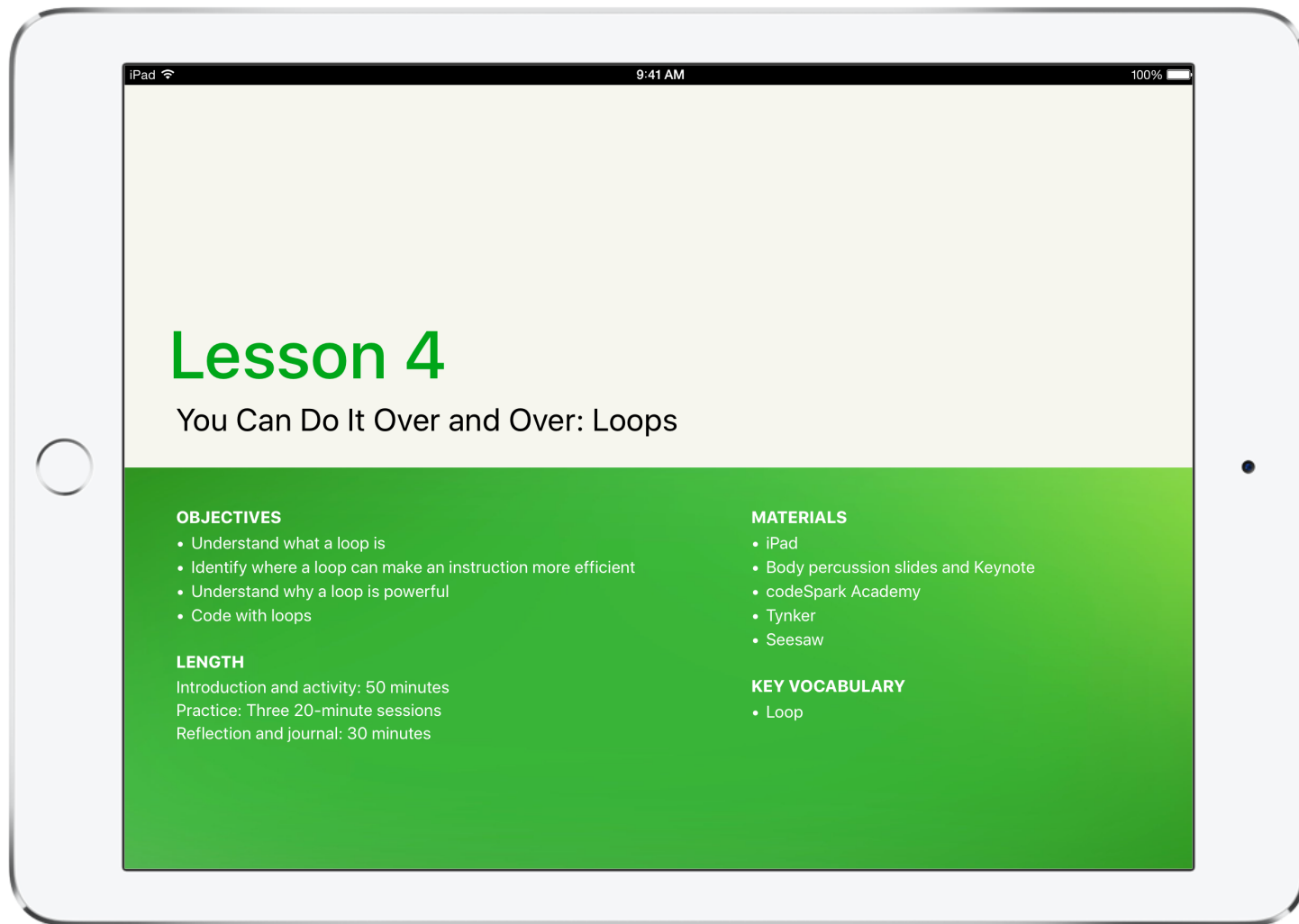




Kursplanshandledning till Get Started with Code

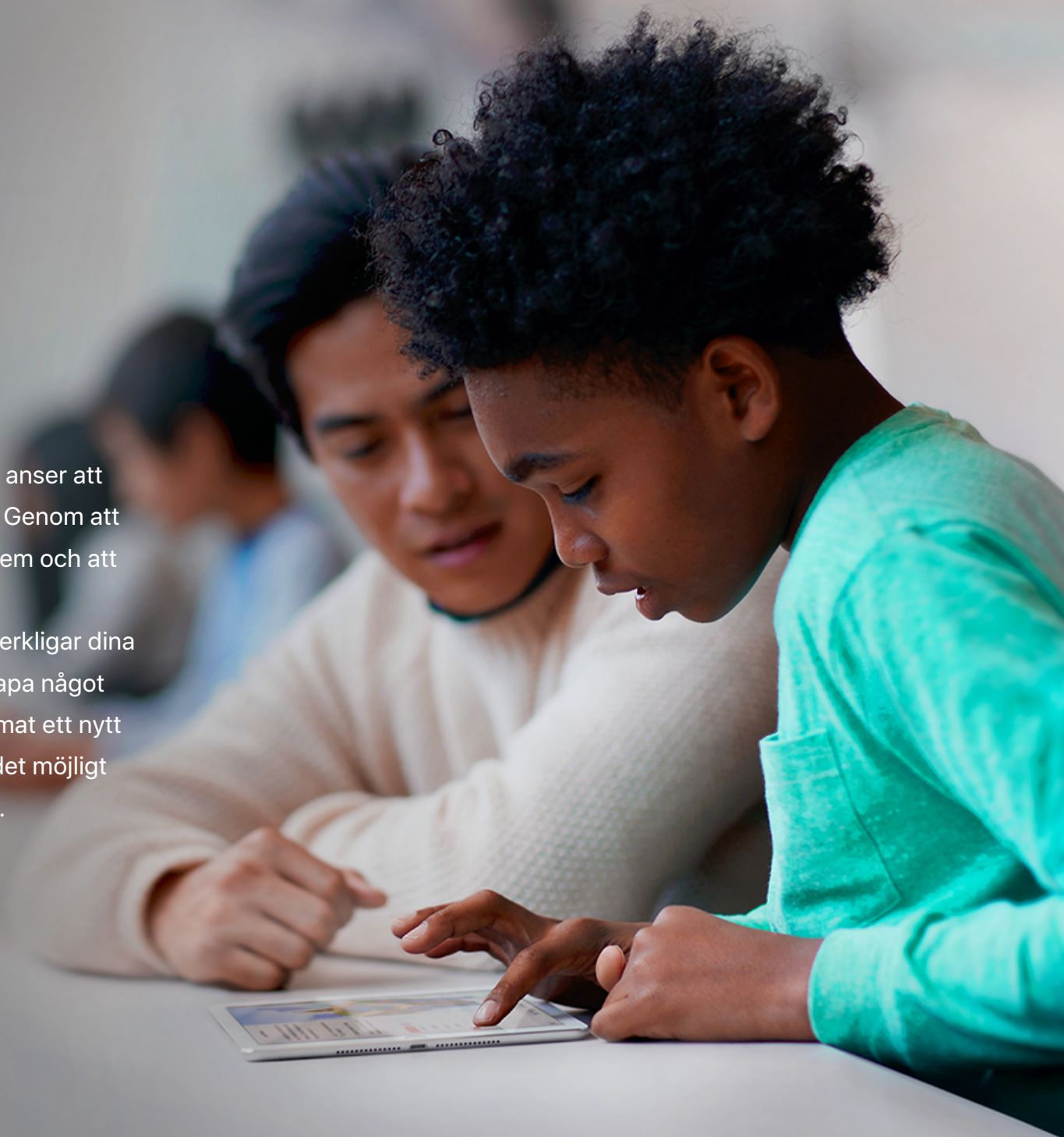
September 2017



Alla kan koda

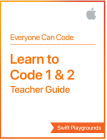
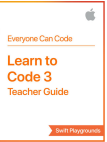
Tekniken har ett språk. Det kallas kod. Och vi anser att det är otroligt viktigt att kunna använda kod. Genom att lära sig programmera lär du dig att lösa problem och att samarbeta på kreativa sätt.

Och det hjälper dig att bygga appar som förverkligar dina idéer. Vi tycker att alla bör få möjlighet att skapa något som kan förändra världen. Därför har vi utformat ett nytt program med verktyg och resurser som gör det möjligt för alla att lära sig, skriva och undervisa i kod.



Kursplan för Alla kan koda

I programmet Alla kan koda ingår en rad resurser som hjälper elever utan erfarenhet av programmering hela vägen till att själva bygga sina första appar. Tabellen nedan visar en översikt av alla tillgängliga studie- och undervisningsresurser.

Kurs	Enhet	Målgrupp	App	Förkunskaper	Översikt	Utbildningsmaterial	Stödresurser	Beräknat antal lektionstimmar
 Get Started with Code 1 Teacher Guide		Förskola upp till årskurs 2		Inga	Börja tänka som programmerare och utforska programmeringskoncept genom praktiska övningar med hjälp av appar med visuell kod.	• Appktioner med codeSpark Academy • Kursen Tynker Space Cadet	• Get Started with Code 1: Teacher Guide	30 timmar som omfattar lärarhandledning och appktioner
 Get Started with Code 2 Teacher Guide		Årskurs 3–5		Inga	Utforska grundläggande programmeringskoncept och öva på att tänka som programmerare med hjälp av appar med visuell kod.	• Kursen Tynker Dragon Spells	• Get Started with Code 2: Teacher Guide	36 timmar som omfattar lärarhandledning och appktioner
 Learn to Code 1 & 2 Teacher Guide		Högstadiet och uppåt		Inga	Lär dig grundläggande programmeringskoncept med riktig Swift-kod.	• Appen Swift Playgrounds • Lektioner: Learn to Code 1 & 2 • iTunes U-kurs	• Learn to Code 1 & 2: Teacher Guide • Apple Teacher Learning Center – Swift Playgrounds-märken	Upp till 85 timmar som omfattar lärarhandledningen och lektionerna i Learn to Code 1 & 2
 Learn to Code 3 Teacher Guide		Högstadiet och uppåt		Learn to Code 1 & 2	Fördjupa programmeringskunskaperna och börja tänka mer som en apputvecklare.	• Appen Swift Playgrounds • Lektioner: Learn to Code 3	• Learn to Code 3: Teacher Guide	Upp till 45 timmar som omfattar Teacher Guide och lektionerna i Learn to Code 3
 Intro to App Development with Swift Xcode		Gymnasium och högskola		Inga	Få praktisk erfarenhet av de verktyg, tekniker och koncept som behövs för att bygga en enkel iOS-app från grunden.	• Boken Intro to App Development with Swift och projektfiler	• Intro to App Development with Swift: Teacher Guide	90 timmar
 App Development with Swift Xcode		Gymnasium och högskola		Inga	Skaffa grundkunskaper i Swift, UIKit och nätverk genom praktiska övningar och handledda projekt. Eleverna kan bygga en egen app som de själva designar i slutet av kursen.	• Boken App Development with Swift och projektfiler	• App Development with Swift: Teacher Guide	180 timmar

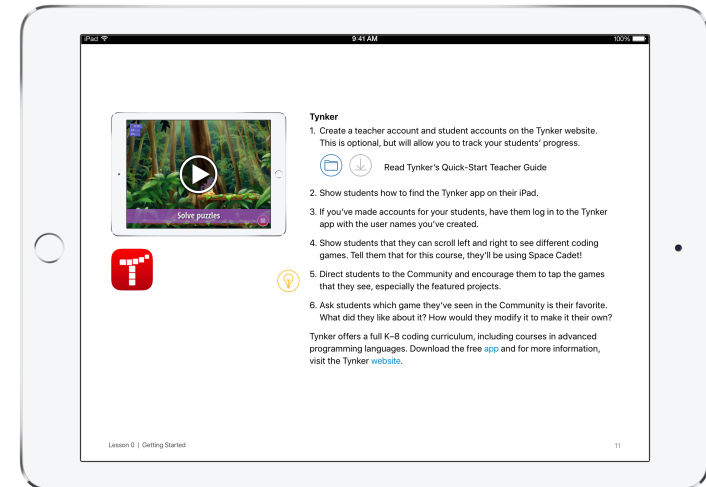
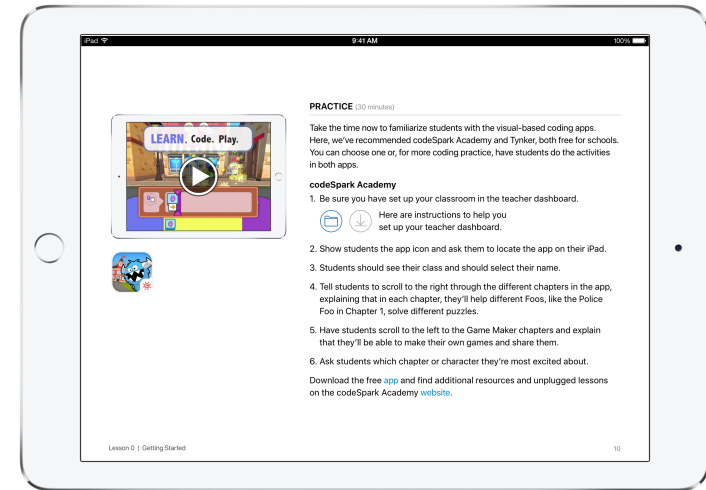
Översikt

Under de tidiga skolorn passar det utmärkt att introducera programmeringskoncept som ett sätt att tänka kring vanliga händelser i verkligheten och i den digitala världen samt utveckla grundläggande kunskaper i datoriserat tänkande. Appar som är särskilt utformade för yngre elever, till exempel codeSpark Academy och Tynker, använder visuella programmeringspussel till att utveckla problemlösningsförmågor, uppmuntra tålamod och främja kreativitet. codeSpark Academy är utformat för barn mellan 5 och 7 år. Det kan spelas av barn som ännu inte läser, andraspråkselever och elever med lässvårigheter eftersom spelet inte innehåller någon text. I Tynker kan barn mellan 5 och 11 år börja experimentera med visuella block och gå vidare med textbaserad programmering medan de löser pussel och bygger projekt.

I klassrummet

Tynker och codeSpark Academy tillsammans med lektionerna i lärarhandledningarna för Get Started with Code är utformade för att hjälpa dig att tidigt föra in programmering i undervisningen. Lektionerna behandlar viktiga programmeringskoncept och visar samtidigt att programmering är ett sätt att tänka som kan användas även i andra ämnen och i vardagen.

Lärarhandledningarna ger det stöd du behöver för att hjälpa dina elever att lösa programmeringspussel, oavsett vilken erfarenhet du har av programmering. De består bland annat av extraaktiviteter, appdesignaktiviteter, reflektionsfrågor, loggboksanvisningar, en betygsskala och annat för att hjälpa dig att fördjupa elevernas förståelse av materialet. Du kan undervisa hela lektioner eller dela in dem i mindre delar. I bilagorna finns det översikter som ger en preliminär anpassning av lektionerna till CSTA:s datavetenskapliga delstandarder på nivå 1 för grundskole- och gymnasieelever.



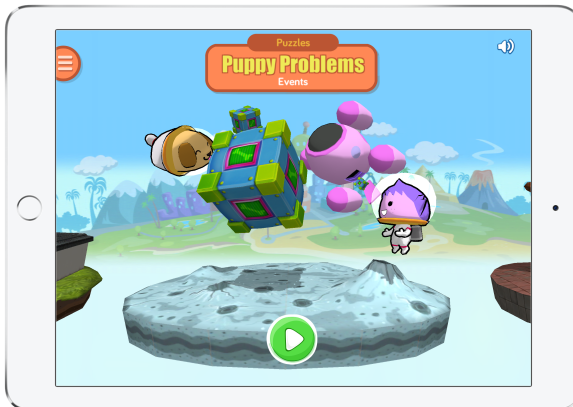
Viktiga funktioner

codeSpark Academy

Lära. Eleverna löser pussel i det här spelet som är helt utan text för att lära sig grundläggande koncept inom datavetenskap, till exempel sekvenser, loopar, villkorliga uttryck med mera.

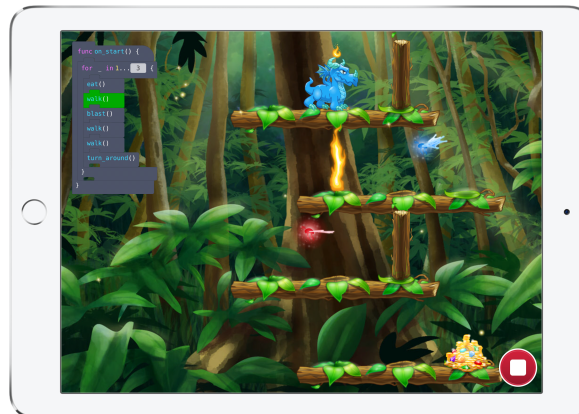
Utveckla. Eleverna använder sedan sin kunskap genom att programmera sina egna projekt i Game Maker.

Självständigt arbete. Det behövs ingen tidigare programmeringserfarenhet för att undervisa, lära sig eller spela. Kursplanen och en lärarpanel för framstegsrapporter finns att hämta kostnadsfritt



Tynker

Programmeringsmiljö. Eleverna går fram i egen takt genom strukturerade programmeringspussel för att lära sig koncept och använda dem på ett kreativt sätt.



Automatiska utvärderingar. En lärarpanel utvärderar elevernas utveckling med pussel, frågor och kodanalyser.

Swift-funktion. Eleverna kan växla mellan ett visuellt läge och Swift-läge när de löser pussel så att de bekantar sig med Swift och får övning inför programmering i framtiden.

Lärarhandledningar för Get Started

Nedladdningsbara filer. Mallar för elevaktiviteter och Keynote-presentationer ger stöd åt elevernas lärande i klassrumsundervisningen.

Facit. Med pusselfacit till Tynker och codeSpark Academy kan du enkelt hjälpa eleverna när de fastnar.

Exempel på elevarbeten. Se exempel på aktiviteter.

Reflektioner. Dessa frågor och anvisningar för diskussion i klassen hjälper dig att utvärdera och stärka kopplingen mellan att använda programmeringskoncept både i och utanför kodmiljön.

Tips och exempel. Handledningen innehåller många förslag på att utöka eller förenkla lektioner.

Appdesignaktiviteter. De här lektionerna vägleder eleverna genom en designprocess där de skapar koncept och prototyper till en appidé som löser ett problem i klassen eller på skolan.

Kursöversikter

Get Started with Code 1

Eleverna börjar tänka som programmerare när de utforskar programmeringskoncept genom interaktiva och praktiska övningar kring vanliga händelser. De får lära sig om kommandon, sekvenser, loopar, händelser och algoritmer. I grupparbeten övar eleverna på att förutse följderna av deras koder och att felsöka både sin egen och andras koder. De får dessutom öva på att använda sina kunskaper i appar med visuell kod där de kan lösa pussel och bygga sina egna skapelser. Frivilliga designaktiviteter vägleder eleverna genom en designprocess där de skapar koncept och prototyper till en appidé som löser ett problem i klassen eller på skolan.

Lektion 0—Getting Started. Ta reda på elevernas förkunskaper vad gäller appar och programmering, förbered väggen för elevarbete och introducera eleverna för viktiga appar som de kommer att stöta på i undervisningen. Eleverna introduceras till olika roller i ett appdesignsteam.

Lektion 1—You Can Order It: Introduction to Sequencing. Eleverna utforskar vanliga sekvenser, bygger sekvenser baserat på en berättelse de känner till och löser pussel i appar med visuell programmering med hjälp av enkla sekvenser. Eleverna utforskar de olika syften som appar fyller.

Lektion 2—You Can Step It: Creating Sequences Eleverna skapar en egen rolig dans medan de utforskar vikten av kodens ordning när de gör instruktionssekvenser och lär sig att ändra ordningen för att göra en annan sekvens. De lär sig fler kommandon och att lösa mer komplexa problem i programmeringsapparna. Eleverna jämför olika appar som hjälper användare att lära sig nya idéer.

Lektion 3—You Can Choose It: Flexible Sequencing. Eleverna lär sig att en del steg i sekvenserna kan läggas i en flexibel ordning och skapar sina egna flexibla sekvenser. De utforskar olika sätt att lösa pussel och delar sina lösningar med varandra. Eleverna utforskar olika appar som hjälper användarna att få kontakt med andra människor i deras närhet.

Lektion 4—You Can Do It Over and Over: Loops. Eleverna identifierar loopar i vanliga sammanhang och bygger sina egna loopar med kroppsslagverk. De utforskar hur loopar representeras i programmering och använder sig av dem för att förenkla koden. Eleverna lär sig om design av användargränssnitt och hur det kan göra en app kul och enkel att använda.

Lektion 5—You Can Fix It: Debugging. Eleverna lär sig vikten av tålamod och att felsöka i olika sammanhang. De utvärderar och använder sina nya programmeringskunskaper till att lösa en utmaning och felsöka andra elevers lösningar. De övar på att förutse kodernas följder och att identifiera fel när koden inte fungerar som den var tänkt. Eleverna ger sig i kast med att designa appar för att lösa ett problem.

Lektion 6—You Can Prompt It: Events and Actions. Eleverna lär sig att programmera med händelser och utforskar hur det gör uppspelning och programmering av appen mer fängslande och dynamisk. De får fundera på hur vi framkallar händelser i vardagen och öva på det genom att skapa en robotfjärrkontroll. Eleverna börjar designa sina egna appar.

Lektion 7—You Can, If You Follow the Rule: IF Statements. Eleverna introduceras för villkorliga uttryck och lär sig känna igen if-uttryck i vardagen. De utgår från if-uttryck som utgör regler i sällskapsspel. De programmerar sedan med hjälp av if-uttryck, vilket gör att deras kod svarar bättre på olika omständigheter i miljön. Eleverna använder flödesdiagram för att visa hur deras appar fungerar.

Lektion 8—You Can Solve It: Algorithms. Eleverna använder sig av all kunskap de har fått hittills och skapar algoritmer som omfattar ett antal steg för att lösa ett problem. De börjar med enkla beskrivningar och fortsätter med att skapa och programmera sina egna labyrintspel. Eleverna skapar prototyper av sina appar och slutför ett avslutande projekt som dokumenterar deras appdesignprocess.

Kursöversikter (fortsättning)

Get Started with Code 2

I Get Started with Code 2 utforskar eleverna grundläggande programmeringskoncept och övar på att tänka som programmerare. Samtidigt som de lär sig om algoritmer, funktioner, loopar, villkorliga uttryck och variabler upptäcker de grunderna till att designa användargränssnitt. Eleverna kommer att arbeta både självständigt och i grupp medan de fördjupar sina programmeringskunskaper genom att lösa riktiga programmeringsproblem, testa varandras kod och designa program till olika robotar. De får också öva på dessa kunskaper i Tynker där de löser olika problem och använder sig av koncepten de har lärt sig från aktiviteterna i undervisningen. Frivilliga designaktiviteter vägleder eleverna genom en designprocess där de skapar koncept och prototyper till en appidé som löser ett problem i klassen eller på skolan.

Lektion 0 – Getting Started. Ta reda på vad eleverna redan kan om appar och programmering, se till att de bekantar sig med appar med visuell programmering, till exempel Tynker, och ställ in elevernas digitala loggböcker med hjälp av en app, till exempel Seesaw. Eleverna introduceras för en appdesignutmaning.

Lektion 1 – Think in Steps: Solving Problems with Algorithms.

Eleverna bekantar sig med algoritmer som en rad instruktioner för att lösa ett problem eller utföra en uppgift. I appen Tynker får eleverna välja en drake och sedan bygga algoritmer för att lösa pussel samtidigt som de lär sig att använda sekvenser effektivt. Eleverna designar och testar algoritmer genom aktiviteter i klassrummet. Eleverna börjar brainstorma fram appar som kan hjälpa till att lösa ett problem.

Lektion 2 – Think in Fixes: Debugging. Eleverna utforskar sina upptäckter och löser problem med algoritmer och koder. De ändrar algoritmer som innehåller fel i Tynker så att de kan skapa ett korrekt program för att lösa

pusslen. Eleverna lär sig om den roll tangentbord spelar i appar och hur de kan använda dessa i sina egna appidéer.

Lektion 3 – Think in Circles: Looking for Loops När eleverna har fått loopar presenterade som ett upprepande mönster får de designa och testa en algoritm för att skapa en Loopy Snake. De använder loopar i Tynker till att lösa pussel genom att upptäcka mönster. Eleverna brainstormar kring hur deras appar kan dra nytta av den inbyggda kameran och mikrofonen.

Lektion 4 – Think in Bits: Composition and Decomposition. Eleverna bryter ned en rutin i delrörelser så att de kan tänka ut en algoritm för att spela upp en muggsång. De löser problem i Tynker genom att bryta ned dem i mindre delproblem. Eleverna funderar på hur pekskärmen kan göra deras app mer interaktiv.

Lektion 5 – Think in Sets: Abstraction. Eleverna utforskar likheter och generaliseringar genom att dela in objekt i olika kategorier och sedan motivera sina val. De använder abstraktion i Tynker till att upptäcka likheter mellan problem och löser alltmer komplexa pussel med alla nya programmeringsverktyg. Eleverna funderar på hur de kan använda verktyg som Bluetooth för att ansluta till enheter i närheten.

Lektion 6 – Think in Patterns: Forming Functions. Eleverna delar in rutiner i funktioner när de skapar kommandon till en robot och byter sedan ut algoritmer för att testa både förutsedda och verkliga resultat. Eleverna använder namn- och anropsfunktioner i Tynker när de återanvänder uppsättningar av instruktioner för att programmera effektivare. Eleverna lär sig om de datatyper som deras app kan ansluta till via gps.

Lektion 7 – Think in Specifics: Conditional Statements. Eleverna ger sig ut på ett virtuellt äventyr där de definierar sitt mål med en rad kvalificerade villkor med hjälp av if-uttryck. De använder if-uttryck till att hantera beslut och alternativ i pusslen i Tynker. Eleverna brainstormar fram innovativa sätt att göra sina appar unika.

Kursöversikter (fortsättning)

Lektion 8 – Think in Cycles: While Loops and Nested Loops.

Eleverna driver en virtuell munkkiosk och använder while-loopar och nästlade loopar till att skapa algoritmer som får munkroboten att glasera tillräckligt många munkar till kunderna. Eleverna använder loopar i Tynker till att korta ner koden. Eleverna skapar appdesignsteam och börjar att skapa en prototyp för en egen app.

Lektion 9 – Think In and Outside the Box: Variables, Input, and Output.

Eleverna designar en algoritm med olika variabler för att delta i en jamsession med poesi, där elever framför en sång eller rap baserat på publikens förslag. De använder variabler i Tynker för att klara mer komplexa pussel och använder alla programmeringskunskaper de har lärt sig hittills. Eleverna genomför intervjuer med användare så att de kan rikta in sig på en målgrupp för sin app.

Lektion 10 – Think in Practice: Design UI. Eleverna analyserar vad en lyckad design innebär och skapar en logga till sin skola. Genom aktiviteter i Tynker använder de sig av alla sina nya kunskaper och slutför Get Started with Code 2-lektionerna. Eleverna lär sig om användargränssnittet och användarupplevelsen och skapar en moodboard för sin appdesign. De skapar en presentation av sin app i det avslutande projektet.

Ytterligare information

Ladda ner Get Started with Code-resurserna

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

Ladda ner Swift Playgrounds-resurserna

- [Learn to Code 1 & 2: iTunes U-kurs](#)
- [Learn to Code 1 & 2: Teacher Guide](#)
- [Learn to Code 3: Teacher Guide](#)
- [Appen Swift Playgrounds](#)

Ladda ner handledningarna till 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](#)

Ytterligare resurser

- Läs mer om programmet [Alla kan koda](#).
- Läs mer om [Swift](#).
- Läs mer om [Xcode](#).
- Få kontakt med andra utbildare på [Apple Developer-forum](#).
- Läs mer om [codeSpark Academy](#).
- Läs mer om [Tynker](#).

Funktionerna kan ändras. Vissa funktioner är inte tillgängliga i alla länder eller på alla språk.

© 2017 Apple Inc. Alla rättigheter förbehålls. Apple, Apples logotyp, iTunes U, Keynote och Xcode är varumärken som tillhör Apple Inc. och är registrerade i USA och andra länder. Swift och Swift Playgrounds är varumärken som tillhör Apple Inc. Övriga namn på produkter och företag som nämns kan varumärken som tillhör sina respektive företag. Produktspecifikationer kan ändras utan föregående meddelande. Detta material tillhandahålls endast i informationssyfte. Apple åtar sig inget ansvar för användning av det. September 2017