

PROGRAMMING: WHAT SHOULD



Sam Patterson

(essay.writing@gmail.com) is a technology integration specialist at Gideon Hausner Jewish Day School in Palo Alto, CA. He brings a background of poetry and an EdD in literacy studies to the tech lessons he designs.

p

rogramming, from offline sequencing activities in kindergarten to command line coding an interactive kiosk presentation in high school, offers so many benefits to learners that it needs to be mandatory in K–12 schooling.

There are many great arguments for the need. England, Estonia, and Finland all have mandated programming in primary schooling. They are focused on trying to meet a European need for 900,000 more information and computer technology workers by the year 2020. These jobs are the foundation of the economy, as few fields are now untouched by automation, programming, and data science.

In a recent blog post titled “A Different Approach to Coding,” Mitchel

Resnick and David Siegel reminded the educational community that programming in schools is much bigger than solving puzzles and mazes. Programming is a means of communication and creative expression.

What needs to be mandatory in all schools is this view of programming as creative expression. Schools need to place creative expression as a top priority—and programming is one of the creative tools all students should learn.

Leveled tutorials and games can be a meaningful piece of the learning-to-program experience, but the real power comes in when teachers design meaningful learning opportunities that use programming as a mode of expression.



Amy Hutchison

(amyhutch@iastate.edu) is an associate professor of literacy at Iowa State University. Her research interests include the study of the new skills, strategies, and dispositions required to read, write, and communicate with digital technology.

a

s I write this article, I have just completed several hours of Codecademy courses in one sitting. Codecademy is a site where anyone can participate in online coding classes for free. I am congratulating myself on my new skills! Yet, I am also chastising myself for spending hours completing a task that, although interesting, did not directly relate to my goals for the day, which are ultimately determined by my job requirements. As beneficial as learning these skills may be, learning them in isolation and without the context of how they might be used in my job is not helpful.

This is a lot like the ways we often use technology in the classroom. We use it in ways that are interesting and engage

students, and may even teach them new skills; but are those skills the ones they are required to learn? Do we help them apply these skills to different contexts?

I am a huge fan of relatively new initiatives such as Codecademy’s Hour of Code and sites such as Scratch and Tynker, which provide opportunities for students of all levels to gain coding skills. I think all students should be exposed to computer science and have the opportunity to develop computer science skills. However, I think it is essential that teachers’ coding instruction is carefully planned and entirely intentional.

Sites such as Scratch provide excellent opportunities for students to learn some of the disciplinary literacies

BE MANDATORY?

Two educators weigh in on one of digital and computer science literacy's hottest topics

How can programming be used in class for expression and assessment? Here are some examples from the curriculum at my school this year:

- Program letters into words for spelling (kindergarten)
- Program a dance and share it with your classmates (kindergarten)
- Create a game that groups like vowel sounds (grade 1)
- Create a dialogue between two characters (grade 2)
- Explore geometry through patterns and loops (grade 2)
- Model the break-up of Pangea (grade 2)
- Create an animated presentation (grade 3)
- Animate a model of photosynthesis with all elements labeled (grade 5)
- Create a calculator (grade 6)
- Model the behavior of molecules in different states of matter (grade 7)
- Build a game that explains how digestion works (grade 7)

When we ask students to write a program that explains, illustrates, or computes, we are asking them to develop and apply a conceptual understanding. This process requires students to engage the content deeply and empowers them to tell their own stories of knowing.

When programming is in all schools, it needs to be there as a part of everyday work and expression. By thinking of programming as a mode of expression, like writing, we can begin to explore the wealth of applications in every grade level. ■



of computer scientists. However, in the school context, these sites will be most effective when they are used in a way that is guided by specific learning goals.

For example, there are many ways Scratch could be used to develop story narratives, to view and summarize text presented in diverse media formats,

for writing technical directions, for engaging in collaborations with diverse partners, and for providing a context in which students must use a range of general academic and domain-specific words and phrases. If these types of activities sound familiar, it's because they are taken from the Common Core

State Standards, and there are many ways to use coding sites to address these standards. However, to do so, instruction and activities involving coding must be thoughtfully planned.

So, should we make coding mandatory? Not until there is clear evidence about the ways it can be taught and integrated effectively. The increasingly available apps and sites intended for teaching coding *can* be great additions to the classroom. They help expose students to careers that many would not otherwise have exposure to. However, these tools should be used intentionally and through an approach that supports the development of students' digital and nondigital literacy skills. ■