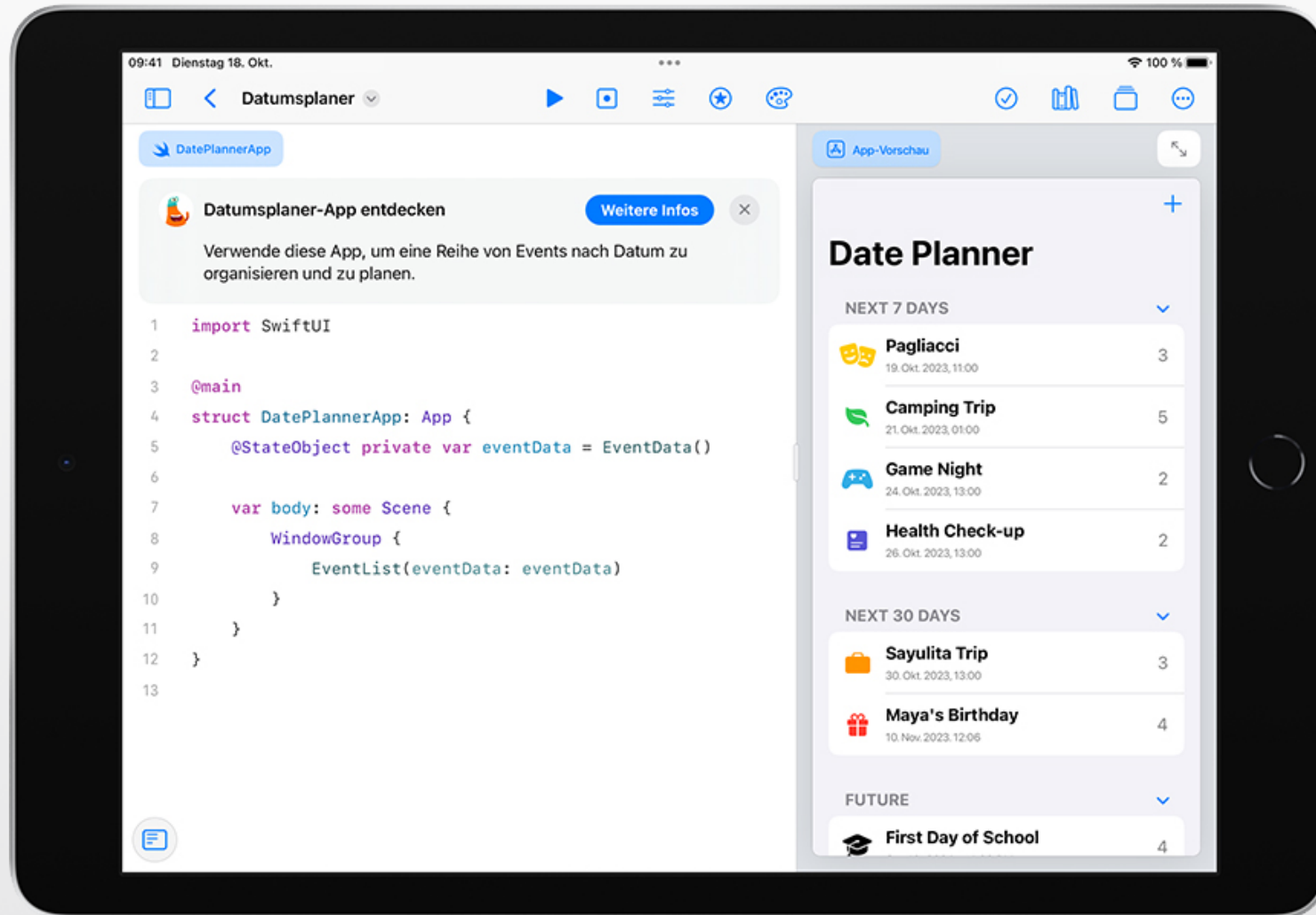


# Apple Everyone Can Code

## Lehrplanführer



# 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 – wie Schleifen und Funktionsdefinitionen – können einfach um bestehenden Code gezogen werden. Dazu muss nur das Keyword, z. B. *for*, berührt werden, um die Steuerung zum Ziehen anzuzeigen.

**Kurzbefehlleiste.** Am unteren Rand des Bildschirms werden QuickType Vorschläge angezeigt, sodass die Schüler:innen den gewünschten 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 aufnehmen, um ihre Arbeit zu demonstrieren.

## Spannende Animationen.

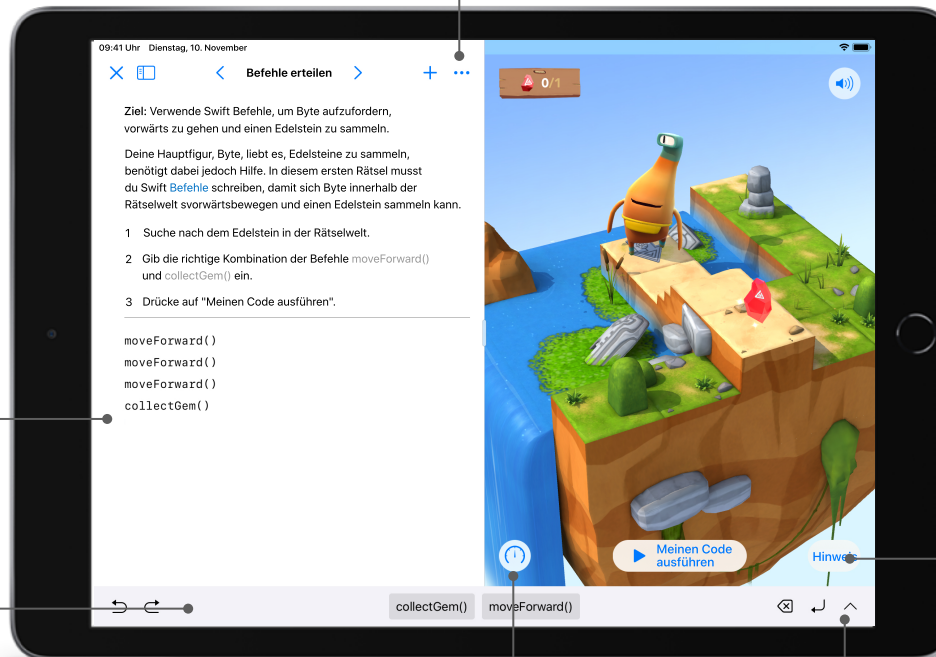
Jeder Abschnitt beginnt mit einer spannenden Animation, in der die Programmierkonzepte auf das reale Leben übertragen werden, um den Schüler:innen das Verständnis zu erleichtern.

**Bedienungshilfen.** Swift Playgrounds wurde mit Blick auf Barrierefreiheit konzipiert. Es nutzt die vielen leistungsstarken Bedienungshilfen von iPadOS und macOS, darunter Schaltersteuerung und VoiceOver. Und es gibt sogar zusätzliche Sprachkommentare zu den Aktionen der Charaktere, wenn Schüler:innen sie mit ihrem Code steuern.

**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.

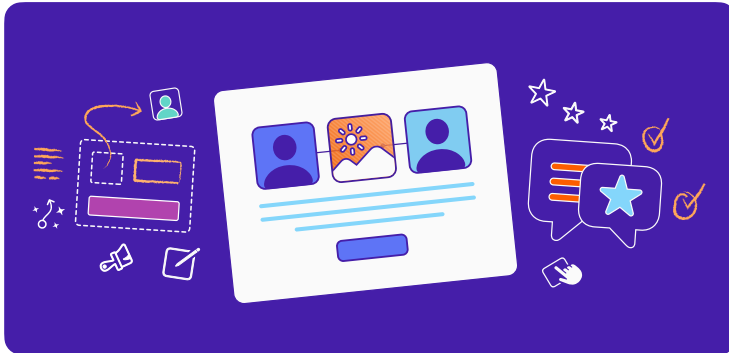
**Onscreen-Tastatur.** Eine speziell für Swift konzipierte Tastatur ermöglicht die schnelle Eingabe von Zahlen und Symbolen, die in Swift häufig verwendet werden.

**Code überprüfen.** Die Schüler:innen können Code schneller oder langsamer ausführen oder schrittweise durchlaufen, um die Codezeilen während ihrer Ausführung hervorzuheben. So können sie einfacher mögliche Fehlerstellen erkennen.



# „Jede:r kann programmieren“ Projekte

Mit den neuen „Jede:r kann programmieren“ Projekten ist es einfacher denn je, App-Design und -Entwicklung mit Swift Playgrounds auf iPad und Mac zu unterrichten und zu lernen. Mit diesen in der Apple Education Community verfügbaren Schritt-für-Schritt-Anleitungen können Lehrkräfte ihren Schüler:innen helfen, grundlegende Fähigkeiten aufzubauen, während sie Apps erstellen und damit Probleme lösen, die ihnen wichtig sind.



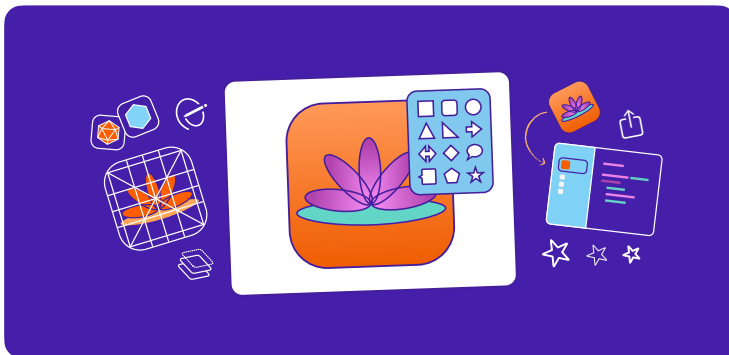
## Eine einfache App entwerfen (2 Stunden)

Führen Sie Ihre Schüler:innen durch das Erstellen eines einfachen App-Prototyps in Keynote für alles, was sie sich ausdenken können.



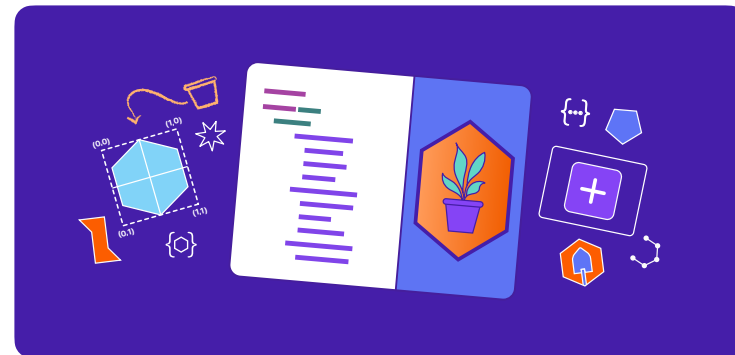
## Mit Stapeln und Formen kreativ werden (1 Stunde)

Stellen Sie App-Entwicklung vor, indem Sie mit Farben, Größen und der Platzierung experimentieren, um eine einzigartige App-Oberfläche zu erstellen.



## Ein App-Symbol entwerfen (1 Stunde)

Inspirieren Sie Schüler:innen, Designprinzipien anzuwenden, um ein tolles App-Symbol für jedes Thema zu erstellen.



## Eigene Formen erstellen (1 Stunde)

Zeigen Sie Ihren Schüler:innen, wie sie ihre App-Benutzeroberfläche auf das nächste Level bringen, indem sie mit SwiftUI eigene Formen programmieren.



## Jede:r kann programmieren – Rätsel

„Jede:r kann programmieren – Rätsel“ ist ein Handbuch zu Swift Playgrounds und umfasst über 45 Stunden an flexiblen Aktivitäten, die Schüler:innen ab der vierten Klasse dabei helfen, Problemlösungsstrategien und kritisches Denken zu entwickeln. 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“ laden >](#)

**Kapitel 1: Befehle (3 Stunden).** Die Schüler:innen lernen, wie wichtig eindeutige, präzise Befehle sind. 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 in einer Definition zusammenfassen, die sie immer wieder verwenden können. Sie programmieren eine Funktion, um eine Tanzroutine für einen Roboter zu choreografieren, und überlegen, welche Arten von Funktionen in alltäglichen digitalen Technologien angewendet werden könnten.

**Kapitel 3: for-Schleifen (3,5 Stunden).** Die Schüler:innen erkennen Muster in ihrer Umgebung und in ihrem Code, und sie lernen, wie sie Schleifen verwenden können, um effizienteren Code zu schreiben.

**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 des Wertes von Variablen auf die Programmausgabe 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, bedingten Code zu schreiben, 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, 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 machen 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 den Nutzen einer while-Schleife, um einen Codeabschnitt zu wiederholen, bis eine Bedingung erfüllt ist.

**Kapitel 10: Arrays und Refactoring (5 Stunden).** Die Schüler:innen lernen, wie Daten in Arrays gespeichert werden, und sie erkunden, wie die Verwendung von Arrays und Refactoring ihren Code vereinfachen kann.

**Inspirationstagebuch.** Die Schüler:innen erstellen ein Tagebuch basierend auf den Programmierkonzepten der einzelnen Kapitel, reagieren auf Aufforderungen und führen Aktivitäten durch, die ihr persönliches Leben und ihre Ideen mit dem Code verbinden.

**Lehrerhandbuch.** Dieses Handbuch unterstützt Lehrkräfte unabhängig von ihrer Programmiererfahrung und bietet inklusive Lernerfahrungen, Bewertungsideen und Tipps, um eine große Bandbreite von Lernenden zu unterstützen. Das Handbuch ist auf die Informatikstandards amerikanischer Highschools der Computer Science Teachers Association (CSTA) für Level 1B (Klassen 3–5) abgestimmt, die die Konzepte Algorithmen und Programmierung sowie die Auswirkungen der Informatik abdecken. [Lehrerhandbuch](#)  
[„Jede:r kann programmieren – Rätsel“ laden >](#)







## Jede:r kann programmieren – Vorschüler:innen

Dieses Handbuch wurde für Pädagog:innen entwickelt, die Schüler:innen im Alter von 5–8 Jahren unterrichten, und umfasst fünf Module: Befehle, Funktionen, Schleifen, Variablen und App-Design. Jedes Modul enthält Lektionen, die den Lernenden helfen, durch Naturwissenschaften, Kunst, Musik und mehr neue Programmierkonzepte zu erkunden. Die Lernenden teilen persönliche Erfahrungen und Ideen, während sie sich mit Programmierthemen beschäftigen. Durch praktische Aktivitäten und Herausforderungen entdecken sie, wie Code funktioniert, und wenden ihre neuen Fähigkeiten an, indem sie in der App „Swift Playgrounds“ Code schreiben. 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“ laden >](#)

**Befehle (3 Stunden).** Die Lernenden verbinden alltägliche Beispiele von exekutiven Funktionen, Sprache und Bewegung mit Code. Dieses Modul konzentriert sich auf die Beschreibung von Schritt-für-Schritt-Anweisungen, das Anordnen 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 Lernenden 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).** Aktivitäten zu Naturwissenschaften, Sprache und Gemeinschaftsbildung werden als Grundlage für das Erlernen von Variablen verwendet. Die Lernenden verknüpfen einen Variablennamen mit einem bestimmten Wert, ändern den Wert einer Variablen und verwenden verschiedene Variablentypen.

**App-Design (3 Stunden).** Die Lernenden werden durch eine Reihe von unterhaltsamen App-Design-Aktivitäten dazu inspiriert, Probleme zu lösen, die ihnen in der Schule oder in der Gemeinschaft wichtig sind. Sie nutzen Design Thinking, Kreativität und Empathie, um eine App-Idee zu entwickeln.



# Bedienungshilfen

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

## Barrierefreie Handbücher

Die Schüler- und Lehrerhandbücher zu „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 Lernenden 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. Sie erfahren, wie sie Swift Playgrounds laden, sich in den Rätselwelten orientieren, benutzerdefinierte Rotoraktionen verwenden, Code eingeben und ein einfaches Rätsel lösen. [Videosammlung hier anzeigen >](#)



## Tactile Puzzle Worlds in Swift Playgrounds

Tactile Puzzle Worlds bietet Unified English Braille, großen Text und geprägte Grafiken, um blinden oder sehbehinderten Schüler:innen bei der Navigation in den Programmerrätseln von Swift Playgrounds zu helfen. 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 laden >](#)

## Programmierkonzepte in amerikanischer Gebärdensprache (ASL)

Barrierefreie Videos in amerikanischer Gebärdensprache unterstützen gehörlose Schüler:innen dabei, Programmieren zu lernen. Mit erweiterten Untertiteln und Texttranskriptionen sind sie für alle verständlich und unterhaltsam. Und das *Lehrerhandbuch „Jede:r kann programmieren – Rätsel“* enthält eine Reihe von Videos in amerikanischer Gebärdensprache, die Programmierkonzepte erklären. [Videosammlung hier anzeigen >](#)



# Treten Sie der Apple Education Community bei



Die Apple Education Community ist ein Online-Weiterbildungszentrum für Lehrkräfte, die Apple Technologie für das Lehren und Lernen einsetzen. Im Learning Center finden Sie großartige Ressourcen und Anerkennungen von Apple und im Forum können Sie von Gleichgesinnten lernen – es ist ein Ort, in dem sich Lehrkräfte vernetzen und austauschen können. Die Mitgliedschaft in der Community ist kostenlos und die Website ist auf jedem Gerät und zu jeder Zeit verfügbar. Profitieren Sie von den Vorteilen und [werden Sie Mitglied](#).

Begleiten Sie Lernende von ihrer ersten Codezeile bis zur Erstellung ihrer ersten App. [Programmieren unterrichten mit Swift Playgrounds](#) in der Community bietet hilfreiche Anleitungen und Video-Tutorials zur Bedienung von Swift Playgrounds, zur Verwendung der In-App-Anleitungen und zum Schreiben von grundlegendem Code. Und wenn Sie bereit sind, helfen Ihnen die für Lehrkräfte entwickelten „[Jede:r kann programmieren](#)“ Projekte, ihre Schüler:innen beim Entwerfen und Erstellen ihrer ersten Apps mit Swift Playgrounds zu unterstützen.

Erfahren Sie mehr über diese Ressourcen und entdecken Sie die Apple Education Community unter [education.apple.com](https://education.apple.com).



© 2024 Apple Inc. Alle Rechte vorbehalten. Apple, das Apple Logo, iPad, iPadOS, Keynote, Mac, macOS, QuickType, Swift, das Swift Logo, Swift Playgrounds, SwiftUI und Xcode sind Marken von Apple Inc., die in den USA und weiteren Ländern eingetragen sind. App Store und „Jede:r kann programmieren“ sind Dienstleistungsmarken von Apple Inc., die in den USA und weiteren Ländern eingetragen sind. Andere hier genannte Produkt- und Herstellernamen sind möglicherweise Marken der jeweiligen Unternehmen. Änderungen der Produktspezifikationen sind vorbehalten. Dieses Material dient ausschließlich zu Informationszwecken. Apple übernimmt keine Haftung hinsichtlich seiner Verwendung. September 2024