I’ve enjoyed several documentaries about the origin of our early (modern) computing systems; i.e., Steve Jobs, Steve Wozniak, Bill Gates, and Paul Allen. My favorite is “The Pixar Story,” because I’ve always been impressed with computer animation (starting with [Luxo Jr.](https://www.youtube.com/watch?v=FI0T0Oj7WFE), a study in light and shading that got as much attention for characterization as it did amazing graphics). Other amazing works as well – I can’t tell you how many times I watched/listened to the Animusic videos (such as [Resonant Chamber](https://www.youtube.com/watch?v=toXNVbvFXyk)) – they captured my interest and amazement. These things transcended the amusement factor and became emotionally impactful. (This is one of the key things that continues to make Pixar stand out, and the reason Disney bought them; and subsequently let the studio do what they want because they possessed skills Disney had yet to achieve.)

This is merely an anecdotal reflection of intentional and choice-based impact; meaning that these animations were designed to have an impact, and it works really well. But these small boxes of electricity-laden wires impact close to everything today. A new car has more than 100 million lines of code. This—to me—is absolutely insane. My family’s car used to turn on the old-fashioned way, a dodgy radio could (sometimes) tune in music, and we had to roll our windows down by turning a crank that stuck out of the door. What is even going on?

The more we want technology to do for us, the more complicated it becomes. If we want our van to remember what radio station to play and what air conditioning settings we prefer, we command and store with code. What boggles my mind is the idea that because all of this is written in a variety of languages that are all ultimately built on a I/O switch, even a complex A.I. program can be built out of physically-moving cogs and gears. “How can a room full of machine parts be referred to as living?” my grandfather, a science-fiction aficionado, will happily ask if the subject comes up. For Ada Lovelace to have recognized the processing potentials of textile machines, sending us down a path where an animated lamp can conjure an emotional reaction, is simply amazing.

She understood how to begin writing a language out of this two-letter-alphabet. These words are meaningless without the input🡪processing🡪output model, enabling us to diversify our interactions, and increase our expectations. There is an estimated 50 languages currently in use today, and somewhere around 9,000 languages in existence. There’s an evolution at work here that starkly parallels the anthropological study of spoken languages through time.

I’ve used HTML and CSS, C++ and whatever programming is installed on the Ti-83plus graphing calculator… I’ve dabbled in block coding too – except I did that AFTER the other ones, so while I build with these “chunks,” I know what is happening beyond this GUI format. My kids are exploring JavaScript, and whatever drives Roblox. I think it’s good for them to begin understanding what happens inside these small boxes of electricity-laden wires, HOW a webpage is oh-so-magically appearing on their computer screen, and even what makes their $5 pedometer watches work.

There’s such a structure to these languages; these languages that drive our cars and remember our air-conditioning presets. What if we all knew how to read and write those languages? What if we could visualize those electrical processes like a Rube-Goldberg machine? What if we had even more minds working on ways to write and rewrite, more minds reimagining the capabilities, and more diversity feeding those lines of code to push technology toward equity? Computers rely on true/false and I/O and stored data and processing routines… but our minds possess something which even the best of today’s A.I. can’t manage: creativity.

I don’t personally think that computer languages should be taught instead of spoken languages (especially as a teacher). I do personally think that the time and attention we give computer science classes isn’t enough. We no longer incidentally use technology, no longer *might* use the internet frequently for research, and we absolutely DO rely on computer technology every day. (I don’t know what I’d do without my phone and watch to help me navigate the day.) We teach languages, math, science, and we even teach teaching so new teachers can teach languages, math, science, and …teaching. We’re told that these subjects activate the right half of the brain and those subjects activate the left half of the brain. And then we do the worst thing imaginable and say “you’ll be proficient with one set or the other.” Maybe because we buy into the yes/no this/that I/O left/right binary ideologies. But writing code is a unique opportunity to exercise the fact-based logic-driven problem-solving experience with the amazing beauty of creativity and art and music.

Lovelace taught herself that balance. We learned from it. If we want to keep growing through new and innovative means – we have to teach it.