CONVERSATION CASE

Kids as Coders

Introduction

What is "coding," and is it a skill set that kids need today? Coding refers to the language of computer programming, and it's becoming an increasingly important topic in education, especially when considering our global economy. Many adults think of coding as an activity only done by a small group of the most mathematically savvy people. But today, coding takes many forms, and coders come in all ages and sizes.

For parents, the first question is whether coding ties into students' academic goals/academic potential/ academic careers. What are the long-term benefits of learning to code? How does that skill set apply outside of the field of computer programming? The second question is *how* kids can learn to code. When is the optimal time to learn to code — elementary school? How is it integrated into an already-packed school day agenda? Today's case is designed to help you consider the issues and opportunities related to teaching kids to code. Take a minute before you dive in to reflect on your current ideas about what "coding" means to your child as a member of today's 21st-century global economy and community.

In Part 1, you will read a case that highlights several different perspectives on teaching kids to code. Several of the excerpts come from responses to the *New York Times*' Room for Debate question:

"Should coding be part of the elementary school curriculum?"[1]

In Part 2, you will watch part of MIT Professor Mitch Resnick's TED Talk about the benefits of teaching kids to code. In it, he discusses Scratch — a project that his team at the MIT Media Lab designed to do just that.

The Case: Part 1

Responding to the question "Should coding be part of the elementary school curriculum?" seventh grader Becky Button responds affirmatively and argues that coding is critical. Button writes,

> Understanding computer code is an important part of what makes us literate in today's technology. Coding involves dividing up a task into its most basic pieces and then putting them together in a logical order. This analytical thinking process has lots of real world applications ... My younger brother, who is in the third grade, already codes. He can do amazing things with code like make his robot do tasks and create his own video games. For a recent social studies project, he used Scratch, a programming language, to make an animated video of the life of George Washington ... **Code is everywhere now.** It's in your phone, it's in your kitchen appliances, it's in your car, it's in the cards we carry in our wallets and it's how we communicate. Learning to code needs to be in the schools. **If kids learn to code, they will understand the language that is shaping their future.**

Hadi Partovi is the founder of code.org, a nonprofit website designed to make learning how to code accessible. Partovi agrees with Button. He argues,

> We teach elementary-school students long division or how weather works because these are relevant, foundational concepts. At a time when most first graders can already navigate through websites and apps,

> why aren't we teaching them how the Internet works or how to program a computer? By high school, it can be too late. In 2013, not one female student took the Advanced Placement Computer Science exam in Mississippi or in Montana. In 11 states, not one black student took the test. In eight states, no Hispanic students. **To make computer science opportunities accessible to all students, we need to start in elementary school, where the playing field is still relatively level.** Students learn fast at a young age, before stereotypes suggest coding is too difficult, just for nerds, or just for boys ... Every student should have a fair chance to take part in building the technology that will change our world.

But not everyone agrees. John Dvorak is a technology writer, and for him, the writing is on the wall: Young kids do not need to learn how to code. In fact, he's convinced that the push to teach kids coding is all a scam. Dvorak writes,

> A child should be developing basic human skills. So called "computational thinking" is not one of them. **A second grader should be running around, throwing a ball, scratching out drawings, learning fine motor skills and developing normal interpersonal social skills.**

In another article from *Wired* magazine[2], Jathan Sadowski argues that pushing people to code will actually **widen the gap between the rich and the poor.** He writes,

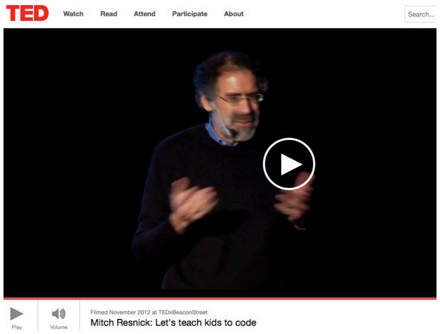
> … is it wrong to teach a person to code? No. I don't deny that coding is a useful skill to have in a modern ubiquitous computing society. It can help people personalize and understand the devices and services they use on a daily basis. It's also good news that methods for teaching kids how to code are improving and becoming more effective, or that kids can ostensibly learn on their own when left to their own devices. **The problem is elevating coding to the level of a required or necessary ability. I believe that is a recipe for further technologically induced stratification.** Before jumping on the everybody-must-code bandwagon, we have to look at the larger, societal effects — or else risk running headlong into an even wider inequality gap.
>
> For instance, the burden of adding coding to curricula ignores the fact that the English literacy rate in America is still abysmal: 45 million U.S. adults are "functionally illiterate" and "read below a 5th grade level," according to data gathered by the Literacy Project Foundation. Almost *half* of all Americans read "so poorly that they are unable to perform simple tasks such as reading prescription drug labels." The reading proficiency of Americans is much lower than most other developed countries, and it's declining.
>
> **We have enough trouble raising English literacy rates, let alone increasing basic computer literacy:** the ability to effectively use computers to, say, access programs or log onto the internet. Throwing coding literacy into the mix means further divvying up scarce resources. Teaching code is expensive. It requires more computers and trained teachers, which many cash-strapped schools don't have the luxury of providing. As software engineer Chase Felker has argued, "I'm not sure it's even possible to teach everyone how to code, but I do know that to mandate programming as a general education requirement would displace something else that we're already failing to teach, and that's not good, either."
>
> [...] But a world where coding dictates the future is not inevitable. Instead of making people adapt to technologies — in the process leaving behind large swaths of society — technologies should adapt to *our* needs and values. As the media theorist Marshall McLuhan said, "There is absolutely no inevitability as long as there is a willingness to contemplate what is happening." There's still plenty of room for our contemplation — and deeper consideration — of how to advance our future without leaving anyone behind."

Consider

- Which perspective do you relate to most closely and why?
- What do you see as the benefits in teaching kids to code?
- How do you see this skill set relating to the needs of today's 21st-century workforce?
- Does coding have a place in school? Or is it best addressed in an outside-of-school setting?
- The debate references an inequality gap. How might we counter such an issue?

The Case: Part 2

Scratch is a programming language and community born out of MIT Media Lab's Lifelong Kindergarten Group. This project is designed to help kids (and adults!) learn how to code and program, and it's free! Watch a segment from Professor Mitch Resnick's TED Talk "Let's Teach Kids to Code."[3] [**Watch 4:27- 13:21.** Note: if you have ample time, you're welcome to watch the full video; you can also watch the rest of the video online for free at a later time.]



Filmed November 2012 at TEDxBeaconStreet

Mitch Resnick: Let's teach kids to code

▶ **https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=en**

Consider

- What stands out to you about Professor Resnick's talk?
- What metaphor makes the most sense to you: Coding as a language? Coding as a subject like reading or writing? As a hobby? Something else?
- Does Professor Resnick convince you that kids should learn to code? Why or why not?
- How do you think you would have benefited from learning how to code in school? Do you think you would have anything to gain in learning how to code today?
- Thinking back to your perspective on coding when you first walked into the room today, has Resnick's talk changed your perspective? If so, how?

REFERENCES

[1] http://www.nytimes.com/roomfordebate/2014/05/12/teaching-code-in-the-classroom

[2] http://www.wired.com/2013/12/stop-thinking-that-coding-is-the-answer-to-all-of-societys-problems/

Kids as Coders

- **Talk to your child's school.**
  Talk to your kid's teacher and school administrators about what types of activities they may or may not be doing around coding. Some schools have actual computer programming classes, while others encourage kids' curiosity about programming through media labs, lunch groups, school clubs, or after-school offerings. At a minimum, some schools are participating in more one-and-done efforts (they are a start!) such as the Hour of Code.

- **Encourage the 4 C's for 21st-century learning.**
  The Partnership of 21st Century Skills (P21) outlines 4 C's — learning and innovation skills that relate to 21st-century readiness: critical thinking, communication, collaboration, and creativity. So how can you complement what your child is doing at school or at home? Learning to code has many benefits similar to solving a math problem, reading music, or learning another language. Aligned to the constructive theory of learning, it gives kids ample time for trial and error, experimentation, exploration, and failing forward. Seymour Papert (Schwarz, 1999), one of the best-known advocates for teaching computer science and programming in schools, sums this up well: "Anyone who has witnessed a toddler using a computer has probably experienced a sense of awe at that child's facility with what for adults can be an infinitely frustrating gadget. It's one thing for a child to play a computer game; it's another thing altogether for a child to build his or her own game. And this, according to Papert, is where the computer's true power as an educational medium lies — in the ability to facilitate and extend children's awesome natural ability and drive to construct, hypothesize, explore, experiment, evaluate, draw conclusions — in short to learn — all by themselves. It is this very drive, Papert contends, that is squelched by our current educational system."

The 4 C's for 21st-century learning is presented by the Partnership for 21st Century Skills (P21): www.p21.org/storage/documents/4csposter.pdf

- **Find resources online.**
  There is a plethora of coding opportunities online. From YouTube videos to sites like Scratch to games like *Minecraft*, kids (and adults) can learn the fundamentals through many easy-to-follow online activities. Some present the basics in block format, while others help structure the thinking process. Many of the apps and sites encourage kids to apply these programming concepts to their interests, whether gaming, fashion, storytelling, or artist expression.

*For more examples of coding apps, see our Graphite top picks.