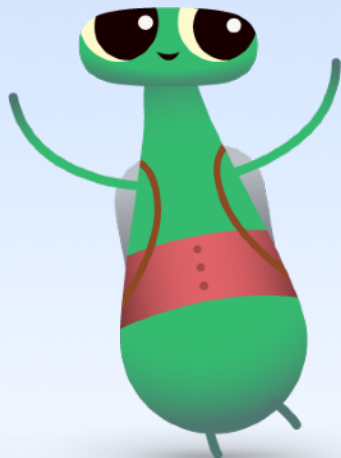




Swift Coding Club



Kit zu Swift Playgrounds

Willkommen beim Swift Coding Club!

Durch Programmieren lernt man, wie man Probleme lösen und auf kreative Weise zusammenarbeiten kann. Und es hilft dabei, Apps zu entwickeln, die Ideen Wirklichkeit werden lassen.

Swift Coding Clubs sind eine tolle Möglichkeit, um Programmieren und das Entwickeln von Apps zu lernen. Aktivitäten rund um Swift, die Programmiersprache von Apple, helfen Ihnen zusammenzuarbeiten, während Sie Programmieren lernen, App-Prototypen erstellen und darüber nachdenken, wie Code in der Welt um Sie herum etwas bewirken kann.

Sie müssen kein Lehrer oder Programmierexperte sein, um einen Swift Coding Club zu leiten. Die Materialien sind zum Selbststudium gedacht, sodass Sie sogar gemeinsam mit Ihren Clubmitgliedern lernen können. Und Sie können die Ideen und Designs Ihres Clubs bei einem Event für Ihre Gemeinschaft, auf dem die Apps präsentiert werden, anerkennen.

Dieses Kit ist in drei Abschnitte gegliedert:



Erste Schritte

Alles, was Sie brauchen, um einen Swift Coding Club zu gründen.



Lernen & Entwickeln

Tipps und Aktivitäten, um Clubsessions zu entwickeln.



Anerkennen

Hilfreiche Ressourcen, um eine App-Präsentation in Ihrer Gemeinschaft zu planen und durchzuführen.

Swift Coding Clubs

Blockbasiertes Programmieren | 8 bis 11 Jahre

Die Grundlagen des Programmierens mit visuell-basierten Apps auf dem iPad kennenlernen.



Swift Playgrounds | Ab 11 Jahren

Swift Code verwenden, um mit Swift Playgrounds auf dem iPad grundlegende Programmierkonzepte zu lernen.



Xcode | Ab 14 Jahren

Lernen, in Xcode auf dem Mac Apps zu entwickeln.



Erste Schritte



1. Clubmaterialien laden

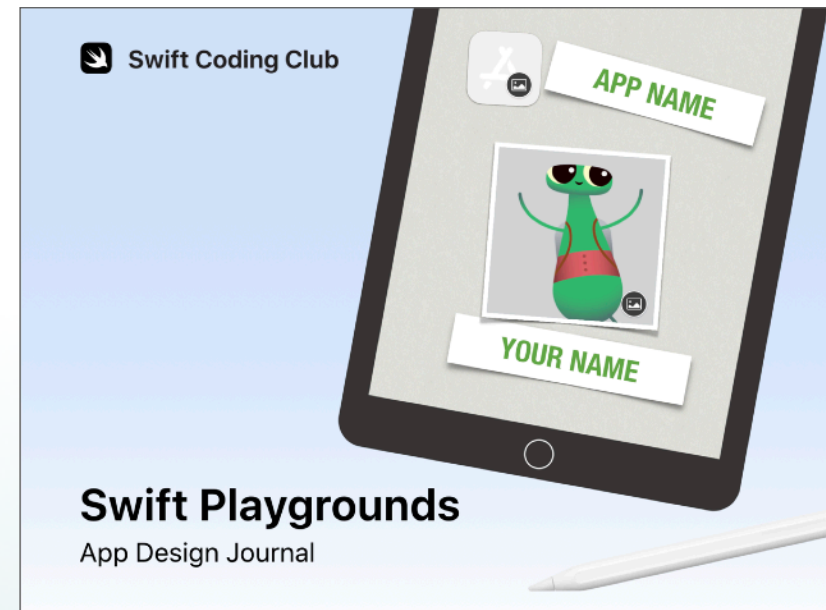
Teilen Sie diese beiden Leitfäden bei Ihrem ersten Clubmeeting über AirDrop mit Clubmitgliedern. Sie können Sie unten laden.



Programmieraktivitäten

Lernen Sie Programmierkonzepte mit diesen unterhaltsamen, gemeinschaftlichen Aktivitäten und lösen Sie Rätsel mit der Swift Playgrounds App auf dem iPad.

[„Swift Playgrounds – Programmieraktivitäten“ laden >](#)



App-Design-Tagebuch

Erkunden Sie den App-Designprozess mit diesem Keynote Tagebuch. Tragen Sie Ideen zusammen, planen und erstellen Sie einen Prototyp und beurteilen Sie die App-Ideen Ihres Clubs.

[„Swift Playgrounds – App-Design-Tagebuch“ laden >](#)



2. Ihre Technik überprüfen

Stellen Sie vor dem ersten Treffen sicher, dass Sie Folgendes haben:

- **iPad.** iPad mini 2 oder neuer, iPad Air oder neuer oder iPad Pro mit iOS 11 oder neuer. Es ist am besten, wenn jede Person ihr eigenes Gerät hat, aber sie können die Geräte auch teilen und zusammen programmieren.
- **Swift Playgrounds App.** [Swift Playgrounds laden >](#)
- **Playgrounds aus Programmieren lernen 1 & 2.** Laden Sie diese Playgrounds in der Swift Playgrounds App.
- **Keynote.** Sie werden die Keynote App auf dem iPad für Ihre App-Prototypen verwenden.
- **Swift Coding Club Materialien.**



3. Einen Plan erstellen.

Hier sind einige Punkte, die Sie beachten sollten:

- Wer sind Ihre Clubmitglieder? Was sind ihre Interessen? Haben sie Programmiererfahrung oder sind sie Anfänger?
- Wie oft wird sich Ihr Club treffen? Wenn Sie ein Sommer-Camp planen, wie viele Stunden mit Programmieraktivitäten wird es geben?
- Welche Technologien stehen dem Club zur Verfügung?
- Was sind die Ziele Ihres Clubs?



4. Es weitersagen.

Machen Sie andere auf Ihren Swift Coding Club aufmerksam. Hier sind einige Ideen und Ressourcen, um neue Mitglieder für Ihren Club zu gewinnen:

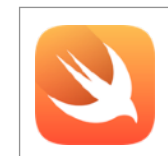
- **Kündigen Sie Ihren Club an.** Nutzen Sie E-Mails, soziale Netzwerke, das Internet, Flyer oder Mundpropaganda, um Ihre Gemeinschaft auf den Club aufmerksam zu machen.
- **Führen Sie eine Informationsveranstaltung durch.** Fragen Sie potenzielle Clubmitglieder, wofür sie sich interessieren und welche Arten von Apps sie erstellen möchten. Sprechen Sie über Ihre Ideen für eine Präsentation der App-Designs und wie die Mitglieder sich daran beteiligen können. Sie können auch ein kurzes Video über den Club online teilen.

Diese Dinge können Ihnen dabei helfen, Ihren Swift Coding Club persönlicher zu machen und ihn zu bewerben:

- **Poster.** [Laden Sie diese kostenlose Vorlage](#) und personalisieren Sie sie, um Ihr eigenes Poster zu erstellen. Drucken Sie es aus und hängen Sie es auf oder erstellen Sie ein digitales Poster, um es online zu teilen. Vergessen Sie nicht zu erwähnen, wann und wo der Club sich trifft und wie man Mitglied wird.
- **Aufkleber und T-Shirts.** Verwenden Sie diese [Swift Coding Club Aufkleber](#), um auf Ihren Club aufmerksam zu machen. T-Shirts sind eine tolle Möglichkeit, um Clubmitglieder zu erkennen, die an Events zur Präsentation von Apps teilnehmen. Laden Sie die [Swift Coding Club T-Shirt-Vorlage](#), um T-Shirts für Ihre Mitglieder zu machen.



Swift Coding Club Poster



Swift Coding Club Aufkleber

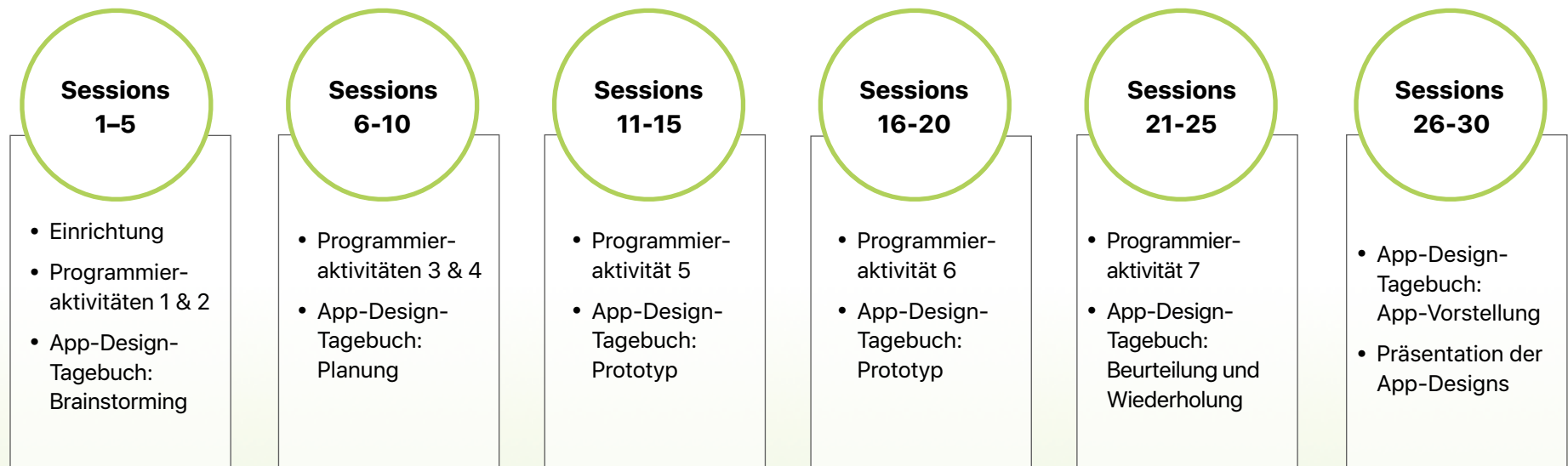


Swift Coding Club T-Shirt



Lernen & Entwickeln

Die Clubmaterialien sind so gestaltet, dass Sie Programmier- und App-Design-Aktivitäten miteinander verbinden können. Sie können außerdem Sessions hinzufügen, die die Interessen Ihrer Mitglieder unterstützen. Unten sehen Sie ein Beispiel für ein Programm mit 30 einstündigen Clubsessions.



Erwägen Sie, Sessions hinzuzufügen, um App-Design- und Programmieraktivitäten zu vertiefen. Diese können zum Beispiel das Erstellen eines Hindernisparcours für eine Drohne oder eine Rettungsmission für einen Roboter enthalten. Um ein Brainstorming für das App-Design anzuregen, können Sie sogar Gastredner einladen oder Exkursionen machen.



Ein leitendes Team zusammenstellen. Eine Gruppe von Mitgliedern zu haben, die helfen, den Club zu leiten, kann es viel einfacher und unterhaltsamer machen. Welche Clubmitglieder eignen sich als Leiter? Überlegen Sie, leitende Positionen für Ihren Club für Events, Programmieren, App-Design und mehr zu ernennen.

Gemeinsam lernen. Clubleiter müssen nicht alles wissen. Helfen Sie Ihren Mitgliedern dabei, Ihre eigenen Recherche- und Problemlösungskompetenzen zu entwickeln und ermutigen Sie sie, anderen zu helfen.

Präsentieren. Ein Event zur Präsentation von Apps ist eine großartige Möglichkeit, um Freunden, Familie, Lehrern und der Gemeinschaft Ihren Club, Ihre App-Ideen und Ihre Programmierkenntnisse zu präsentieren. Vielleicht können Sie dadurch sogar neue Mitglieder gewinnen. Auf [Seite 13](#) finden Sie Tipps für die Durchführung Ihrer eigenen App-Präsentation.



Ideen austauschen. Einige Mitglieder werden sich dafür interessieren, Spiele zu erstellen. Andere möchten vielleicht Apps erstellen, um Swift zu lernen, Menschen zu helfen oder Roboter zu steuern. Denken Sie über Möglichkeiten nach, wie Mitglieder gemeinsam an Projekten arbeiten können, die ihnen wichtig sind.

Gemischte Zweiergruppen erstellen. Manchmal können Mitglieder mit fortgeschritteneren Fähigkeiten andere abhängen. Versuchen Sie, ob diese Mitglieder mit einem Anfänger als Partner programmieren können. Jemandem etwas beizubringen, ist eine gute Gelegenheit, um zu lernen.



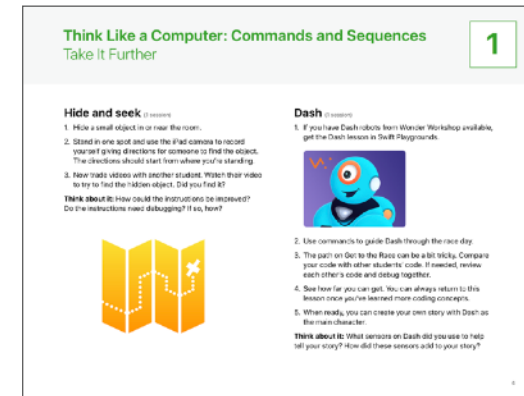
Swift Playgrounds – Programmieraktivitäten



Programmieraktivitäten: Diese gemeinschaftlichen Aktivitäten rund um Swift Playgrounds stellen grundlegende Programmierkonzepte und -fähigkeiten vor.

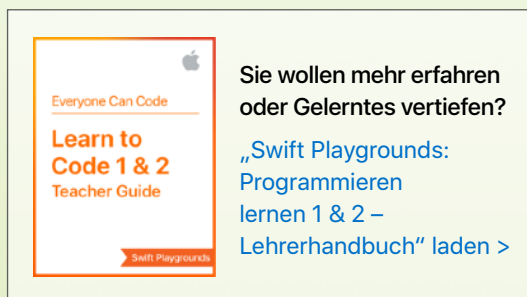


Programmierkonzepte: In jeder Aktivität lernen Clubmitglieder ein grundlegendes Programmierkonzept kennen und erkunden es in einem alltäglichen Kontext. Dann wenden sie das Programmierkonzept an, um Rätsel in Swift Playgrounds zu lösen.



Das Gelernte vertiefen: Jedes Programmierkonzept hat zwei Aktivitäten, um das Gelernte zu vertiefen. Die erste Aktivität vertieft das Verständnis des Programmierkonzepts und fördert Kommunikation und Teamwork. Die Mitglieder nutzen das iPad, um das Gelernte in einem kreativen Projekt anzuwenden.

Die zweite optionale Aktivität fordert Mitglieder heraus, das Konzept in einem Playground aus den Bereichen „Herausforderungen“, „Ausgangspunkte“ und „Abonnemente“ in Swift Playgrounds anzuwenden. Einige Aktivitäten erfordern bestimmte verbundene Geräte.



Tipps zum Lernen mit Swift Playgrounds



Zuerst die Rätsel erkunden. Ermutigen Sie die Clubmitglieder, die Welt von Byte in der Live-Ansicht zu zoomen und zu drehen, damit sie sich eine klare Vorstellung vom Ziel machen können. Sie können die Welt auch im Vollbildmodus ansehen, indem sie die Trennlinie zwischen den beiden Fenstern berühren und halten und sie nach links bewegen.

Die Rätsel aufteilen. Die Rätsel werden schwierig. Die Clubmitglieder können ein Rätsel aufteilen, wenn es ihnen hilft, die Schritte durchzudenken, und das Rätsel zu lösen. Sie können in Pages oder Notizen ihre Schritte planen und aufschreiben, bevor sie den Code eingeben.

Einen Helpdesk einrichten. Richten Sie einen Ort ein, an dem Clubexperten anderen Mitgliedern ihre Hilfe anbieten können.



Auf verschiedene Arten lösen. Jedes Rätsel hat viele Lösungen. Wenn Mitglieder früh fertig sind, ermutigen Sie sie, sich andere Lösungsmöglichkeiten für die Rätsel zu überlegen. Eine flexible Denkweise und der Vergleich von verschiedenen Lösungen kann ihnen helfen, ihre Fähigkeit zum kritischen Denken zu verbessern.


Programmierpaare. Lassen Sie Clubmitglieder gemeinsam auf einem iPad arbeiten. Sie können Ideen zusammentragen, wie sie die Rätsel lösen wollen, und sich beim Programmieren abwechseln.

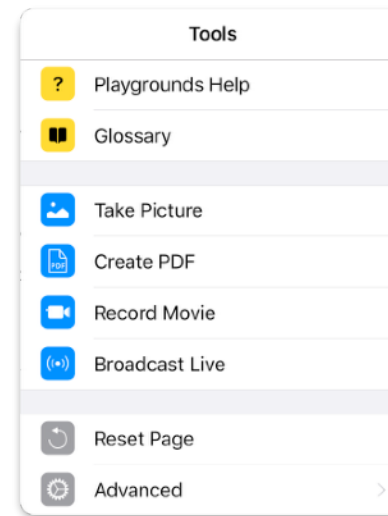
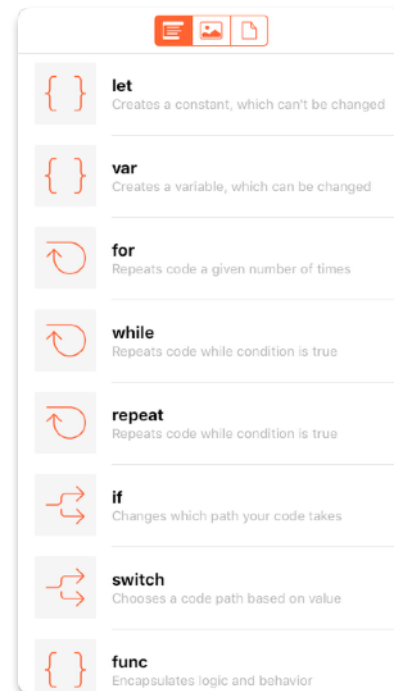
Bedienungshilfen einsetzen. Swift Playgrounds funktioniert gut mit den in iOS integrierten Bedienungshilfen, damit jeder Programmieren lernen kann. So können Programmierer zum Beispiel die Farben umkehren, Graustufen aktivieren und zoomen, um die Sichtbarkeit anzupassen.

Swift Playgrounds erkunden

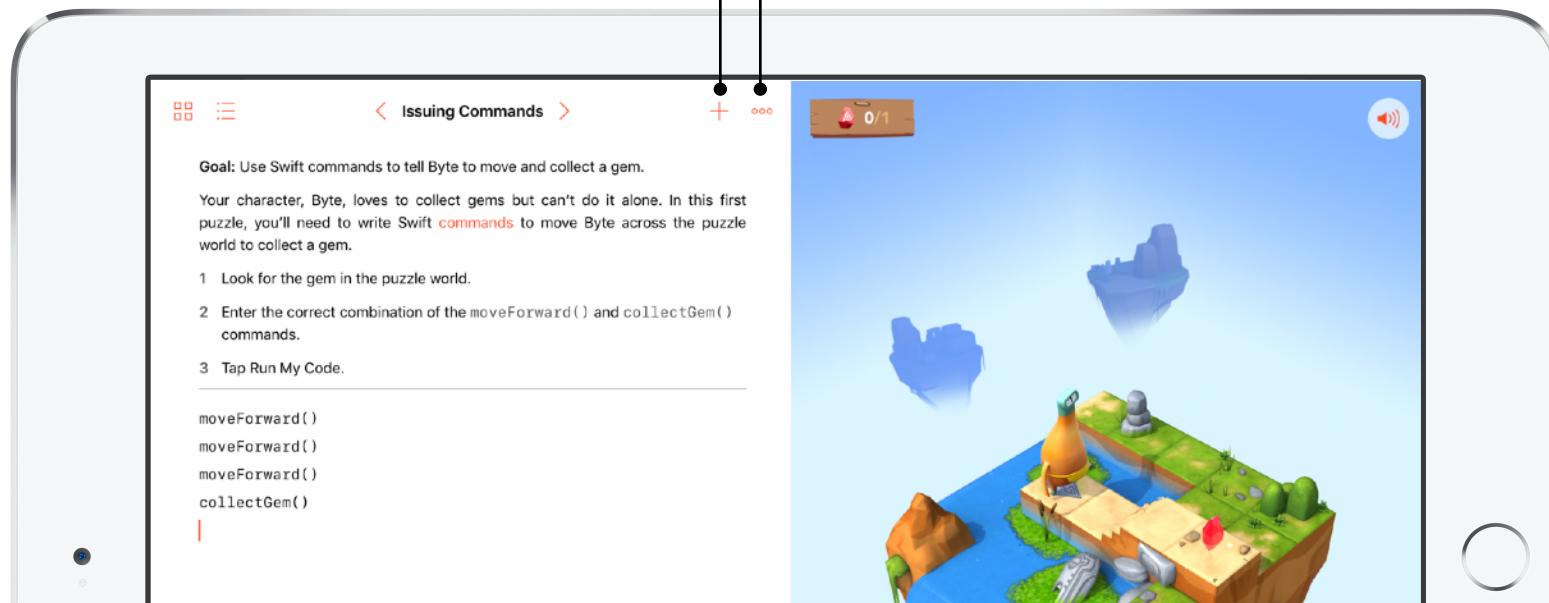


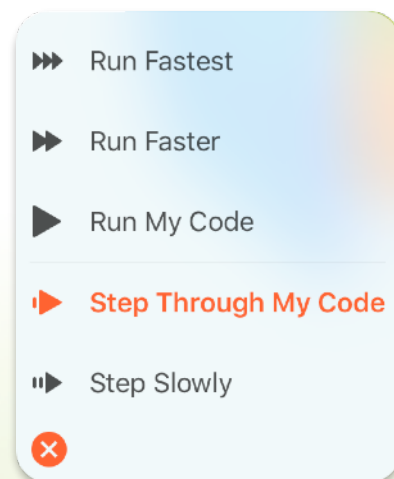
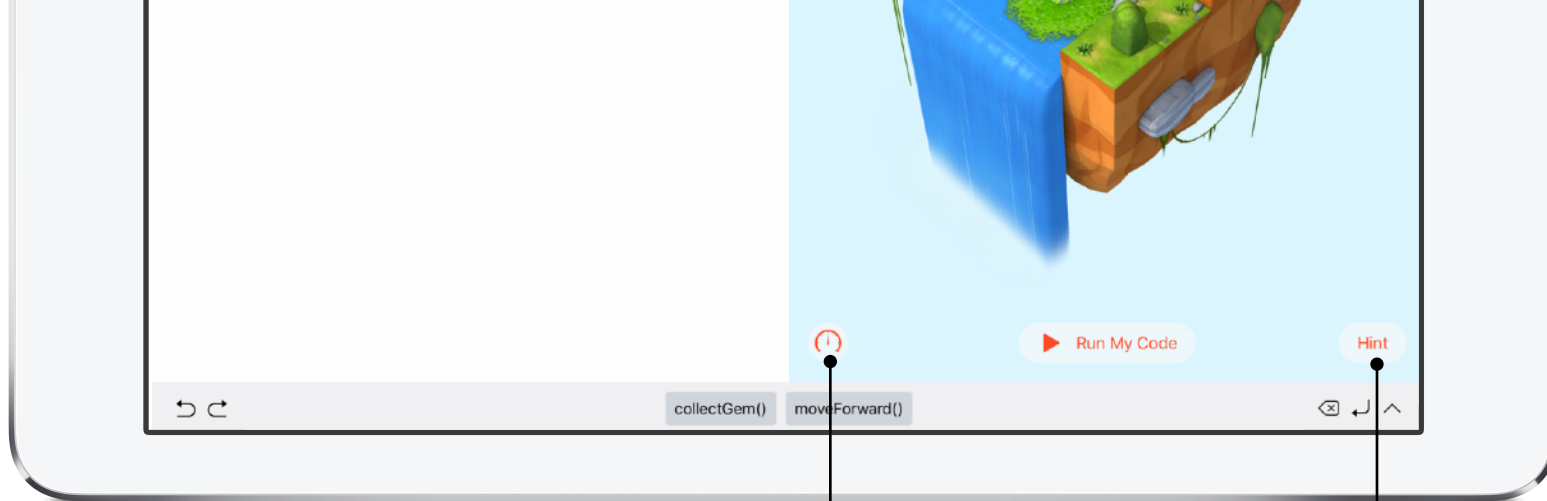
Snippet Mediathek.

Tippen Sie in der Symbolleiste auf , um auf die Snippet Mediathek zuzugreifen und häufig verwendete Codeteile schnell aus der Mediathek zu ziehen, ohne viel eintippen zu müssen.



Eine Figur wählen. Personalisieren Sie Ihr Erlebnis, indem Sie auf den Charakter tippen und einen anderen auswählen.





Hinweis. Dieses Feature bietet Vorschläge, um den Schülern zu helfen. Es zeigt auch irgendwann die Lösung eines Rätsels an, jedoch können Programmierer die Lösung nicht einfach kopieren und einsetzen. Um weiterzukommen, müssen sie trotzdem die Schritte abschließen und den Code selbst schreiben.

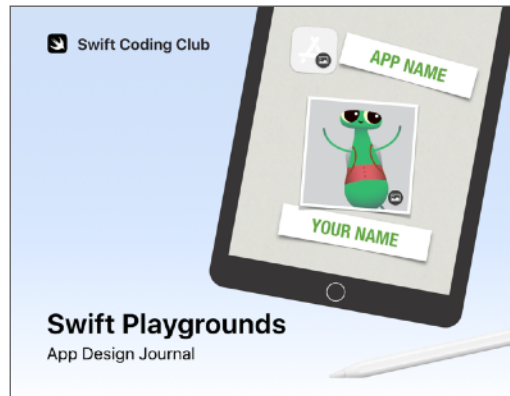
Die Geschwindigkeit steuern.

Beschleunigen oder verlangsamen Sie den Code.

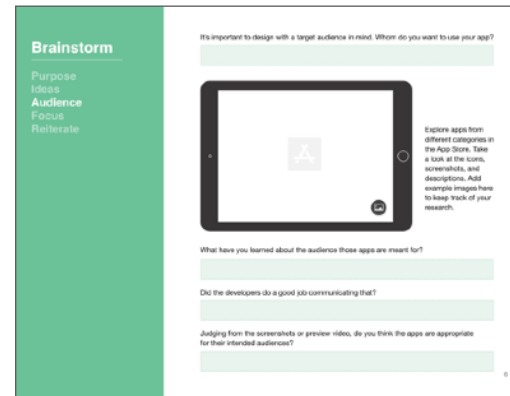
Code hervorheben, während er ausgeführt wird. Wählen Sie „Schrittweise ausführen“, um jede Zeile des Codes während der Ausführung hervorzuheben und besser verstehen zu können, was der Code macht.



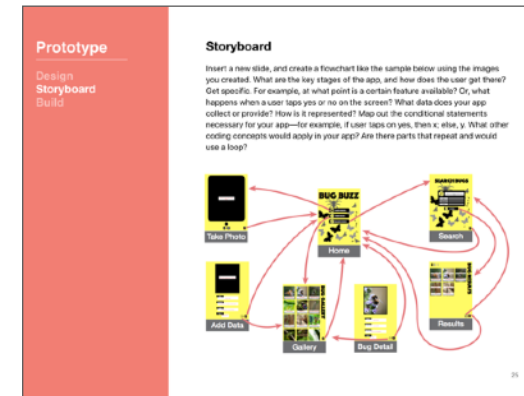
App-Design-Tagebuch



Die Programmierer lernen mit diesem Keynote Tagebuch mehr über App-Features und entwickeln eine App, um ein Problem in der Gemeinschaft zu lösen.



Die Clubmitglieder arbeiten in kleinen Teams, um Ideen zusammenzutragen und die App-Lösung zu planen. Dann erstellen sie einen funktionierenden Prototyp der App in Keynote.



Das Tagebuch führt Programmierer durch den Prozess der Beurteilung ihrer Designs und der Verbesserung ihrer Prototypen – genau wie bei professionellen App-Entwicklern.

Die Mitglieder erstellen eine dreiminütige Präsentation oder ein Video zur Vorstellung Ihrer App, um ihre Arbeiten zu präsentieren.



Anerkennen



Präsentation der App-Designs

Der Prozess des App-Designs und die Präsentation der Apps sind großartige Möglichkeiten, um die Gemeinschaft miteinzubeziehen und zu erkunden, welches Potenzial Apps haben, um aktuelle Probleme zu lösen. Die Präsentation ist außerdem die perfekte Gelegenheit, um zu zeigen, wie talentiert Ihre Clubmitglieder sind!

1. Das große Event planen. Legen Sie ein Datum für den Tag der Präsentation fest und laden Sie Schüler, Lehrer, Eltern und Gemeindemitglieder dazu ein.

Geben Sie jedem Team die Möglichkeit, seine App vorzustellen und eine kurze Fragerunde abzuhalten. Wenn Sie eine große Gruppe haben, können Sie den Club in zwei Runden aufteilen, sodass sich die Mitglieder die Vorstellungen der anderen Gruppe ansehen können.

Erwägen Sie, das Event mit einer lustigen Diashow von Fotos zu beenden, die während der Clubsessions entstanden sind.



2. Design-Awards. Ein freundschaftlicher Wettbewerb kann eine großartiger Motivator sein. Inspirieren Sie Clubmitglieder, indem Sie Auszeichnungen für spezielle Stärken beim App-Design verleihen. Erwägen Sie Auszeichnungen für:

- Die beste Entwicklung
- Die beste Innovation
- Das beste Design
- Die beste Vorstellung

Sie können auch das Publikum zur Teilnahme motivieren, indem Sie einen Publikumspreis vergeben.



Sie können dieses [Zertifikat](#) laden und an verschiedene Auszeichnungen anpassen.



3. Jury und Mentoren finden. Jury und Mentoren können Lehrer oder Mitarbeiter sein, Schüler mit Programmiererfahrung, Experten für Entwicklung oder Design, Mitglieder des Schulvorstands, regionale Gemeindeleiter oder Personen, die von der App-Idee profitieren würden.

Die Jury muss nicht bis zur Präsentation warten, um den Club kennenzulernen. Erwägen Sie, sie als Gastredner einzuladen, wenn die Schüler in der Brainstorming- oder Planungsphase Ihres App-Designprozesses sind, damit sie ihr Fachwissen teilen können.

4. Einen Sieger wählen. Die Jury kann das Schema auf der nächsten Seite verwenden, das ihnen beim Bewerten der App-Vorstellungen und beim Feedback hilft. Sie könnten das Schema außerdem vor der Präsentation mit den Programmierern teilen, als Teil der Bewertungsphase des App-Designprozesses.

5. Teilen und inspirieren. Vielleicht möchten Sie die Präsentationen der Apps aufzeichnen. Teilen Sie die Aufzeichnungen mit der Gemeinschaft und erstellen Sie ein Video mit Highlights, um zukünftige Clubmitglieder zu inspirieren.



Bewertungsschema

[Laden >](#)

Kategorie	Anfänger (1 Punkt)	Fortgeschritten (2 Punkte)	Gut (3 Punkte)	Sehr gut (4 Punkte)	Punkte
Inhalt der Vorstellung	Teilt grundlegende Informationen wie Zweck und Zielgruppe	Gibt klare Beispiele für den Zweck der App, ihr Design und wie sie den Bedürfnissen der Benutzer gerecht wird	Beinhaltet eine klare und überzeugende Beschreibung des zu lösenden Problems, der Marktnachfrage, Zielgruppe und wie das Design der App auf die Bedürfnisse der Benutzer eingeht	Enthält ein überzeugendes und auf Beweisen basierendes Argument, das zeigt, wie die App die Benutzerbedürfnisse erfüllt, übertrifft oder neu definiert	
Art der Präsentation	Informativ; ein Teammitglied stellt vor	Selbstbewusst, enthusiastisch; mehr als ein Teammitglied stellt vor	Ansprechende, gelungene Verwendung von Bildern, um die Geschichte zu unterstützen; das Team hebt die Beiträge der einzelnen Mitglieder hervor	Kreatives, erinnerungswürdiges Geschichtenerzählen; überzeugende visuelle Unterstützung, flüssige Übergänge zwischen den Teammitgliedern	
Benutzeroberfläche	Konsistente Bildschirme, die den Zweck der App unterstützen	Klares, funktionales Design mit vertrauten Elementen; Prototyp unterstützt grundlegende Aufgaben der Benutzer	Elegantes, prägnantes, ansprechendes Design; Farbeinsatz, Layout und Lesbarkeit gut durchdacht; Prototyp vermittelt den Benutzern ein klares Gefühl für die Navigation	Das Design ermöglicht es dem Benutzer, mit den Inhalten zu interagieren; Prototyp nutzt Animationen, Farben und Layout, um ein nahtloses, ansprechendes Erlebnis zu ermöglichen	
Benutzererlebnis	Klare Zielsetzung; Benutzer können ein oder mehrere Ziele verfolgen	Einheitliche und standardmäßige Navigation; intuitiver Weg durch die Inhalte der App	Anpassbar an Bedürfnisse der Benutzer; berücksichtigt Barrierefreiheit, Datenschutz und Sicherheit	Innovativ, überraschend und begeisternd; gibt Benutzern ein neuartiges Erlebnis, das die App von Mitbewerbern abhebt	
Programmierkonzepte	Gewisser Zusammenhang zwischen der Funktionalität der App und dem zugrunde liegenden Code erkennbar	Erklärung, welche Rolle allgemeine Programmierkonzepte wie Datentypen, bedingte Logik oder Berührungseignisse für die App spielen	Beschreibung bestimmter Programmieraufgaben, die für den Aufbau der App nötig waren; Demonstration, wie dieser Code zur Funktionalität der App beiträgt	Erklärung zur Architektur der App, ihrer Datenstruktur, Algorithmen und Features; Diskussion des Entscheidungsprozesses bei der Entwicklung dieses Ansatzes	
Technische Besprechung (Optional) <i>Für funktionale App-Prototypen in Xcode. Die Jurymitglieder sollten mit den Best Practices für die Entwicklung mit Swift und für iOS vertraut sein.</i>	Der Swift Code funktioniert in bestimmten Beispielen; der Code ist grundlegend ohne Abstraktion	Der Code funktioniert durchgehend ohne Fehler; der Code ist grundlegend mit einigen Andeutungen von Abstraktion	Der Code ist mit Namenskonventionen von Swift organisiert; starke Hinweise auf Abstraktion; iOS Richtlinien werden befolgt	Der Code ist gut mit Kommentaren dokumentiert; Swift Features werden effizient eingesetzt; nutzt Organisation wie Model-View-Controller	
Anmerkungen:					0 Gesamt- ergebnis



Swift Coding Club

Swift Playgrounds

Zertifikat

Vergeben an

Für



Unterschrift

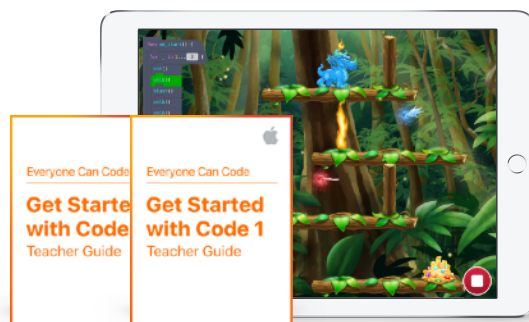
Datum

Nächste Schritte

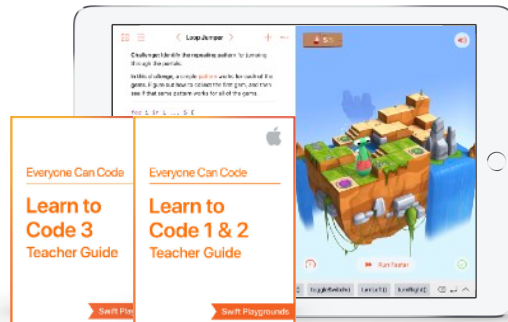
Der Swift Coding Club ist erst der Anfang Ihres Programmierabenteuers. Das Programm „Jeder kann programmieren“ bietet unterhaltsame, hilfreiche Ressourcen, um Programmierer auf ihrem Weg vom Erlernen der Grundlagen auf dem iPad bis zum Entwickeln echter Apps auf dem Mac zu begleiten.

Und Sie müssen nicht bei den Clubaktivitäten aufhören. Umfassende Lehrerhandbücher ermöglichen es Lehrern zudem, das Programmieren in den Unterricht einzubringen – mit an den Lehrplan angeglichenen Schritt-für-Schritt-Lektionen für alle Altersstufen vom Kindergarten bis zur Uni.

[Alle Ressourcen zu „Jeder kann programmieren“ >](#)



Laden Sie
den Lehrplanführer
„Erste Schritte mit Code“ >



Laden Sie
den Lehrplanführer
zu Swift Playgrounds >



Laden Sie
den Lehrplanführer
„App Development with Swift“ >



