Edutopia
7 Apps for Teaching Children Coding Skills

October 10, 2013

Image credit: Veer

Editor's Note: *Helen Mowers, co-creator of the Tech Chicks podcast, contributed to this post.*

It's hard to imagine a single career that doesn't have a need for someone who can code. Everything that "just works" has some type of code that makes it run. Coding (a.k.a. programming) is all around us. That's why all the cool kids are coding . . . or should be. Programming is not just the province of pale twenty-somethings in skinny jeans, hunched over three monitors, swigging Red Bull. Not any more! The newest pint-sized coders have just begun elementary school.

If you're concerned that that a) elementary school students don't have the ability to code, b) there's no room in the curriculum, and c) you don't possess coding chops to teach programming skills, throw out those worries. The following sites and apps can help anyone who has basic reading skills grasp the basics of *thinking and planning in order to make things happen* (the whole purpose of coding) and create applications: interactive games, quizzes, animations, etc. Best of all, many of these tools are free, or almost free, and require no coding background or expertise!

In no particular order, we have listed all the coding apps that are appropriate for young learners. We've used many of them with elementary-aged students.

GameStar Mechanic

Platform: Web

Cost: $2 per student

GameStar Mechanic teaches kids, ages 7-14, to design their own video games. Your students will love completing different self-paced quests while learning to *build* game levels. The site integrates critical thinking and problem-solving tasks. An app embedded within Edmodo makes logins easy for students.

Scratch

Platform: Web

Cost: Free!

Designed by MIT students and staff in 2003, Scratch is one of the first programming languages we've seen that is created specifically for 8-to-16-year-olds. Originally a multi-platform download, Scratch is now web-based and more accessible. Students use a visual programming language made up of bricks that they drag to the workspace to animate sprites. Various types of bricks trigger loops, create variables, initiate interactivity, play sounds, and more. Teaching guides, communities and other resources available on the website will help instructors get started. You don't have to be a programming expert to introduce Scratch -- we learned right along with the students!

Tynker

Platform: Web

Cost: Free! (with Premium upgrade option)

Although Tynker is relatively new, we definitely count it as one of our favorite coding apps. The interface looks similar to Scratch. But while Scratch was designed to program, Tynker was built to *teach* programming. The app features starter lesson plans, classroom management tools, and an online showcase of student-created programs. Lessons are self-paced and simple for students to follow without assistance.

Move the Turtle

Platform: iOS (iPad and iPod)

Cost: $2.99

We love Move the Turtle, a gamified way to learn programming procedures. The main character reminds us of the old Logo turtle used to teach kids computer programming during the reign of the Apple IIe. Each new level of achievement increases in difficulty and teaches a new command that directs the turtle to reach a star, make a sound, draw a line, etc. A free play "compose" mode lets students move the turtle however they want.

Hopscotch

Platform: iPad

Cost: Free!

Hopscotch looks a lot like Scratch and Tynker and uses similar controls to drag blocks into a workspace, but it only runs on the iPad. The controls and characters are not as extensive as Scratch and Tynker, but Hopscotch is a great tool to begin helping students without coding experience learn the basics of programming, logical thinking and problem solving.

Daisy the Dinosaur

Platform: iPad

Cost: Free!

From the makers of Hopscotch, Daisy targets the youngest coders. The interface is similar to Hopscotch but much simpler. There is only a dinosaur to move and only basic functions to use, but for your younger students, this is an excellent introduction to programming.

Cargo-Bot

Platform: iPad

Cost: Free!

Cargo-Bot is another game that teaches coding skills. On each level, the objective is to move colored crates from one place to another by programming a claw crane to move left or right, and drop or pick up. The game was actually programmed on an iPad, using a touch-based coding app called Codea, which is based on the programming language Lua. Elementary students will learn the logical thinking required to eventually do "real" text-based programming using Lua -- but Lua is not for young learners. For elementary students, stick with Cargo-Bot.

We hope these descriptions have whetted your appetite and that you'll incorporate coding into your curriculum. Even if a student never intends to pursue programming as a career, learning to code will still foster problem-solving skills, spark creativity and enhance logical thinking. Code.org, a pro-coding education nonprofit, features dozens of quotations about computer programming from famous and important people who believe that coding should be part of the core curriculum for every child. One of our favorite quotes is by Maria Klawe, a computer programmer and inventor who says, "Coding is today's language of creativity. All our children deserve a chance to become *creators* instead of *consumers* of computer science [emphasis added]."

How have you brought coding into your classroom? Do you have any resources to share?