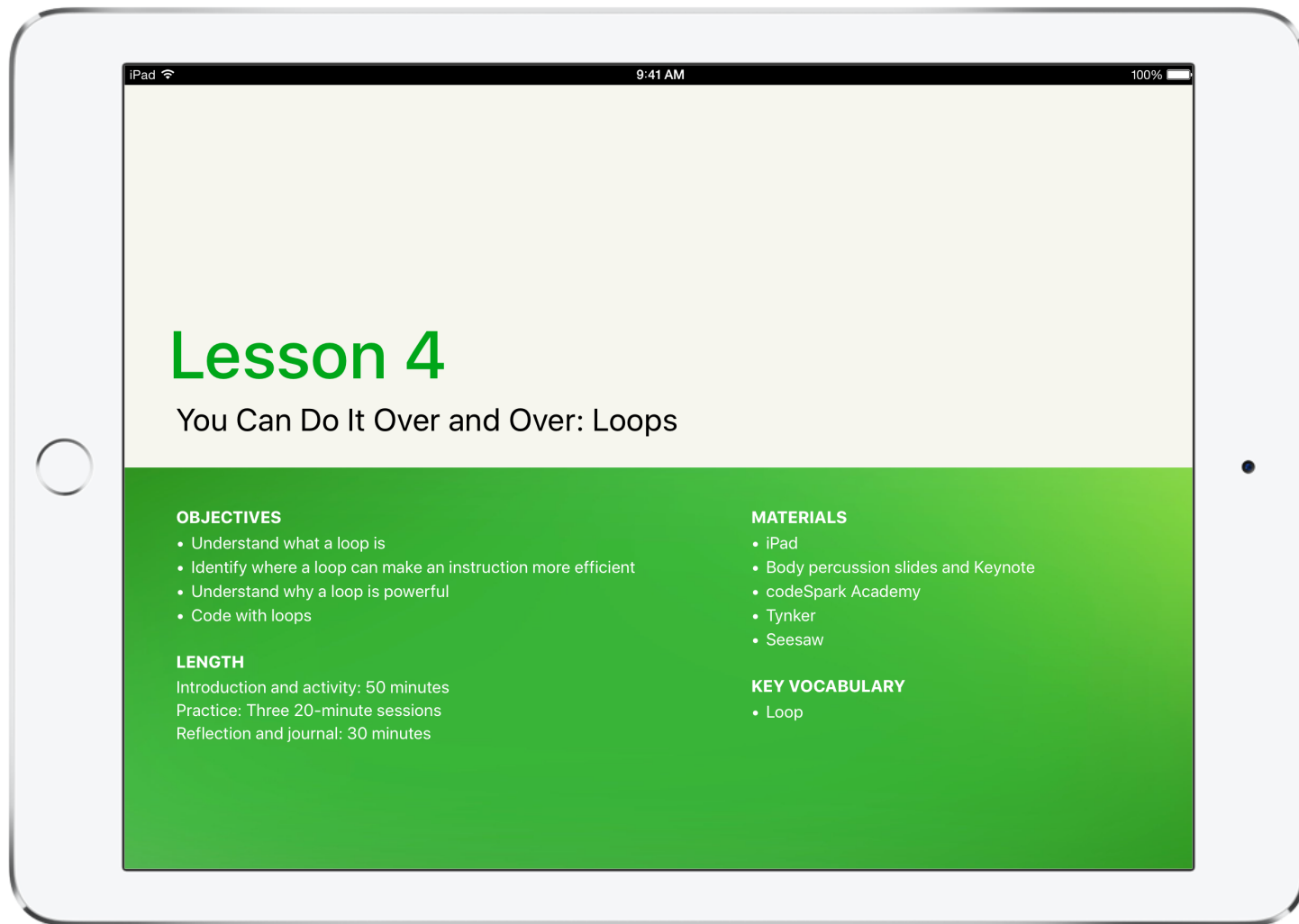




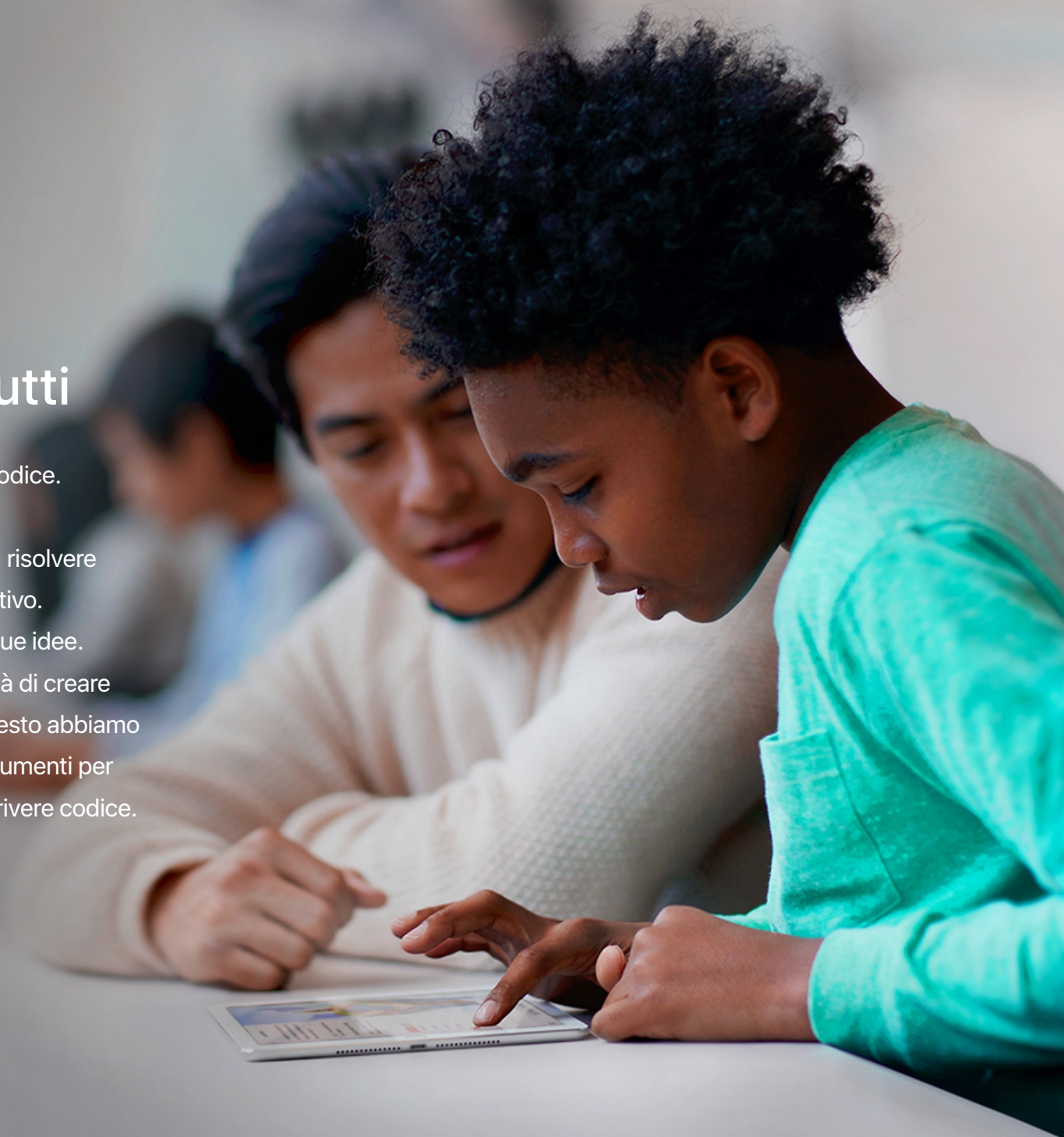
# Guida al corso "Get Started with Code"

Settembre 2017









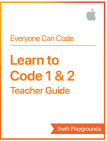


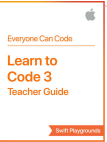








# Programmare è per tutti

La tecnologia ha un linguaggio che si chiama codice. Riteniamo che la capacità di programmare sia fondamentale. Saper programmare ti insegna a risolvere i problemi e a lavorare con gli altri in modo creativo. E ti permette di creare app che danno vita alle tue idee. Crediamo che ognuno debba avere la possibilità di creare qualcosa che possa cambiare il mondo. Per questo abbiamo ideato una nuova serie di corsi con risorse e strumenti per consentire a tutti di imparare, e insegnare, a scrivere codice.



# Corsi "Programmare è per tutti"

I corsi "Programmare è per tutti" contengono una serie di risorse per accompagnare gli studenti dal non avere alcuna esperienza di programmazione alla realizzazione della loro prima app. La tabella qui sotto offre una panoramica di tutte le risorse gratuite a supporto della didattica.

Progettazione	Dispositivo	Destinatari	App Store	Prerequisiti	Panoramica	Materiali didattici	Risorse di supporto	Ore di lezione
		Dalla scuola materna alla 2° elementare		Nessuno	Iniziare a pensare come programmatori attraverso attività pratiche per esplorare i concetti della programmazione grazie ad app con approccio visivo.	<ul style="list-style-type: none"> <li>Lezioni dell'app codeSpark Academy</li> <li>Corso Space Cadet di Tynker</li> </ul>	<ul style="list-style-type: none"> <li>Get Started with Code1 – Guida per gli insegnanti</li> </ul>	30 ore, considerando le lezioni della Guida per gli insegnanti e delle app
		Dalla 3° alla 5° elementare		Nessuno	Esplorare i concetti di base della programmazione ed esercitarsi a pensare come programmatori grazie ad app con approccio visivo.	<ul style="list-style-type: none"> <li>Corso Dragon Spells di Tynker</li> </ul>	<ul style="list-style-type: none"> <li>Get Started with Code 2 – Guida per gli insegnanti</li> </ul>	36 ore, considerando le lezioni della Guida per gli insegnanti e dell'app
		A partire dalla scuola media		Nessuno	Imparare i concetti di base della programmazione con codice Swift reale.	<ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Lezioni di "Learn to code 1 &amp; 2"</li> <li>Corso iTunes U</li> </ul>	<ul style="list-style-type: none"> <li>Learn to code 1 &amp; 2 – Guida per gli insegnanti</li> <li>Badge Swift Playgrounds dall'Apple Teacher Learning Center</li> </ul>	Fino a 85 ore, considerando le lezioni della Guida per gli insegnanti e di "Learn to code 1 & 2"
		A partire dalla scuola media		Learn to code 1 & 2	Ampliare le proprie abilità di programmazione per iniziare a pensare come veri sviluppatori di app.	<ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Lezioni di "Learn to code 3"</li> </ul>	<ul style="list-style-type: none"> <li>Learn to code 3 – Guida per gli insegnanti</li> </ul>	Fino a 45 ore, considerando le lezioni della Guida per gli insegnanti e di "Learn to code 3"
		Scuola superiore e università		Nessuno	Fare pratica con gli strumenti, le tecniche e i concetti necessari per sviluppare da zero un'app iOS di base	Ebook e file di progetto del corso "Intro to App Development with Swift"	<ul style="list-style-type: none"> <li>Intro to App Development with Swift – Teacher Guide</li> </ul>	90 ore
		Scuola superiore e università		Nessuno	Acquisire competenze di base su Swift, UIKit e le reti attraverso lezioni pratiche e progetti guidati. Alla fine del corso gli studenti saranno in grado di sviluppare un'app progettata da loro.	Ebook e file di progetto del corso "App Development with Swift"	<ul style="list-style-type: none"> <li>App Development with Swift – Teacher Guide</li> </ul>	180 ore

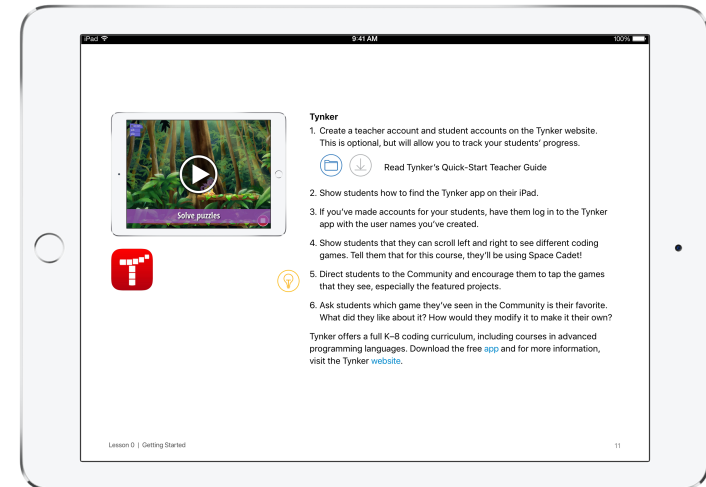
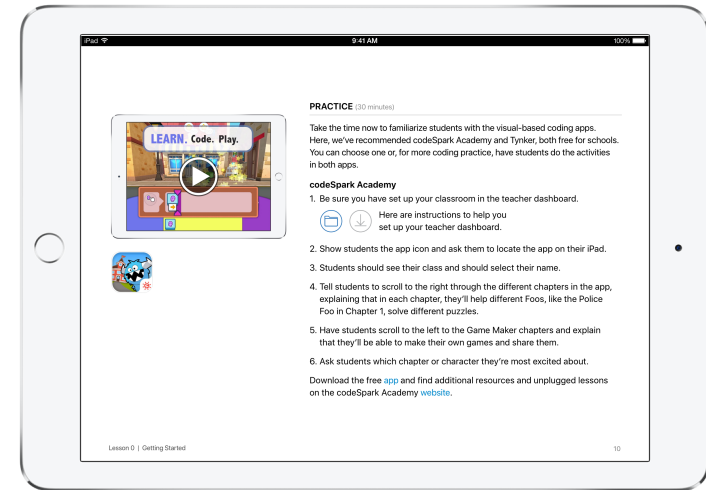
# Panoramica

I primi anni di scuola sono il momento ideale per introdurre concetti di programmazione come modo di pensare sia al mondo che ci circonda sia a quello digitale, e per sviluppare le competenze di base del pensiero computazionale. Le app progettate appositamente per gli studenti più giovani, come codeSpark Academy e Tynker, usano livelli di programmazione basati su immagini per sviluppare abilità di problem solving, incoraggiare la perseveranza e promuovere la creatività. L'app codeSpark Academy è pensata per gli alunni dai cinque ai sette anni. L'interfaccia è priva di parole, perciò possono giocarci tutti, anche chi non sa ancora leggere, chi sta imparando l'inglese e chi ha difficoltà di lettura. Con Tynker gli studenti dai cinque ai sette anni iniziano a sperimentare con i blocchi visivi per poi passare alla programmazione testuale man mano che completano i livelli e creano i propri progetti.

## In aula

Tynker, codeSpark Academy e le lezioni contenute nelle guide per gli insegnanti dei corsi "Get Started with Code" sono pensate per aiutarti a integrare la programmazione nei primi anni della scuola elementare. Le lezioni illustrano i concetti chiave del codice e dimostrano come questi rappresentino un modo di pensare applicabile anche ad altre aree disciplinari e alla vita di tutti i giorni.

Grazie alle guide per gli insegnanti puoi aiutare gli alunni quando non riescono a completare i livelli, indipendentemente dalla tua esperienza di programmazione. Le guide contengono anche attività di approfondimento, domande di riflessione, idee per tenere un diario e schede di valutazione per favorire la comprensione del materiale da parte degli studenti. Puoi tenere le lezioni in un unico blocco o per sezioni separate. Le tabelle delle corrispondenze nell'appendice forniscono un allineamento preliminare delle lezioni agli standard informatici provvisori per le scuole primarie e secondarie (K–12 Interim Computer Science Standards) della Computer Science Teachers Association (CSTA) per il livello 1.





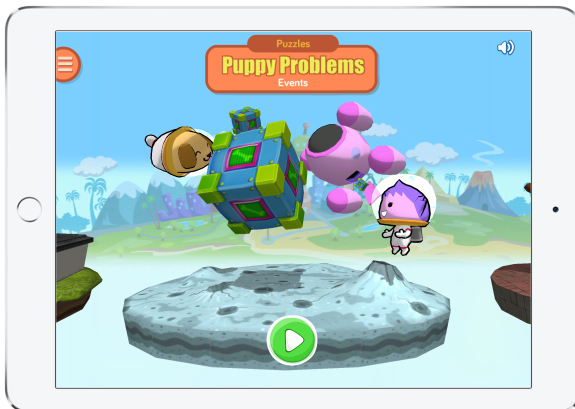
# Caratteristiche principali

## codeSpark Academy

**Approfondimenti.** L'interfaccia di questo gioco è priva di parole: completando i livelli si imparano concetti base dell'informatica come le sequenze, i cicli e le istruzioni condizionali.

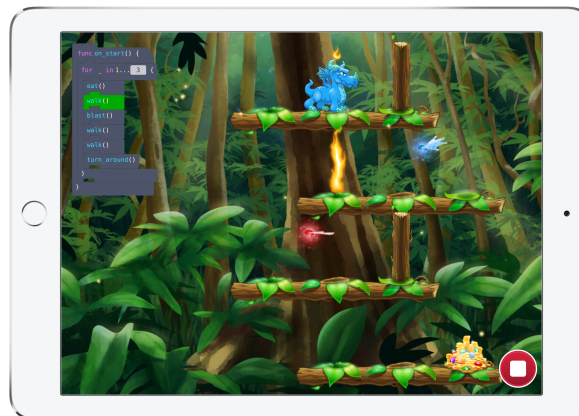
**Compilare.** I ragazzi applicano poi quanto appreso scrivendo il codice di propri progetti in Game Maker.

**In autonomia.** Per insegnare, imparare o giocare con questa app non bisogna saper già programmare. Gli insegnanti hanno gratuitamente a disposizione una serie di lezioni e dashboard in 10 lingue per seguire i progressi degli studenti.



## Tynker

**Ambiente di programmazione.** Gli studenti procedono al proprio ritmo attraverso giochi di programmazione con cui imparano e applicano i concetti in modo creativo.



**Valutazione automatica.** Una dashboard permette agli insegnanti di valutare le competenze dei ragazzi tramite livelli, quiz e analisi del codice.

**Modalità Swift.** Nel risolvere le varie sfide, gli studenti possono alternare i blocchi visivi con quelli di Swift. Così imparano a conoscere il linguaggio Swift e si preparano per il futuro.

## Guide per gli insegnanti "Get Started with Code"

**File da scaricare.** A supporto della didattica in classe si possono scaricare presentazioni Keynote e modelli per le attività degli studenti.

**Soluzioni.** Per aiutare i ragazzi quando si bloccano, hai a disposizione le soluzioni ai livelli di Tynker e codeSpark Academy.

**Esempi di lavori degli studenti.** Per spunti sulle attività.

**Riflessioni.** Questi suggerimenti e domande per le discussioni in classe ti aiutano a ripassare e rafforzare il collegamento tra l'applicazione del concetto dentro e fuori l'ambiente di programmazione.

**Consigli ed esempi.** Le guide contengono anche varie idee per approfondire o semplificare le lezioni.

**Attività di progettazione di app.** Queste lezioni guidano gli studenti attraverso il processo di design per concepire e realizzare il prototipo di un'app che risolva un problema della classe o della scuola.

# Descrizione dei corsi

## Get Started with Code 1

Esplorando i concetti della programmazione attraverso coinvolgenti attività pratiche e interattive legate a situazioni quotidiane, gli studenti iniziano a pensare come dei veri programmatori. Imparano cosa sono i comandi, le sequenze, i cicli, gli eventi e gli algoritmi. Collaborando gli uni con gli altri, si eserciteranno a prevedere il risultato del loro codice e a eseguire il debug dei loro programmi e di quelli dei compagni. Alleneranno le proprie capacità con app di programmazione visiva con cui completare livelli e progettare le proprie creazioni. E le attività facoltative guideranno gli studenti attraverso il processo di design per concepire e realizzare il prototipo di un'app che risolva un problema della classe o della scuola.

**Lezione 0 – Per cominciare.** Scopri cosa sa la classe sulle app e sulla programmazione, prepara il pannello di lavoro e presenta agli studenti le app che verranno usate durante le lezioni. Agli alunni vengono illustrati i vari ruoli all'interno di un team di progettazione di app.

**Lezione 1 – Puoi ordinarlo: introduzione alle sequenze.** Gli studenti esplorano sequenze della vita quotidiana, costruiscono una sequenza basata su una storia conosciuta e completano livelli in app di programmazione visiva utilizzando sequenze semplici. Gli alunni scoprono i vari scopi delle app.

**Lezione 2 – Puoi suddividerlo in passaggi: creare sequenze.** Esaminando l'importanza dell'ordine quando si mettono in sequenza le istruzioni, gli studenti scoprono che le stesse azioni si possono ordinare in modo diverso per creare sequenze diverse, poi creano la propria sequenza di ballo. Con le app di programmazione scoprono altri comandi e risolvono problemi ancora più complessi. Confrontano varie app che li aiutano a imparare cose nuove.

**Lezione 3 – Puoi sceglierlo: sequenze flessibili.** Gli studenti imparano che alcuni passaggi di una sequenza si possono ordinare in modo flessibile, e creano le proprie sequenze flessibili. Scoprono vari modi possibili per completare un livello e condividono con i compagni il codice che hanno usato. Esplorano le app che aiutano gli utenti a entrare in contatto con altre persone nella propria comunità.

**Lezione 4 – Puoi ripeterlo all'infinito: cicli.** Gli studenti individuano i cicli presenti in situazioni quotidiane ed esprimono la propria creatività sviluppando cicli di percussioni corporee. Osservano come i cicli sono rappresentati nella programmazione e li usano per snellire e semplificare il codice che scrivono. Imparano la progettazione dell'interfaccia utente e la sua importanza nel creare app divertenti e facili da usare.

**Lezione 5 – Puoi correggerlo: debugging.** Dopo aver compreso l'importanza della perseveranza e del debugging in vari contesti, gli studenti ripassano e usano le loro nuove capacità di programmazione per risolvere una sfida e fare il debug delle soluzioni dei compagni. Si esercitano a prevedere il risultato del loro codice e individuano i bug che gli impediscono di funzionare come prestabilito. Entrano nel vivo della progettazione di app con cui risolvere un problema.

**Lezione 6 – Puoi attivarlo: eventi e azioni.** Gli studenti scoprono in che modo gli eventi permettono alle loro app di funzionare in modo più coinvolgente e reattivo, e imparano a programmare con gli eventi. Esaminano come provochiamo gli eventi nella vita di tutti i giorni e costruiscono un telecomando per robot per esercitarsi a richiamare eventi. Iniziano a progettare le proprie app.

**Lezione 7 – Puoi, se segui la regola: istruzioni IF.** Gli studenti imparano le istruzioni condizionali e scoprono come riconoscere le istruzioni IF nella vita di tutti i giorni. Ragionano sulle istruzioni IF presenti come regole in molti giochi da tavolo, poi le applicano al proprio codice, rendendolo più reattivo ai condizionamenti dell'ambiente circostante. Usano i diagrammi di flusso per mostrare il funzionamento delle app.

**Lezione 8 – Puoi risolverlo: algoritmi.** Riunendo tutto quanto appreso finora, gli studenti elaborano algoritmi che prevedono una serie di passaggi per risolvere un problema. Partono da esercizi semplici, per poi arrivare a progettare e scrivere il codice per un labirinto. Realizzano i prototipi delle proprie app e completano un progetto finale per documentare il processo di design.

# Descrizione dei corsi (continua)

## Get Started with Code 2

In “Get Started with Code 2” esploreranno i concetti di base della programmazione e si eserciteranno a pensare come programmatori. Impareranno cosa sono gli algoritmi, le funzioni, i cicli, le istruzioni condizionali e le variabili. E scopriranno le basi della progettazione di un’interfaccia utente. Lavorando sia da soli sia in gruppi, consolideranno le proprie competenze risolvendo problemi reali di programmazione, testando il codice dei compagni e progettando programmi per una serie di robot. Inoltre si eserciteranno con Tynker per risolvere problemi di vario tipo e applicare i concetti appresi durante le lezioni. E le attività facoltative guideranno gli studenti attraverso il processo di design per concepire e realizzare il prototipo di un’app che risolva un problema della classe o della scuola.

**Lezione 0 – Per cominciare.** Scopri cosa sa la classe sulle app e sulla programmazione, fai conoscere agli studenti un’app di programmazione visiva come Tynker, e imposta i diari digitali dei ragazzi con un’app come Seesaw. Agli alunni viene presentata una sfida di progettazione di un’app.

**Lezione 1 – Pensare per passaggi: risolvere problemi con gli algoritmi.** Gli studenti scoprono gli algoritmi come serie di istruzioni per risolvere un problema o completare un’operazione. Nell’app Tynker, scelgono un uovo di drago da far schiudere, quindi creano algoritmi per completare i livelli esercitandosi con le sequenze. Durante le attività di classe elaborano e verificano gli algoritmi, e cominciano a fare brainstorming per un’app con cui risolvere un problema.

**Lezione 2 – Pensare per correzioni: debugging.** Gli studenti cercano e correggono gli errori nei loro algoritmi e nel codice. In Tynker modificano gli algoritmi contenenti bug per creare un programma corretto con cui completare i livelli. Imparano che ruolo hanno le tastiere all’interno delle app e come possono inserirle in quelle che stanno ideando.

**Lezione 3 – Pensare per cicli: cercare i cicli.** Dopo aver imparato che i loop sono strutture ripetute, gli studenti elaborano e verificano un algoritmo per creare un “serpente ripetente”. Usano i cicli per completare i livelli in Tynker individuando regolarità, e fanno brainstorming sui modi in cui la loro app potrebbe sfruttare la fotocamera e il microfono integrati.

**Lezione 4 – Pensare per parti: composizione e decomposizione.** Per elaborare un algoritmo con cui eseguire una “canzone del bicchiere”, gli studenti scompongono la routine nelle varie mosse. Risolvono problemi in Tynker scomponendoli in sottoproblemi più piccoli, e pensano ai modi in cui il touchscreen può rendere più interattiva la loro app.

**Lezione 5 – Pensare per serie: astrazione.** Gli studenti esplorano le analogie e le generalizzazioni suddividendo gli oggetti in categorie e spiegando la loro logica. In Tynker usano l’astrazione per individuare analogie tra diversi problemi e completando livelli sempre più complessi grazie a tutti i loro nuovi strumenti per scrivere codice. Pensano come usare tecnologie quali il Bluetooth per collegarsi ai dispositivi nelle vicinanze.

**Lezione 6 – Pensare per schemi: formare le funzioni.** Nel creare una routine per un robot, gli studenti la scompongono in funzioni, poi si scambiano algoritmi per testare i risultati attesi e reali. In Tynker usano le funzioni Name e Call per riutilizzare set di istruzioni e scrivere codice in modo più efficiente. Infine scoprono a quali tipi di dati la loro app può collegarsi grazie al GPS.

**Lezione 7 – Pensare per specifiche: istruzioni condizionali.** Gli studenti fanno un viaggio virtuale definendo la propria destinazione attraverso determinate condizioni utilizzando le istruzioni if. In Tynker usano le istruzioni if per gestire decisioni e alternative nei livelli. Fanno brainstorming per trovare modi innovativi per rendere uniche le loro app.

**Lezione 8 – Pensare per cicli: cicli while e cicli nidificati.** Gestendo un chiosco virtuale di ciambelle, gli studenti usano i cicli while e i cicli nidificati

# Descrizione dei corsi (continua)

per creare algoritmi per il robot delle ciambelle, che li userà per glassare ciambelle per tutti i clienti. In Tynker usano i cicli per abbreviare il proprio codice. Infine formano team di progettazione per iniziare a creare i prototipi delle loro app.

## **Lezione 9 – Pensare dentro e fuori dagli schemi: variabili, input e output.**

Usando le variabili per elaborare un algoritmo per una gara di poesia, gli studenti si esibiscono in una canzone o in un rap partendo dai suggerimenti dei compagni. In Tynker usano le variabili per completare livelli più complicati e applicando tutto quanto appreso finora sulla programmazione. Infine fanno delle interviste per definire gli utenti target per le proprie app.

**Lezione 10 – Pensare nella pratica: progettare la UI.** Gli studenti analizzano esempi di buona progettazione e creano un segnale per la scuola. Nell'attività con Tynker, usano tutte le competenze acquisite completando le lezioni del corso "Get Started with Code 2". Imparano che cos'è un'interfaccia e l'esperienza utente, e creano una tavola di stile per progettare la propria app. Nel progetto finale creano la presentazione per la loro app.



# Altre informazioni

## Scarica le risorse per i corsi "Get Started with Code"

- [Tynker](#)
- [codeSpark Academy](#)
- [Get Started with Code 1](#)
- [Get Started with Code 2](#)

## Scarica le risorse di Swift Playgrounds

- [Learn to code 1 & 2 - Corso iTunes U](#)
- [Learn to code 1 & 2 - Guida per gli insegnanti](#)
- [Learn to code 3 - Guida per gli insegnanti](#)
- [App Swift Playgrounds](#)

## Scarica le guide "App Development with Swift"

- [Intro to App Development with Swift](#)
- [Intro to App Development with Swift - Teacher Guide](#)
- [App Development with Swift](#)
- [App Development with Swift - Teacher Guide](#)

## Risorse aggiuntive

- Scopri di più sull'iniziativa [Programmare è per tutti](#).
- Scopri di più su [Swift](#).
- Scopri di più su [Xcode](#).
- Entra in contatto con altri educatori nei [forum degli sviluppatori Apple](#).
- Scopri di più su [codeSpark Academy](#).
- Scopri di più su [Tynker](#).

Le funzioni descritte possono subire modifiche. Alcune funzioni potrebbero non essere disponibili in tutte le lingue o in tutte le aree geografiche.

© 2017 Apple Inc. Tutti i diritti riservati. Apple, il logo Apple, iTunes U, Keynote e Xcode sono marchi di Apple Inc., registrati negli USA e in altri Paesi. Swift e Swift Playgrounds sono marchi di Apple Inc. Tutti gli altri prodotti e nomi di aziende citati potrebbero essere marchi registrati dei rispettivi proprietari. Le specifiche dei prodotti possono subire modifiche senza preavviso. Il presente materiale è fornito a puro titolo informativo; Apple non si assume alcuna responsabilità in merito al suo utilizzo. Settembre 2017