieren-  
lernen-fuer-kinder/calliope-mini-  
projekte/](http://rheinwerk-verlag.de/zeit-fuer-weiterbildung/programmieren-lernen-fuer-kinder/calliope-mini-projekte/)



# Mythos N° 11: „Programme werden nur in schwer nachvollziehbaren Quellcodes geschrieben.“

## Anwendung



„Guter Code oder schlechter Code – Definitionen dafür, woran die Code-Qualität erkennbar ist, gibt es viele. Die Lesbarkeit von Code stellt aber häufig ein wichtiges Kriterium dar.“

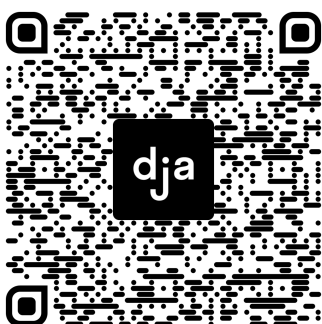
»**Besserer Code mit fünf Tipps. Der Code Codex: Ein Knigge für guten Code?**« *entwickler.de* Klose, Ann-Cathrin (2017)

[entwickler.de/online/development/code-codex-knigge-guter-code-579782243.html](https://entwickler.de/online/development/code-codex-knigge-guter-code-579782243.html)



# Mythos N° 12: „Programmieren entstand erst mit der Erfindung des Computers.“

## Anwendung



„Ganz in Vergessenheit geriet sie nie, war sie doch die Tochter eines berühmten Mannes. Aber es dauerte nach ihrem Tod über 100 Jahre, bis ihre eigenen Leistungen erkannt und Ada Lovelace endlich ihretwegen bekannt wurde.“

»**Ada Lovelace – Gegen alle Widerstände: Ouvertüre des Computer-Zeitalters**«

*Deutsches Patent- und Markenamt* Dr. Potthast, Björn, Jan (2021)  
[dpma.de/dpma/veroeffentlichungen/aktuelles/patentefrauen/adalovelace/index.html](https://dpma.de/dpma/veroeffentlichungen/aktuelles/patentefrauen/adalovelace/index.html)

# Scratch kennenlernen

@Trainer:innen · Moderationsbriefing · 3.4

Die Teilnehmer:innen sollen in die Praxis des Programmierens eingeführt werden. Dabei lernen sie den Aufbau eines Codes kennen und bekommen grundlegende Befehle einer Programmiersprache bausteinartig an die Hand gegeben.

## Ablauf

Für diese Aufgabe wird die Programmiersprache *Scratch* von der gleichnamigen Webseite »[scratch.mit.edu](https://scratch.mit.edu)« als Methode genutzt. Das Tool ist kostenlos, in über 60 Sprachen verfügbar und ermöglicht einen spielerischen und niedrighschwelligeren Zugang zum Code-Schreiben.

Für den Einstieg können die Teilnehmer:innen sich schrittweise durch die Anleitungen auf der Webseite klicken. So bietet *Scratch* unter anderem ein „Erste Schritte“-Tutorial an, welches Besucher:innen den Seitenaufbau erklärt. Danach können sie sich durch weitere Tutorials klicken und beispielsweise versuchen, das bekannte Pong-Spiel nachzubauen. Dabei könnte es sinnvoll sein, die Teilnehmer:innen zunächst alleine die Plattform entdecken zu lassen. Später können sie sich von den „Projekten für Einsteiger“ inspirieren lassen und versuchen, diese zu remixen – entsprechende Ideen finden sie in der Projektbeschreibung auf Englisch. Vielleicht möchten die Teilnehmer:innen aber auch ein eigenes Projekt angehen. Hierbei ist ein Austausch sinnvoll, um sich von den anderen inspirieren zu lassen und für ein gegenseitiges Staunen über die eigenen Leistungen und die Anwendungen anderer Nutzer:innen zu sorgen.

Am Ende können die Teilnehmer:innen in Kleingruppen oder im Plenum ihre Erfahrung schildern und sich gegenseitig ihre Projekte vorstellen.

## Hinweise zur Moderation

- Der Ablauf und die Arbeitsanweisungen sind in dieser Aufgabe relativ offen angelegt, damit die Teilnehmer:innen die Möglichkeiten von *Scratch* selbstständig entdecken können. Für den Fall, dass die Teilnehmer:innen davon überfordert sind, bietet es sich jedoch an, einige konkreten Arbeitsanweisungen parat zu haben, auf welche im Zweifelsfall zurückgegriffen werden kann.
- *Scratch* ist für 8-16-Jährige konzipiert und daher ein spannendes Tool, welches Jugendarbeiter:innen auch an junge Menschen weitertragen können.
- Für diese Aufgabe lässt sich sowohl die Zeit als auch die Gruppengröße variabel einteilen. Die Webseite bietet viele Möglichkeiten, sich auch weit über den Zeitraum eines Seminars hinaus den dort angelegten Projekten zu widmen.
- Die Webseite ist für Desktop Computer ausgelegt, funktioniert aber auch auf mobilen Geräten. Beispielsweise lässt sie sich problemlos mit einem Tablet und angeschlossener Tastatur bedienen. Für *Android*-Tablets oder *iPads* gibt es zusätzlich die App *Scratch Jr.*, die sich an 5-7-Jährige richtet, ihrem Desktop-Äquivalent allerdings nur in ein paar verfügbaren Funktionen nachsteht.
- Viele Anleitungen für die Tutorial-Projekte findest du von *Scratch* selbst als [PDF in Form von Karten](#). Je nach Digitalfokus des Seminars kann es sinnvoll sein, diese auszudrucken.

## digitale jugend arbeit

Kompetenzbereich

Inhalts- und  
Medienentwicklung

Kompetenz

Verstehen und  
Entwickeln von Code

Stufe

Vertiefung

Methode

Toolbasiertes Lernen

Ausstattung

Bildungsmaterialien +  
Endgeräte mit  
Tastatur empfohlen

Dauer

90+ Minuten



Hier geht es zur zentralen  
Downloadseite der Materialien:  
»[bit.ly/dja-material](https://bit.ly/dja-material)«







Grumpy Cat

ONE DOES NOT SIMPLY



~~WALK INTO MORDOR~~  
RESIST BACON

One Does Not Simply Meme



Spongebob rainbow Meme



Drake Hotline Bling Meme



Bernie Sanders mittens Meme



Success Kid