Coding: An essential literacy

by Roxanne Ciddor, Bialik College

We teach kids maths, despite the fact that very few will need maths in their professional lives and calculators are prolific. We teach kids the fundamentals of music, although few will become professional musicians. We teach kids to read and write, despite the fact that few will become professional writers. This is because understanding and being able to do are very different. Reading to a child has its merits but teaching a child to write unlocks countless possibilities for communication and understanding. Why, then, do we teach children how to *read* technology but not how to *write* it? Coding is the literacy of technology and it as essential to success in our students’ future professional lives as English literacy and numeracy.

So what is coding? It refers to the building blocks of technology. It is the fundamental principle behind how and why technology operates.

Some common misconceptions about coding:

* Computers are so complicated; our students could never learn enough to actually do anything with it.
  + Contrary to popular opinion, computers are not in themselves intelligent. They merely run the code that is programmed into it by a human. The general concepts behind even the most complex program are understandable to most average students.
* Programming is like learning Latin: interesting but redundant.
  + Programming is the language of technology. Understanding it enhances our use of technology.
* Programming is only needed for students planning to become programmers in adult life.
  + All students can benefit from the concepts and ideas behind programming. Almost all industries will require workers with technological literacy in the future.
* Teaching programming will require teacher retraining and lots more expensive technology.
  + Amazing resources already exist that allow kids to program using literally any device and as simply or with as much complexity as they choose. Teachers can learn alongside their students, modelling teaching and learning dispositions. It is not something that needs to be completely understood before it can be taught.
  + Teaching coding encourages teachers to develop a “technology mindset”. Similar to Carol Dweck’s Growth Mindset, a technology mindset refers to the need for teachers to let go of their positions as the font of all knowledge and expect students’ technology skills to exceed their own.

So much time is spent teaching children to use programs they could figure out for themselves and which will probably be replaced by new programs before they have even left school. Technology is an essential and all-pervasive part of modern life. They connect the physical world to the virtual world. There is very little we do daily that does not, on some level, involve technology. If we don’t teach children to understand this technology, then we are sentencing them to be mere consumers of technology, increasingly at the mercy of those can create and maintain these devices. As such, we will create a social and intellectual divide between those who *make* and those who *use* technology.

The need to teach programming, especially from a young age, has been recognised by several countries. At the end of 2011, the then Mayor of New York Michael Bloomberg tweeted his new year’s resolution: that he would learn to code. American President, Barak Obama is supporting initiatives designed to promote coding in schools, like the recent “hour of power”. In Estonia, all first graders are being taught how to code. The next version of the British national curriculum, due in September this year, will include computer programming at all levels, starting at age five. In Australia, very few schools have joined the coding revolution.

While teaching coding is not designed to turn all students into computer scientists, the statistics for the industry are startling. These are American statistics but they indicate global trends:

* by the year 2020, computer jobs will have grown 22%
* of over 1.4 million jobs all over the world, the strongest demand being for software developers
* the job growth for programmers is growing at two times the United States national average
* if the current trend continues only 30% of those computer-related jobs will be filled

(<http://www.business2community.com/infographics/cracking-code-infographic-0795244#!TR8V3>)

By teaching coding, more students will see careers in the computer industry as possible. This includes previously marginalised groups, such as girls. By teaching coding in Australia we are also combatting a decline in manufacturing by fuelling a move towards creating a country rich in innovation and home-grown technologies.

Employment opportunities aside, by learning to code, children will learn skills relevant to all areas of life, such as:

* problem solving and adaptation
* collaboration
* digital citizenship
* resilience and determination
* creativity
* logic and reasoning
* design process principles
* communication skills
* documentation skills

How do we teach coding? Coding can be easily taught through existing websites and applications. The most outstanding program is MIT’s Scratch (<http://scratch.mit.edu/>). For devices, “Daisy the Dinosaur”( [www.daisythedinosaur.com/](http://www.daisythedinosaur.com/)) and “Hopscotch” (<https://www.gethopscotch.com/>) use the same block structure as Scratch yet are approachable to young students. For teaching and learning more complex coding and programming languages, try Code Academy (<http://www.codecademy.com/>).

At Bialik, our coding journey is starting in the Primary School with Scratch. Scratch is a visual programming language developed by Massachusetts Institute of Technology. To code, users connect blocks to create a script. Each object, or sprite, has its own script and sprites can interact with each other and their background.

Scratch allows for a physical interaction with the program created using voice, webcam or basic computer circuitry, such as the MaKey MaKey (<http://www.makeymakey.com/>), opening even more possibilities to teach and learn.

One of the most exciting aspects of teaching students to code is the ability for students to share and learn from any student in the world. In Scratch, all programs created are open source, allowing anyone to “see inside” the programming to understand and then adapt or learn from the coding behind it. For students, there is no better validation they can receive for their efforts than the use of something they have made by a peer they may never meet.

The reason to teach coding to our children can best be summed up by Mitch Resnick, head of the MIT team that created Scratch:

“they are learning to code but more importantly, they are coding to learn” (<http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code>)

**References**

BBC News, (2014). Are teachers ready for the coding revolution?. [online] Available at: http://www.bbc.com/news/technology-25857276 [Accessed 3 Jun. 2014].

Curtis, S. (2013). Teaching our children to code: a quiet revolution - Telegraph. [online] Telegraph.co.uk. Available at: http://www.telegraph.co.uk/technology/news/10410036/Teaching-our-children-to-code-a-quiet-revolution.html [Accessed 3 Jun. 2014].

Gardiner, B. (2014). Adding coding to the curriculum. [online] The New York Times. Available at: http://mobile.nytimes.com/2014/03/24/world/europe/adding-coding-to-the-curriculum.html?referrer [Accessed 3 Jun. 2014].

Higgin, Holderman, Stokes, Wilson, Oh, E., Holderman, E., Holderman, E., Mendoza, K. and Peppler, K. (2014). How to Use Scratch for Digital Storytelling. [online] Graphite.org. Available at: http://www.graphite.org/blog/how-to-use-scratch-for-digital-storytelling?utm\_source=2014\_05\_15\_Graphite\_Weekly\_Newsletter&utm\_medium=email&utm\_campaign=weekly [Accessed 3 Jun. 2014].

Morrison, N. (2013). Teach Kids How To Code And You Give Them A Skill For Life. [online] Forbes. Available at: http://www.forbes.com/sites/nickmorrison/2013/12/27/teach-kids-how-to-code-and-you-give-them-a-skill-for-life/ [Accessed 3 Jun. 2014].

Naughton, J. (2012). Why all our kids should be taught how to code. [online] the Guardian. Available at: http://www.theguardian.com/education/2012/mar/31/why-kids-should-be-taught-code [Accessed 3 Jun. 2014].

Resnick, M. (2014). Let's teach kids to code. [online] Ted.com. Available at: http://www.ted.com/talks/mitch\_resnick\_let\_s\_teach\_kids\_to\_code [Accessed 3 Jun. 2014].

Ted.com, (2013). Should we start teaching children in Primary and/or Secondary School "Coding"? | A conversation on TED.com. [online] Available at: http://www.ted.com/conversations/16752/should\_we\_start\_teaching\_child.html [Accessed 3 Jun. 2014].

Wallace, B. and Plus, B. (2014). Cracking the Code [Infographic]. [online] Business 2 Community. Available at: http://www.business2community.com/infographics/cracking-code-infographic-0795244#!CGPc5 [Accessed 3 Jun. 2014].

YouTube, (2014). "Code Stars" - Short Film. [online] Available at: https://www.youtube.com/watch?v=dU1xS07N-FA [Accessed 3 Jun. 2014].

YouTube, (2014). An Introduction to the Scratch Programming Language for Education. [online] Available at: https://www.youtube.com/watch?v=4y6J2jXjU34 [Accessed 3 Jun. 2014].