CLIVE THOMPSON

We're All Coders Now

Writing software shouldn't be just for engineers—it can be as easy as ABC.

HOW DO YOU STOP people from texting while driving? Last spring, Daniel Finnegan had an idea. He realized that one of the reasons people type messages while they're in the car is that they don't want to be rude—they want to respond quickly so friends don't think they're being ignored. I So what if the phone knew you were driving—and responded on its own? I Normally, Finnegan wouldn't have been able to do anything with his insight. He was a creative-writing major at the University of San Francisco, not a programmer. But he'd enrolled in a class where students were learning to use Google's App Inventor, a tool that makes it pretty easy to hack together simple applications for Android phones by fitting bits of code together like Lego bricks. I Finnegan set to work, and within a month he'd created an app called No Text While Driving. When you get into your car, you hit a button on the app and it autoresponds to incoming texts with "I'm driving right now, I'll contact you shortly." I've used the app, and it's terrific: By getting you off the hook socially, it makes your driving safer. It ought to be available—mandatory, even—on every phone. I Finnegan's story illustrates a powerful point: It's time for computer programming to be democratized. I Software, after all, affects almost everything we do. Pick any major problem—global warming, health care, or, in Finnegan's case, highway safety—and clever software is part of the solution. Yet only a tiny chunk of people ever consider learning to write code, which means we're not tapping the creativity of a big chunk of society. I Serious leaders already know this. "Every time I talk to generals in the military, they talk about how they can't find enough young people who know how to program," says Douglas Rushkoff, author of *Program or Be Programmed*, a new book that argues that everyday people should learn to code.

What's more, knowing programming changes your worldview. "You learn that every problem is made up of smaller problems," says Kevin Lawver, a web designer whose 11-year-old son, Max, has spent the past few years designing programs using kid-friendly languages like Scratch. Frankly, companies like Facebook and Google would probably face a lot tougher scrutiny if their users understood how software works. Facebook users would know it's not that hard to program finely grained controls over who sees what on Facebook (a service that is, as computer scientist Eben Moglen semijokingly puts it, just "some PHP doodads"). The current mystique around software allows companies to claim that the way they're doing things is the only way possible, when it isn't.

But isn't programming inherently hard? Sure. So are lots of things. Hell, cooking dinner involves lethal implements, a fire inside your house, and ingredients (like raw chicken) that can poison you if they're not correctly prepared. We teach kids how to do that safely; we can do the same with programming.

It'd be great if programming became part of the curriculum, but that probably won't happen, given how slowly schools change. The good news is that—much as the "maker" set is relearning how to build stuff—a grassroots movement is creating tools that let even liberal arts majors hack together a program. In recent years, we've seen the release of oodles of languages designed to make it easy for kids (or adults!) to write code, from Processing to Scratch to Google's App Inventor. In fact, I just used App Inventor to make a program that lets my toddler and kindergartner call family members by touching their pictures.

Got a problem you need to solve? When you can program it yourself, there's always an app for that.

EMAIL *clive@clivethompson.net.*