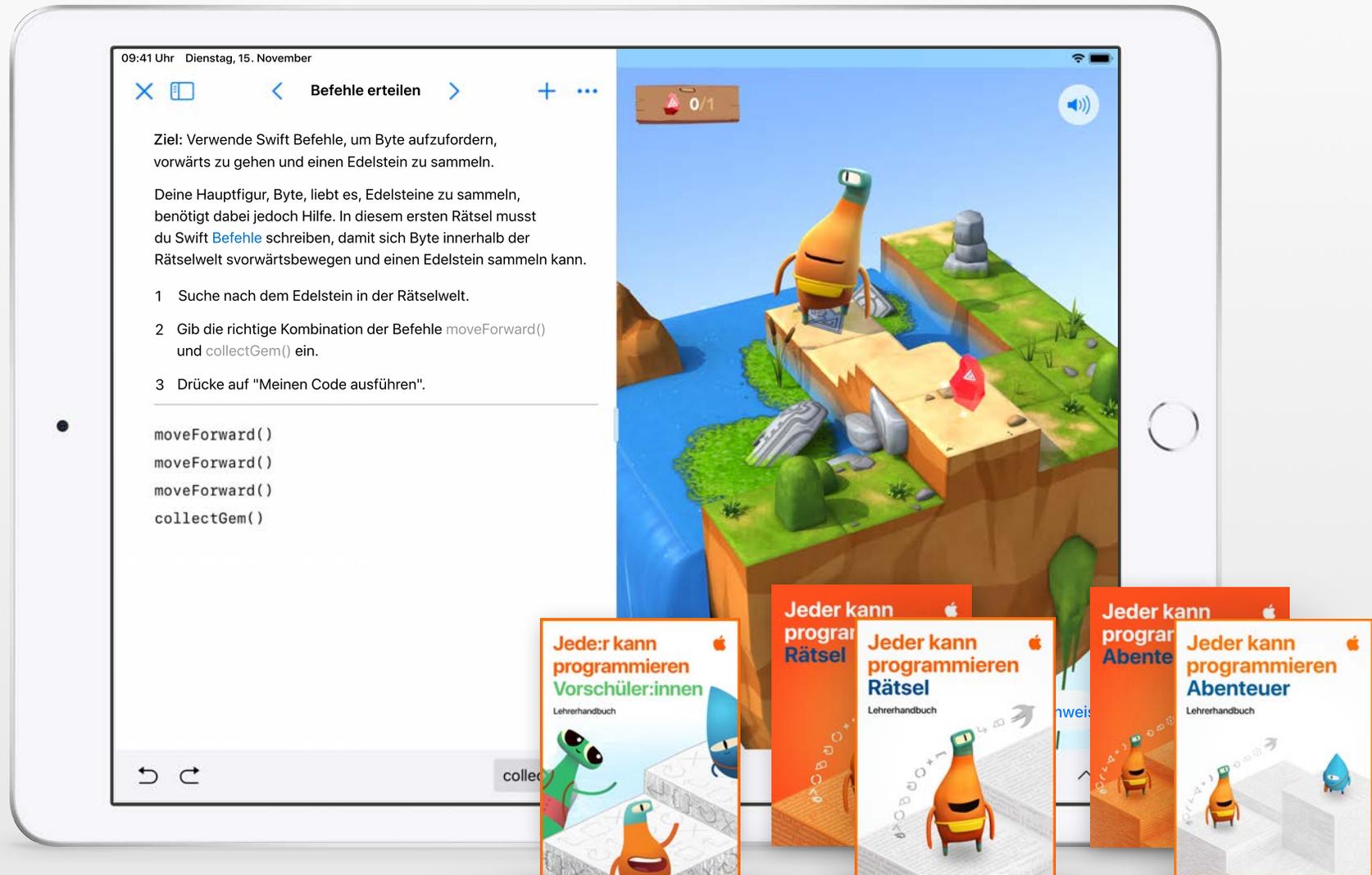


Apple Jede:r kann programmieren

Lehrplanführer



09:41 Uhr Dienstag, 15. November

Befehle erteilen

Ziel: Verwende Swift Befehle, um Byte aufzufordern, vorwärts zu gehen und einen Edelstein zu sammeln.

Deine Hauptfigur, Byte, liebt es, Edelsteine zu sammeln, benötigt dabei jedoch Hilfe. In diesem ersten Rätsel musst du Swift **Befehle** schreiben, damit sich Byte innerhalb der Rätselwelt svorwärtsbewegen und einen Edelstein sammeln kann.

- 1 Suche nach dem Edelstein in der Rätselwelt.
- 2 Gib die richtige Kombination der Befehle `moveForward()` und `collectGem()` ein.
- 3 Drücke auf "Meinen Code ausführen".

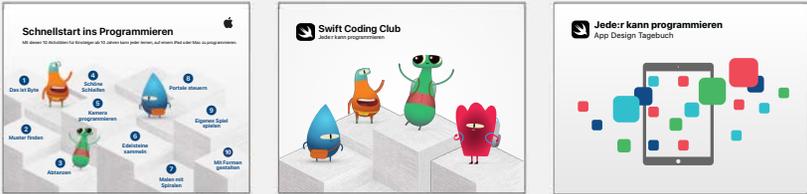
```
moveForward()  
moveForward()  
moveForward()  
collectGem()
```

collec

The image shows a tablet displaying a lesson plan for the game 'Jede:r kann programmieren'. The screen is divided into two main sections. On the left, there is a text-based instruction panel with a title 'Befehle erteilen' (Issuing Commands). It includes a goal, a description of the character 'Byte', and a three-step task list. Below the list is a code editor with four lines of Swift code: 'moveForward()', 'moveForward()', 'moveForward()', and 'collectGem()'. On the right, there is a 3D game environment showing a character named 'Byte' on a platform, with a red gem nearby. The game interface includes a score '0/1' and a speaker icon. In the foreground, four book covers are displayed, each with the title 'Jede:r kann programmieren' and a subtitle: 'Vorschüler:innen', 'Rätsel', 'Abenteuer', and 'Abenteuer'. The covers feature the game's character and various programming-related icons.

Programmierlehrplan

Ob Schüler:innen neu beim Programmieren sind oder bereits ihre ersten Apps entwickeln: Apple bietet Programme zur Unterstützung des Lehrens und Lernens mit Swift, der Programmiersprache, die von professionellen Entwickler:innen verwendet wird, um einige der leistungsstärksten Apps der Welt zu erstellen.

GRUNDSCHULE	SEKUNDARSTUFE	OBERSTUFE	HOCHSCHULE
<h2>🍏 Jede:r kann programmieren</h2>		<h2>🍏 In Swift entwickeln</h2>	
 <p>16 Stunden 45 Stunden 45 Stunden Nur iPad</p>		 <p>180 Stunden 180 Stunden 180 Stunden 180 Stunden</p>	
			
<p>Swift Playgrounds auf iPad und Mac   </p>		<p>Xcode auf Mac  </p>	

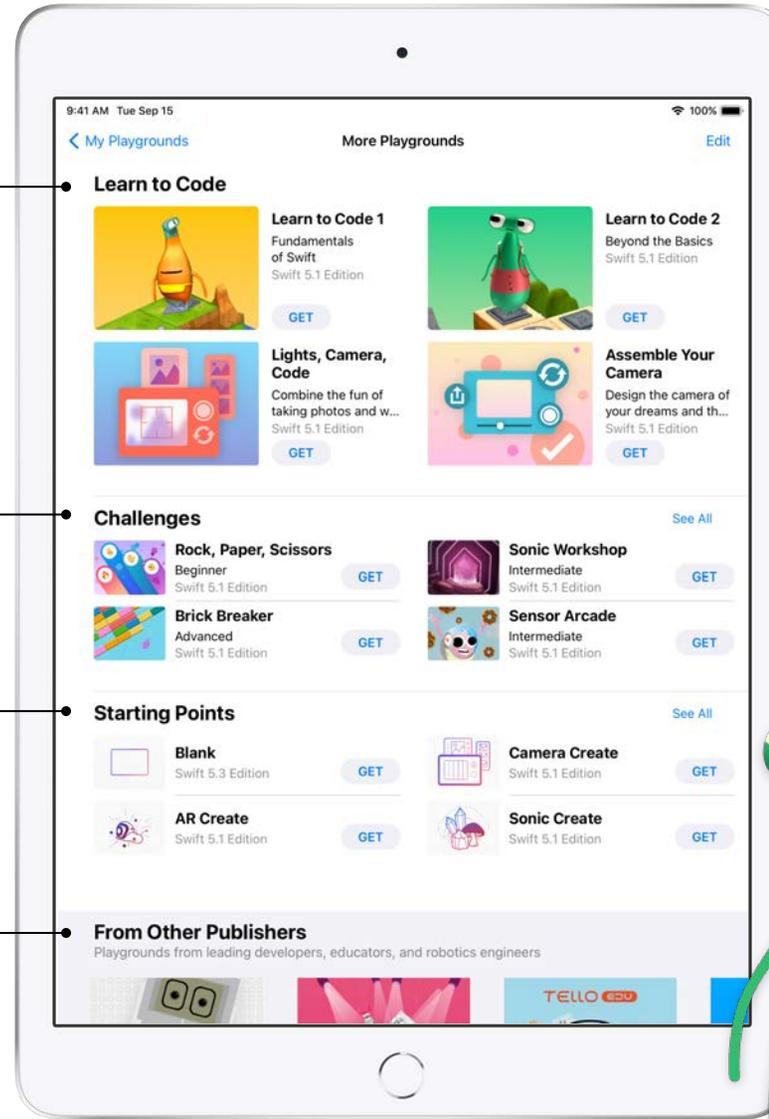
Umfang von Swift Playgrounds

Programmieren lernen. Die Schüler:innen lernen die Grundlagen von Swift, indem sie Charaktere durch eine 3D-Welt führen. Dann wenden sie ihr Wissen über Swift an und erstellen eine Kamera, einen Fotoeditor und eine Kamera-App von Grund auf neu.

Herausforderungen. Die Schüler:innen entdecken ihre Interessen oder probieren etwas Neues aus. In projektbasierten Playgrounds erstellen sie lustige Kreationen und bauen dabei ihre Programmierfähigkeiten aus.

Ausgangspunkte. Die Schüler:innen nutzen das Gelernte, um etwas ganz Neues zu erschaffen und einfach mit Freunden, der Familie oder der ganzen Welt zu teilen.

Von anderen Publishern. Die Schüler:innen können Kanäle entdecken, die lustige neue Herausforderungen von führenden Entwicklern und Publishern enthalten. Sie können ihre Favoriten abonnieren und die neuesten Kreationen direkt in der App erhalten.



Swift Playgrounds erkunden

Echter Swift Code. Im Zentrum von Swift Playgrounds steht dieselbe Swift Programmiersprache, mit der heute viele der führenden Apps im App Store geschrieben sind.

Interaktive Umgebung. Während der Code auf der linken Seite des Bildschirms erstellt wird, lassen sich auf der rechten Seite durch einfaches Tippen sofort die Ergebnisse anzeigen.

Zum Bearbeiten berühren. Komplexe Strukturen, die anderen Code umgeben, z. B. Schleifen und Funktionsdefinitionen, können einfach vor und hinter bestehenden Code bewegt werden. Dazu muss nur das Schlagwort (z. B. „for“) berührt werden, um die Bedienelemente zum Bewegen anzuzeigen.

Kurzbefehlleiste. Am unteren Rand des Bildschirms werden QuickType Vorschläge angezeigt, sodass die Schüler:innen benötigten Code durch einfaches Tippen auf die Kurzbefehlleiste eingegeben können.

Integriertes Glossar. Die Schüler:innen können zum besseren Verständnis die Definitionen von bestimmten Begriffen nachschlagen.

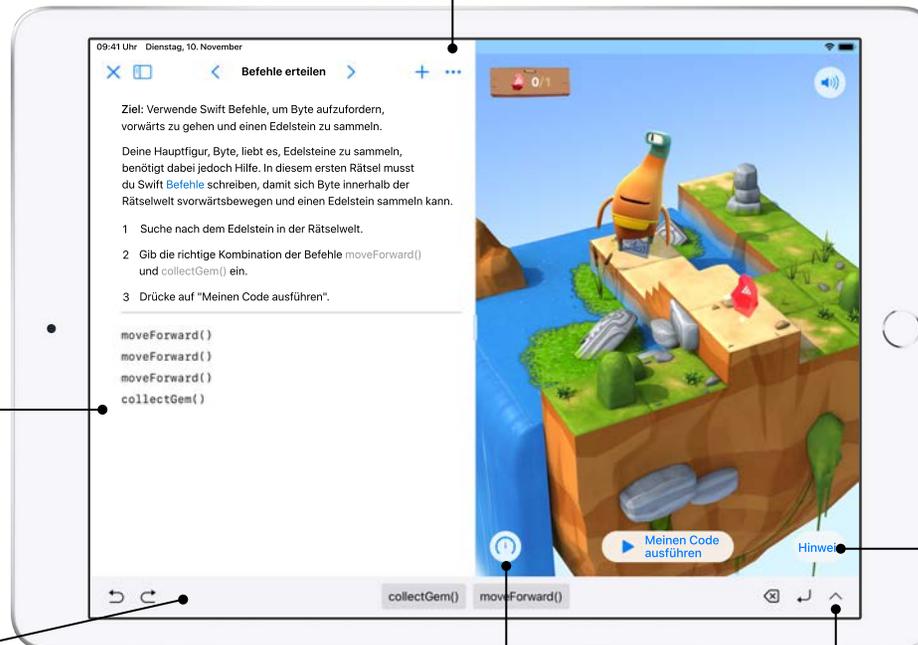
Aufnehmen und teilen. Die Schüler:innen können ihre Aktivitäten auf dem Bildschirm als Beleg für ihre Arbeit aufnehmen.

Spannende Animationen. Jeder Abschnitt beginnt mit einer spannenden Animation, in der die Programmierkonzepte auf das reale Leben übertragen werden, um den Schüler:innen das Verständnis zu erleichtern.

Bedienungshilfen. Swift Playgrounds wurde mit Blick auf Bedienungshilfen konzipiert. Es nutzt die vielen iPadOS Bedienungshilfen wie Schaltersteuerung und VoiceOver und bietet zusätzlich Sprachkommentare, die die Handlungen der Charaktere beschreiben, während sie von den Schüler:innen mithilfe von Code gesteuert werden.

Nützliche Hinweise. Die Schüler:innen können sich jederzeit helfen lassen, falls sie nicht mehr weiterwissen. In vielen Fällen werden die Hinweise mit der Eingabe von Code dynamisch geändert.

Onscreen-Tastatur. Eine speziell für Swift konzipierte Tastatur ermöglicht die schnelle Eingabe von Zahlen und Symbolen, die in Swift häufig verwendet werden.



Code überprüfen. Code kann schneller oder langsamer ausgeführt oder schrittweise abgearbeitet werden, um die Codezeilen während ihrer Ausführung hervorzuheben. Dadurch können die Schüler:innen einfacher die Bereiche erkennen, in denen Fehler auftreten könnten.

Jede:r kann programmieren – Umfang und zeitlicher Ablauf



Jede:r kann programmieren – Vorschüler:innen

Dieser Leitfaden wurde für Trainer:innen der Klassenstufe 3 entwickelt und enthält fünf Module: Befehle, Funktionen, Schleifen, Variablen und App-Design. Jedes Modul enthält Lektionen, die den Lernenden helfen, durch Wissenschaft, Kunst, Musik u.v.m. neue Programmierkonzepte zu erkunden. Die Lernenden teilen persönliche Erfahrungen und Ideen, während sie Programmierthemen erkunden. Durch praktische Aktivitäten und Herausforderungen entdecken sie, wie Code funktioniert, und wenden ihre neuen Fähigkeiten an, indem sie Code in der App „Swift Playgrounds“ erstellen. Im Modul „App-Design“ beginnen die Lernenden damit, Design-Thinking-Fähigkeiten in einem abschließenden App-Design-Projekt zu entwickeln. [Jede:r kann programmieren – Vorschüler:innen herunterladen >](#)

Befehle (3 Stunden). Die Schüler:innen verbinden alltägliche Beispiele von exekutiven Funktionen, Sprache und Bewegung mit Code. Dieses Modul konzentriert sich auf die Beschreibung von schrittweisen Anweisungen, das Sortieren von Schritten in der richtigen Reihenfolge und das Testen und Debuggen von Befehlen im Code.

Funktionen (3 Stunden). Über Kunst, Gesang sowie soziales und emotionales Lernen entdecken die Schüler:innen Funktionen. Dieses Modul zielt darauf ab, große Probleme in kleine Schritte zu zerlegen, Funktionen zur Lösung eines Problems zu erstellen und Funktionen zu benennen.

Schleifen (3 Stunden). Kunst, Sport, Musik, und wieder von vorne! Zu den Programmierthemen gehören das Identifizieren von Schleifen und das Wiederholen von Befehlsfolgen, um Rätsel und Aufgaben zu lösen.

Variablen (3 Stunden). Wissenschaft, Sprache und gemeinschaftsbildende Aktivitäten werden als Grundlage für das Erlernen von Variablen verwendet. Die Schüler:innen verknüpfen einen Variablennamen mit einem bestimmten Wert, ändern den Wert einer Variablen und verwenden verschiedene Variablentypen.

App-Design (mind. 1 Stunde). Die Schüler:innen werden inspiriert, mit einer Reihe lustiger App-Design-Aktivitäten Probleme in der Schule oder der Gesellschaft zu lösen, die sie persönlich betreffen. Sie nutzen Design Thinking, Kreativität und Empathie, um eine App-Idee zu entwickeln.





Jede:r kann programmieren – Rätsel

Für Schüler:innen ab Klassenstufe 4 liefert *Jede:r kann programmieren* einen Leitfaden für Swift Playgrounds, der Problemlösungsstrategien sowie die Fähigkeit zum kritischen Denken in mehr als 45 Stunden mit flexiblen Aktivitäten stärkt. Jedes Kapitel hilft den Schüler:innen, auf vorhandenem Wissen aufzubauen, mit neuen Programmierkonzepten zu experimentieren, ihre Kenntnisse anzuwenden und kreativ zu kommunizieren, wie Programmierung ihr Leben beeinflusst. [Jede:r kann programmieren – Rätsel herunterladen >](#)

Kapitel 1: Befehle (3 Stunden). Die Schüler:innen erfahren die Wichtigkeit eindeutiger, präziser Befehle. Sie gliedern Code in Sequenzen auf, um ein bestimmtes Ziel zu erreichen, und untersuchen die Verwendung von Befehlen in alltäglichen digitalen Technologien.

Kapitel 2: Funktionen (3 Stunden). Die Schüler:innen untersuchen die Leistungsfähigkeit von Funktionen, indem sie Befehle zu einer Definition gruppieren, die sie immer wieder verwenden können. Sie programmieren eine Funktion, um eine Tanzroutine für Roboter zu choreografieren, und denken darüber nach, welche Arten von Funktionen in den alltäglichen digitalen Technologien angewendet werden könnten.

Kapitel 3: for-Schleifen (3,5 Stunden). Die Schüler:innen erkennen Muster in der Welt um sie herum und in ihrem Code und lernen, wie sie mithilfe von Schleifen effizienteren Code schreiben können.

Kapitel 4: Variablen (4 Stunden). Die Schüler:innen lernen, wie Computer Informationen speichern, und erkunden das Programmieren mit Variablen, um Daten nachzuverfolgen und flexible Programme zu erstellen. Sie untersuchen, wie sich Änderungen an den Werten von Variablen auf die Ausgaben von Programmen auswirken.

Kapitel 5: Bedingte Anweisungen (4 Stunden). Die Schüler:innen erkunden, wie boolesche Logik uns in unserem alltäglichen Leben und bei der Programmierung hilft, Entscheidungen zu treffen. Sie üben das Schreiben von bedingten Anweisungen, um sich ändernde Bedingungen zu antizipieren.

Kapitel 6: Typen und Initialisierung (5 Stunden). Die Schüler:innen erkunden, wie Programmierer:innen Typen für eine effizientere Programmierung einsetzen. Sie lernen, die Typen entsprechend deren Methoden und Eigenschaften zu beschreiben.

Kapitel 7: Funktionen mit Parametern (4 Stunden). Die Schüler:innen untersuchen Verfahren, bei denen zusätzliche Informationen erforderlich sind, um das gewünschte Ergebnis zu erhalten. Sie lernen, wie sie ihre Funktionen mithilfe von Parametern, über die zusätzliche Angaben bereitgestellt werden, flexibler und leistungsfähiger gestalten können.

Kapitel 8: Logische Operatoren (6 Stunden). Die Schüler:innen erkunden Szenarien, in denen sie vor dem Treffen einer Entscheidung mehrere Faktoren berücksichtigen müssen. Sie lernen, logische Operatoren zu verwenden, um auf mehrere Bedingungen zu reagieren.

Kapitel 9: while-Schleifen (4,5 Stunden). Die Schüler:innen erkunden die Leistungsfähigkeit einer while-Schleife für die Wiederholung eines Codeabschnitts, bis eine bestimmte Bedingung erreicht ist.

Kapitel 10: Arrays und Refactoring (5 Stunden). Die Schüler:innen lernen, wie Daten in Arrays abgelegt werden, und erkunden, wie sich ihr Code über die Programmierung mit Arrays und Refactoring vereinfachen lässt.

Inspirationstagebuch: Die Schüler:innen erstellen ein Tagebuch basierend auf den Programmierkonzepten jedes Kapitels, während sie auf Aufforderungen reagieren und Aktivitäten durchführen, die ihr persönliches Leben und ihre Ideen mit dem Code verbinden.

Lehrerhandbuch: Dieses Handbuch unterstützt Lehrkräfte jedes Niveaus von Programmiererfahrung und bietet inklusive Lernerlebnisse, Bewertungsideen und Tipps, um eine große Bandbreite von Lernenden zu unterstützen. [Rätsel – Lehrerhandbuch herunterladen >](#)





Jede:r kann programmieren – Abenteuer

Jede:r kann programmieren – Abenteuer wurde für Schüler:innen ab 10 Jahren konzipiert und schließt an *Jede:r kann programmieren – Rätsel* an. In zirka 45 Stunden voller praktischer Aktivitäten können die Schüler:innen mit Hardwarefeatures und ereignisgesteuerter Programmierung experimentieren und ihre kreativen Ideen in Form von Code ausdrücken. Jedes Kapitel deckt anspruchsvollere Inhalte in der Swift Playgrounds App ab, z. B. Sensor Arcade, Blu's Adventure, Sonic Workshop und Augmented Reality. Die Schüler:innen lernen während des Kurses die unterschiedlichen Konzepte der App-Entwicklung sowie den App-Designprozess kennen, während sie sich für ein Abschlussprojekt mit Swift Playgrounds vorbereiten. [Jede:r kann programmieren – Abenteuer herunterladen >](#)

Kapitel 1: Objekte in Ansichten (6 Stunden). Die Schüler:innen untersuchen die Verwendung von Koordinaten zum Platzieren von Objekten in einer Ansicht (kurz für sichtbarer Bereich). Sie üben das Programmieren mit Koordinatenpaaren – den Schnittpunkten zwischen einer vertikalen und einer horizontalen Achse.

Kapitel 2: Ereignisse und Handler (5 Stunden). Anhand von Berührungseignissen erlernen die Schüler:innen die Grundlagen der Ereignisbehandlung. Sie untersuchen Funktionen mit Parametern und fügen Code hinzu, um ihre Bilder und Texte interaktiv zu gestalten.

Kapitel 3: Arrays (8 Stunden). Als Fortsetzung des Lernens zur Funktionalität von Ereignissen rekapitulieren und erweitern die Schüler:innen ihr Verständnis von Arrays. Die Schüler:innen lernen neue Operatoren und Methoden kennen, die sie für Arrays verwenden können. Außerdem befassen sie sich mit einigen Funktionalitäten von Arrays sowie mit Operatoren und verschachtelten for-Schleifen.

Kapitel 4: Weitere Ereignisse und Handler (5 Stunden). Die Schüler:innen kombinieren ihr Verständnis von Ereignissen und Handlern mit ihrem Verständnis von Arrays. Sie erlernen, welche Arten von Argumenten an eine Ereignishandler-Funktion übergeben werden können. Sie untersuchen die vielen Eingabearten, die ihnen über die iPad Gerätehardware zur Verfügung stehen.

Kapitel 5: Funktionen als Argumente (8 Stunden). Die Schüler:innen erkunden, wie sie Funktionen als Argumente innerhalb einer anderen Funktion mithilfe von Closures verwenden. Diese Erweiterung von Ereignissen unterstützt sie beim Eintauchen in das Thema der Augmented Reality.

Kapitel 6: Rückgabetypen und Ausgaben (3 Stunden). Die Schüler:innen lernen, wie sie eine Funktion verwenden, um einen bestimmten Typ zurückzugeben. Bis zu diesem Punkt haben die Schüler:innen Funktionen zum Verpacken von Prozeduren verwendet. Nun können sie einen Typ erstellen, der sich in anderen Teilen eines Programms verwenden lässt.

Kapitel 7: Klassen und Komponenten (7 Stunden). Die Schüler:innen untersuchen die Grundlagen von Komponenten und wie sich diese zusammenfügen lassen, um etwas Neues zu erschaffen. Sie setzen Design Thinking ein und untersuchen Typen. Die Schüler:innen lernen Klassen kennen, um Daten in einem Programm zu gruppieren und zu organisieren. Sie erfahren, wie diese Komponenten erstellt und miteinander verbunden werden, um App-ähnliche Erlebnisse zu erstellen.

Lehrerhandbuch: Dieses Handbuch unterstützt Lehrkräfte unabhängig von deren Programmiererfahrung und bietet inklusive Lernerlebnisse, Bewertungsideen und Tipps, um eine große Bandbreite von Lernenden zu unterstützen. [Abenteuer – Lehrerhandbuch herunterladen >](#)



Weitere Ressourcen

Diese Ressourcen bieten angehenden Programmier:innen zusätzliche Möglichkeiten, zu lernen, Prototypen zu erstellen und ihren Einfallsreichtum zu teilen. Sie sind eine großartige Möglichkeit, die Welt des Programmierens in der Schule, außerhalb der Schule oder zu Hause vorzustellen.



Schnellstart ins Programmieren

Dieses PDF-Handbuch enthält 10 Programmieraktivitäten für Schüler:innen ab 10 Jahren. Schüler:innen lernen zu programmieren – in der Schule oder zu Hause mit der kostenlosen Swift Playgrounds App für iPad und Mac.

[Schnellstart ins Programmieren herunterladen >](#)

Swift Coding Clubs

Swift Coding Clubs bieten eine großartige Chance, Programmierkonzepte bei Zusammenreffen nach der Schule, in Sommercamps oder in anderen ungezwungenen Lernumgebungen einzuführen. Das modulare Design des Swift Coding Clubs macht ihn perfekt für Einsteiger und Fortgeschrittene.

[Swift Coding Club Kit herunterladen >](#)

App-Design-Tagebuch

Mithilfe des App-Design-Tagebuchs können Schüler:innen den App-Design-Prozess anwenden, um Probleme in ihrer Schule oder Gemeinschaft zu lösen.

Das Tagebuch fordert die Schüler:innen auf, Ideen zu entwickeln, zu planen, Prototypen zu erstellen und ihre eigenen App-Ideen zu bewerten. Dies mündet in einer Präsentation des App-Prototyps. [App-Design-Tagebuch herunterladen >](#)

Leitfaden zur Präsentation von Apps

Die Schüler:innen sollten ermutigt werden, ihre Programmierleistungen im Rahmen von Community-Events wie Projektdemonstrationen oder Vorstellungen von Apps mit der breiteren Gemeinschaft zu teilen. Der Leitfaden zur Präsentation von Apps bietet praktische Unterstützung beim Planen und Durchführen eines Präsentationsevents.

[Leitfaden zur Präsentation von Apps herunterladen >](#)

Inklusiver Lehrplan

Wir bei Apple glauben, dass alle die Chance haben sollten, etwas zu erschaffen, das die Welt verändern kann. Für Lehrkräfte, die Schüler:innen mit Behinderung unterrichten, haben wir Ressourcen zu den Bedienungshilfen entwickelt, mit denen alle Schüler:innen programmieren lernen können.

Barrierefreie Handbücher

Die Schüler- und Lehrerhandbücher für „Jede:r kann programmieren“ wurden für VoiceOver optimiert und enthalten Videos mit erweiterten Untertiteln und Audiobeschreibungen. Jede Lektion wurde so gestaltet, dass sie barrierefreie Inhalte, Aktivitäten und Übungen enthält, damit alle Schüler:innen das Programmieren erlernen können.

Swift Playgrounds und Anleitungsvideos mit VoiceOver

Lehrkräfte und Schüler:innen können Anleitungsvideos vom Hadley Institute for the Blind and Visually Impaired ansehen, die ihnen den Einstieg in die Nutzung von Swift Playgrounds mit VoiceOver erleichtern. Die Schüler:innen erfahren, wie sie Swift Playgrounds laden, sich in den Rätselwelten orientieren, benutzerdefinierte Rotoraktionen verwenden, Code eingeben und ein einfaches Rätsel lösen. [Sammlung der Videos anzeigen >*](#)



Tactile Puzzle Worlds in Swift Playgrounds

Tactile Puzzle Worlds bietet Unified English Braille, großen Text und geprägte Grafiken, um blinden Schüler:innen und Schüler:innen mit Seheinschränkungen dabei zu helfen, durch die Programmierrätsel in Swift Playgrounds zu navigieren. Lehrkräfte können die tastbaren Grafiken mit ihren eigenen Brailledruckern ausdrucken oder gedruckte Exemplare bestellen. Der Download von Tactile Puzzle Worlds unten umfasst PDF Versionen der tastbaren Grafiken sowie Anleitungen zum Ausdrucken oder Bestellen von Exemplaren für Schüler:innen.

[Tactile Puzzle Worlds herunterladen >*](#)

Programmierkonzepte in American Sign Language (ASL)

Barrierefreie Videos in American Sign Language unterstützen gehörlose Schüler:innen dabei, programmieren zu lernen. Mit erweiterten Untertiteln und Texttranskriptionen sind sie so gestaltet, dass sie jeder verstehen und genießen kann. Das Lehrerhandbuch „Rätsel“ umfasst eine Reihe von Videos zur American Sign Language, in denen das Programmierkonzept erläutert wird. [Sammlung der Videos anzeigen >*](#)



* Diese Ressourcen sind derzeit nur auf Englisch verfügbar.

Unterstützung von Lehrkräften

Wir bieten zahlreiche Angebote, mit denen wir Lehrkräfte und Administratoren dabei unterstützen, Swift Playgrounds und Apple-Produkte bereitzustellen, zu verwalten und im Unterricht einzusetzen.

Jede:r kann programmieren – Ressourcen herunterladen

- [App „Swift Playgrounds“ für iPad](#)
- [App „Swift Playgrounds“ für Mac](#)

- [Jede:r kann programmieren – Vorschüler:innen](#)
- [Jede:r kann programmieren – Rätsel](#)
- [Jede:r kann programmieren – Rätsel – Lehrerhandbuch](#)
- [Jede:r kann programmieren – Abenteuer](#)
- [Lehrerhandbuch „Jede:r kann programmieren – Abenteuer“](#)

- [Schnellstart ins Programmieren](#)
- [Swift Coding Club Kit](#)
- [App-Design-Tagebuch](#)
- [Leitfaden zur Präsentation von Apps](#)

Weitere Informationen über Ressourcen zu „In Swift entwickeln“

„In Swift entwickeln“ bietet Lernenden praktische Erfahrung beim Entwerfen und Entwickeln von Apps mit Swift in Xcode, um sie auf die Hochschule, den Beruf und sogar eine branchenweit anerkannte Zertifizierung in der App-Entwicklung vorzubereiten.

- [In Swift entwickeln – Lehrplanführer](#)
- [In Swift entwickeln – Professional Learning-Kurse über Canvas](#)
- [Sammlung zu „Programmieren unterrichten“ in Apple Books](#)

Programmieren unterrichten

Diese Seite auf [apple.com](#) enthält Informationen für den Programmierunterricht sowie Programme von Apple, die Lehrkräfte beim Programmierunterricht in allen Altersstufen unterstützen – egal, ob sie gerade erst anfangen oder bereit für die Zertifizierung der Lernenden in Swift sind. [Weitere Infos >](#)

Apple Teacher

Apple Teacher ist ein kostenloses professionelles Selbstlernprogramm, das unbegrenzten Zugang zu Lernressourcen und -inhalten rund um den Einsatz von Technologie im Bildungsbereich bietet. [Weitere Infos >](#)

Apple Professional Learning

Apple Professional Learning Specialists vermitteln praxisorientierte, immersive Erfahrungen, die Lehrkräften dabei helfen, innovative Unterrichtspraktiken zu entwickeln und Schüler:innen für intensivere Lernerlebnisse zu begeistern. Weitere Informationen können Sie per E-Mail an AppleProfessionalLearning@apple.com anfordern.

